МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

М. МИНУ

М. МИНУ

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

ТЕОРИЯ И АЛГОРИТМЫ

Перевод с французского и предисловие
А. И. ШТЕРНА

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1990

ББК 22.18
М61
УДК 519.85

Collection Technique et Scientifique
des Télécommunications

MICHEL MINOUX

PROGRAMMATION MATHÉMATIQUE

THÉORIE ET ALGORITHMES

DUNOD

Минуа М. Математическое программирование. Теория и алгоритмы: Пер. с фр. и предисловие А. И. Штерна.— М.: Наука. Гл. ред. физ.-мат. лит., 1990.— 488 с.— ISBN 5-02-013980-7.

С единых позиций рассматриваются разделы математического программирования. Отражаются новые достижения. Излагаются теория и алгоритмы конечномерной и бесконечномерной оптимизации, в частности методы решения задач вариационного исчисления и оптимального управления, дискретное и динамическое программирование, способы декомпозиции больших систем. Рассматриваются разнообразные приложения. Простота и наглядность изложения совмещаются со строгостью доказательств. Вспомогательные сведения из других разделов математики даются в специальных главах и приложениях.

Для научных работников и инженеров, работающих в области прикладной математики, *а также для студентов вузов.*

Табл. 21. Ил. 90. Библиогр. 578 назв.

М $\frac{1602110000-088}{053(02)-90}$ 25-89

ISBN 5-02-013980-7

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ ПЕРЕВОДЧИКА

Исторически первой теоремой математического программирования может считаться теорема Ферма об обращении в нуль производной дифференцируемой функции в точке экстремума (сформулированная, кстати, задолго до появления понятий производной и дифференцируемой функции). Эта теорема имеет, помимо ее исключительного по широте приложений теоретического содержания, ясный вычислительный аспект, поскольку она связывает решение экстремальной задачи с нахождением корней алгебраических уравнений. С тех пор сложность экстремальных задач, возраставшая во многом в связи с потребностями практики, увеличилась неизмеримо. Но и современное математическое программирование, вобравшее в себя все то из экстремальных задач, что направлено на явное определение экстремума, и располагающее богатым арсеналом мощных вычислительных средств, сохраняет тесную связь с фундаментальными структурами анализа. Книга Мишеля Минуруководителя научных исследований одного из парижских университетов и профессора Высшей национальной школы передовой техники и Центральной школы искусств и ремесел, не только хорошо излагает многообразие методов столь далеких друг от друга разделов математического программирования, как линейное и выпуклое программирование, одномерная оптимизация, оптимизация с ограничениями и без ограничений, целочисленное программирование, методы декомпозиции больших задач, динамическое программирование и бесконечномерная оптимизация (и это изложение ведется по возможности унифицированно, вокруг нескольких основных идей — многозначных отображений, седловой точки, теории двойственности, глобальной сходимости, функции возмущений). Она также демонстрирует тесное родство группы тщательно изученных задач математического программирования с оптимальными задачами на графах, в разработке которых автор принимал существенное участие,— задачами, связанными с потоками, мультипотоками и кратчайшими путями; привлечение комбинаторных средств приводит и к некоторым новым алгоритмам в решении старых задач. Книга удачно сочетает ясность изложения движущих теоретических идей с тщательной продуманностью предлагаемых алгоритмов решения задач, и поэтому, как и указывает в предисловии автор, будет полезна широкому кругу читателей — от практиков до исследователей.

А. И. Штерн

ПРЕДИСЛОВИЕ

Если условиться считать зарождением математического программирования открытие симплекс-метода в 1947 г., то можно считать, что наша дисциплина насчитывает четыре десятилетия.

Первое десятилетие рассматривается как период развития линейного программирования и создания теоретических основ нелинейного программирования.

Второе посвящено зарождению теории решеток, дискретного и невыпуклого программирования, динамического программирования и теории управления, а также методов декомпозиции больших систем. Третье десятилетие было отмечено доведением до зрелого состояния всех вспомогательных дисциплин, а также развитием теории недифференцируемой оптимизации и, наконец, наилучшим соединением математического программирования с теорией графов, которое привело к комбинаторной оптимизации. Это десятилетие положило также начало теории сложности вычислений, влияние которой на математическое программирование еще сильно ощущается в четвертом десятилетии.

Разумеется, нельзя считать простой исторической случайностью, что появление математического программирования совпало с появлением компьютеров. На первом этапе в математическом программировании решение линейной задачи с сотней переменных и десятком ограничений расценивалось как огромный шаг вперед. Сегодня же решают линейные задачи с десятками тысяч переменных и тысячами ограничений; решаются также задачи течения в сетях с несколькими миллионами дуг и даже задачи, считавшиеся недоступными, такие как задача о коммивояжере, число переменных в которой может доходить до 100 000.

Все это есть результат развития как теории алгоритмов, так и компьютеров, как математики, так и электроники. По этому поводу я хотел бы заметить, что если развитие компьютеров — заслуга по преимуществу американцев, то развитие математической основы алгоритмов и «логики» математического программирования есть результат географически сильно распыленных исследований и здесь вклад французских ученых находится в первых рядах.

В настоящее время математическое программирование достигло определенной степени зрелости. Для большинства составляющих его разделов опубликовано достаточно много монографий и учебников. Иное положение с работами, относящимися к синтезу, до-

статочно широкому для изучения математического программирования с единых позиций.

Книга Мишеля Мину представляет собой попытку заполнить этот пробел, и притом осуществленную весьма удачно.

Изложение организовано вокруг нескольких центральных понятий и успешно покрывает очень широкий спектр вопросов, не сбиваясь на излишнюю специализацию или фрагментарность.

В некоторых вопросах, таких как методы субградиента, читателю представлен детальный отчет о важнейших современных достижениях, собранных вместе впервые в этой книге.

В других вопросах дается лишь введение и отправная точка для дальнейших более глубоких и детальных исследований.

Эта книга будет настольной книгой как математика, так и практического инженера.

Эгон Балаш,
профессор университета
Карнеги-Меллон,
Питтсбург, США

ВВЕДЕНИЕ

Объектом для теоретического изучения в математическом программировании являются задачи оптимизации — от их постановки до нахождения алгоритмов решения.

Присутствие в названии дисциплины термина «программирование» можно объяснить исторически тем, что первые исследования и первые приложения развивались в прямом контакте с экономикой и операционным исчислением.

Вполне естественно, что терминология отражает тесную связь, существовавшую между математической постановкой задачи и ее экономической интерпретацией (изучение оптимальной экономической программы).

Именно тогда был предложен [1]) термин «линейное программирование» для изучения теоретических и алгоритмических задач, связанных с оптимизацией линейных функций при линейных ограничениях.

Кун и Таккер [2]) в том же смысле использовали название «нелинейное программирование» для изучения нелинейных задач оптимизации с ограничениями или без них. Термин «целочисленное программирование» предложил Гомори [3]) для задач оптимизации, в которых на переменные наложено ограничение — принимать только целочисленные значения, тогда как термин «динамическое программирование» был введен Р. Беллманом [4]) для общего метода оптимизации динамических систем (т. е. развивающихся с течением времени).

Однако, несмотря на очевидное различие этих тем, затронутых между 1945 и 1960 гг., постепенный рост осознания глубокого родства между различными классами задач — как по структуре, так и по методам — быстро привел к их объединению в лоне новой, более широкой дисциплины — так называемого математического программирования [5]), что свидетельствует о широком унифицирующем движении, которое выглядит еще не вполне достигшим своих целей.

---

[1]) *Dantzig G. B.* (1949), «Programming in a linear structure», Econometrica, vol. 17, n° 1.

[2]) *Kuhn H. W., Tucker A. W.* (1951), «Nonlinear programming», Econometrica, vol. 19, p. 50—51.

[3]) *Gomory R. E.* (1958), «An algorithm for integer solutions to linear programs», Princeton, IBM Mathematics research project, Technical Report, n° 1.

[4]) *Bellman R.* (1957), Dynamic programming, Princeton University Press.

[5]) Термин «математическое программирование» впервые официально появился, по-видимому, в 1959 г. в названии международного совещания: The RAND Symposium on Mathematical Programming, Santa Monica, California (1959).

В настоящее время математическое программирование является одним из наиболее активно развивающихся разделов прикладной математики, и для этого имеются многочисленные основания. Может быть, главным из них является многообразие видов и важность его приложений как в технике, так и в других областях прикладной математики. Не пытаясь исчерпать все, можно указать приложения:

— в исследовании операций: оптимизация технико-экономических систем (планирование, эконометрия), транспортные задачи, управление (в том числе запасами) и т. д.;

— в численном анализе: аппроксимация, регрессия, решение линейных и нелинейных систем, численные методы, связанные с включением методов конечных элементов, и т. д.;

— в автоматике: распознавание систем, оптимальное управление системами, фильтрация, управление производством, роботы и т. д.;

— в технике: управление размерами и оптимизация структур, оптимальное планирование сложных технических систем — таких, как информационные системы, сети компьютеров, транспортные и телекоммуникационные сети и т. д.;

— в математической экономике: решение больших макроэкономических моделей (типа модели Леонтьева и родственных ей), микроэкономических моделей или моделей предпринимательства, теория принятия решений и теория игр.

Но важность и ценность математического программирования связаны также с тем, что оно дает адекватные понятийные рамки для анализа и решения многочисленных задач прикладной математики. Важность понятия седловой точки в теории игр общеизвестна, и многие ее методы решения имеют своим источником исследования по математическому программированию. В численном анализе вариационная формулировка многих задач и распространение на случай функциональных пространств основных конечномерных алгоритмов приводят к систематически используемым инструментам изучения уравнений с частными производными или задач оптимального управления. В комбинаторном программировании важнейшие базовые алгоритмы (в задачах о потоках, в графах, о матроидах и пересечении матроидов) происходят из исследований по математическому программированию и используют понятия двойственности, дополнительности и унимодулярности. Множество накопленных таким образом результатов привело к созданию около 1974 г. теории сложности, которая, как известно, является объектом интенсивных исследований в связи с ее теоретическими и практическими следствиями в прикладной математике и в информатике.

Этим, без всякого сомнения, объясняется тот факт, что с математическим программированием в последние 30 лет связана высокая активность исследований и значительный объем публикаций. Но среди этого разнообразия в центре внимания постоянно находится проработка основных понятий и тенденция к унификации, тем более, что история этой науки отмечена некоторыми замеча-

тельными синтетическими работами, которые часто давали импульс потоку новых и плодотворных исследований. В линейном программировании можно указать появившиеся в 1962 г. книги Хэдли и Симоннара, как и, конечно, монументальный труд Данцига 1963 г. В нелинейном программировании нужно особенно обратить внимание на книгу Зангвилла (1969 г.), где в первый раз появилась общая теория сходимости, как и на книгу Люенбергера (1973 г.), в которой был систематически проведен сравнительный анализ алгоритмов с точки зрения понятия скорости сходимости. В целочисленном программировании нужно указать труд Ху (1969 г.), содержащий, в частности, синтетическое изложение работ Гомори, и книгу Горфинкеля и Немхаузера (1972 г.). В области комбинаторной оптимизации отметим работу Лавлера (1976 г.) и книгу Гондрана и Мину (1979 г.), тогда как в динамическом программировании следует сослаться на классический для дальнейшего текст Беллмана (1957 г.). В области оптимизации больших систем укажем замечательное синтетическое изложение Ласдона (1970 г.), который показал, в частности, важность теории двойственности и недифференцируемой оптимизации в методах разложения. Наконец, в области бесконечномерной оптимизации книги Люенбергера (1969 г.) и Сеа (1971 г.) достигают некоторой унификации бесконечномерного математического программирования и современных методов численного анализа.

Помимо указанных выше нескольких основных работ, каждая из главных тем математического программирования отражена в обширной специальной литературе. Но, напротив,— и это довольно любопытно — приводимая библиография показывает, что вплоть до последнего времени почти полностью отсутствуют достаточно обширные синтетические работы, покрывающие если и не все, то хотя бы большую часть следующих тем: линейное программирование, целочисленное программирование, нелинейное программирование, оптимизация больших систем, динамическое программирование, бесконечномерная оптимизация.

Настоящая книга как раз и имеет целью заполнить, насколько это удастся, указанную лакуну, представляя самую широкую возможную панораму теории и методов математического программирования, вплоть до самых последних исследований.

Глава 1 посвящена введению в некоторые основные понятия оптимизации. Там можно найти, в частности, основные понятия и результаты выпуклого анализа, постоянно используемые в книге в дальнейшем, а также введение в изучение сходимости алгоритмов, приводящее, с одной стороны, к теории глобальной сходимости (основанной на понятии многозначного отображения) и к теореме Зангвилла, а с другой стороны — к понятиям асимптотической сходимости и скорости сходимости, столь полезным в сравнительном анализе алгоритмов.

Глава 2 представляет краткое и сжатое изложение теории и алгоритмов линейного программирования. За тщательным изучением вырожденности или численных задач, связанных с порождением

кодов линейного программирования, которые требуют исследований очень большого объема, мы отсылаем к библиографии и довольствуемся этим. Действительно, нам кажется более предпочтительным сосредоточиться на основных результатах, не специфичных для линейного программирования (таких, как двойственность), которые мы найдем в следующих главах в гораздо более общей форме.

В главе 3 мы описываем основные алгоритмы одномерной оптимизации, которые образуют, как известно, основную составную часть методов нелинейного программирования и которые частично обусловливают их практическую эффективность. Свойства этих алгоритмов изучаются с точки зрения глобальной сходимости итеративных схем, содержащих эти алгоритмы.

В главе 4 изучается нелинейная оптимизация без ограничений; эта глава содержит, в частности, достаточно полный обзор основных численных методов, приложимых к дифференцируемым функциям. Изучаются также некоторые методы (алгоритмы субградиента), приложимые к задачам оптимизации не всюду дифференцируемых функций. Эти задачи приобрели в последние годы особенную важность, так как они появляются во многих областях: в алгоритмах, основанных на теории двойственности, в методах разложения, в комбинаторном и целочисленном программировании.

Затем в двух отдельных главах излагаются задачи нелинейной оптимизации с ограничениями. В главе 5 изучаются прямые методы (называемые также исходными методами); она начинается изложением, связанным с условиями Куна — Таккера, которые составляют, в известном смысле, ее теоретическую основу; затем вводятся фундаментальные понятия седловой точки и функции возмущения, которые встречаются при изучении достаточных условий оптимальности. Наконец, чередой проходят важнейшие классы прямых методов: методы возможных направлений, линеаризация, проекция.

Глава 6 посвящена теории двойственности и прямо или косвенно использующим двойственность методам: лагранжевым методам и методам штрафа. Здесь можно найти, в частности, изложение новейших результатов, касающихся распространения теории двойственности на невыпуклые задачи, а также подробный обзор методов расширенных лагранжианов, которые в настоящее время представляются наиболее мощными общими алгоритмами для решения сильно нелинейных задач.

В главе 7 мы представляем целый ряд методов целочисленного программирования, обращая особое внимание на связи этих методов с теорией, изложенной в предыдущих главах. Так, методы сечений представляют собой замечательное развитие техники линейного программирования. Лагранжево ослабление, связанное с методами оптимизации с помощью субградиентов, является систематическим средством получения хороших оценок снизу и делает методы разветвления в большом числе комбинаторных задач наиболее эффективными алгоритмами. Представление задач цело-

численного программирования в конечных группах, а также их формулировка в терминах кратчайших путей являются источником нового, очень многообещающего направления исследований, которое ведет к настоящей теории двойственности в целочисленном программировании и к новым алгоритмам, основанным на лагранжевом ослаблении.

Глава 8 посвящена оптимизации систем больших размерностей и техническим приемам обобщенного линейного программирования, лежащим в основе важнейших методов разложения. В отличие от традиционного изложения, здесь метод разложения Данцига — Вольфе изучается в контексте лагранжева ослабления и теории двойственности, что позволяет дать его изложение в духе предыдущих глав.

Подробно изучаются также как техника разложения по ресурсам, так и разложение Бендерса; указаны некоторые важные приложения к оптимизации больших сетей.

Глава 9 начинается с напоминания основных принципов динамического программирования, и, в частности, мы обращаем внимание на связи между теоремой оптимальности и принципом оптимальности. Существенные идеи и понятия вводятся на простых примерах. С другой стороны (порывая в этом отношении с традиционными изложениями), мы ведем изложение, избегая каких-либо ссылок на оптимизацию динамических систем (которые представляют собой не более чем одну из областей приложения динамического программирования). Затем мы даем обзор семейства основных технических приемов сокращения вычислений, благодаря которым динамическое программирование позволяет создавать эффективные методы решения. Важность и многообразие приложений динамического программирования побуждают в качестве заключения описать некоторые характерные примеры, такие как задачи целочисленной оптимизации, оптимизация динамических систем, фильтрация марковских процессов.

Наконец, глава 10 рассматривается как введение в бесконечномерную оптимизацию и ее важнейшие приложения: оптимальное управление системами, описываемыми обыкновенными дифференциальными уравнениями или уравнениями с частными производными. Кроме того, мы здесь коротко и ясно приводим основные сведения о фундаментальных математических структурах (банаховых и гильбертовых пространствах), а также об основных теоретических результатах (теорема о слабой компактности, теорема Вейерштрасса), чтобы показать, как осуществляется бесконечномерное обобщение основных методов, использовавшихся в конечномерном случае.

Этот краткий обзор по главам дает ясно понять, что, несмотря на большое разнообразие затрагиваемых тем, эта книга построена в своей основе вокруг небольшого ядра основных идей: теории многозначных отображений и глобальной сходимости, понятий седловой точки и функции возмущений, теории двойственности и ее обобщений. Именно эти идеи составляют сущность математического

программирования, и именно они реализуют единство настоящей книги.

Выбор содержания различных глав осуществлялся с учетом двойного требования: с одной стороны, дать читателю некоторое количество непосредственно применимых инструментов (алгоритмов) для решения задач оптимизации, которые могут ему представиться; с другой стороны, дать теоретические и концептуальные рамки, достаточные для понимания и проверки этих алгоритмов, а возможно, и для создания новых. Именно поэтому изложение теоретических результатов осуществляется — в той мере, в какой это возможно,— с заботой о прояснении связей с приложениями и с работой алгоритмов. Кроме того, всякий раз, когда не остается никакой возможности избежать исключения из текста слишком специальных теоретических исследований или слишком технических доказательств, сохранение которых могло бы повредить ясности и понятности текста в целом, мы стараемся указать их и отослать читателя к литературе.

С учетом важности рассматриваемой в книге области и усилий по унификации изложения, осуществленных и представленных в ее тексте, настоящая книга должна составлять полезный инструмент для работы — с одной стороны, для изучающих предмет, а с другой — для исследователей, которые могут использовать ее и для ссылок, и как ключ, открывающий доступ к библиографии, и, наконец,— для инженеров и практиков, которые могут использовать книгу как подробный путеводитель, помогающий им найти среди большого числа предлагаемых теорией методов именно те, которые будут наиболее эффективны при решении их задач.

Материал этой книги частично был выработан в ходе чтения нескольких курсов, которые автор вел в некоторых инженерных школах, таких как Высшая национальная школа телекоммуникации, Центральная школа искусств и ремесел, Высшая национальная школа передовой техники (ENSTA). Он является также результатом двенадцатилетних исследований и ежедневной практической работы в области математического программирования и комбинаторной оптимизации в Национальном центре изучения телекоммуникаций и в Высшей национальной школе передовой техники.

Я считаю нужным выразить здесь мою живейшую благодарность всем тем, кто, прямо или косвенно, смог помочь мне в продумывании и создании настоящей книги.

М. Мину

ГЛАВА 1
ОСНОВНЫЕ ПОНЯТИЯ

§ 1. Математическое программирование. Определения

Математическое программирование в самом общем виде можно определить как задачу оптимизации с ограничениями в пространстве $\mathbf{R}^n$:

$$\begin{gathered}\min f(x),\\ g_i(x)\leqslant 0, \qquad i=1,\ \ldots,\ m, \qquad (P)\\ x\in S\subset \mathbf{R}^n.\end{gathered}$$

Здесь вектор $x\in\mathbf{R}^n$ имеет компоненты x_1, x_2, ..., x_n, которые и являются неизвестными рассматриваемой задачи.

Функция f называется *целевой функцией* (иногда ее называют экономической функцией), а множество условий $g_i(x)\leqslant 0$ $(i=1, 2, \ldots, m)$ и $x\in S$ называется *ограничениями* задачи.

Различие (в (P)) двух типов ограничений ($g_i(x)\leqslant 0$ и $x\in S$), которое на первый взгляд может показаться искусственным, объясняется здесь двояким образом. Во-первых, как легко видеть из табл. 1, оно позволяет объединить в одной общей формулировке основные классы задач, встречающихся в последующих главах. Во-вторых, как мы увидим ниже, часто случается, что оба типа ограничений используются по-разному в алгоритмах решения задач. Например, в линейном программировании ограничения неотрицательности переменных, вообще говоря, неявно учитываются (см. гл. 2). То же самое можно сказать об ограничениях целостности для переменных в целочисленном программировании (см. гл. 7). В дальнейшем мы рассматриваем лишь случай минимизации. Это не будет сужением постановки, поскольку поиск максимума функции f сразу же сводится к задаче минимизации функции $g=-f$.

С другой стороны, заметим, что задача (P) содержит и ограничения типа равенства. В самом деле, ограничение типа $h(x)=0$ всегда можно заменить двумя неравенствами: $h(x)\leqslant 0$ и $-h(x)\leqslant 0$.

Решением задачи (P) называют любой вектор x, удовлетворяющий ограничениям, т. е. такой, что

$$g_i(x)\leqslant 0, \quad i=1, \ldots, m, \quad x\in S.$$

Оптимальным решением (или *глобальным оптимумом*) задачи (P) называют решение, минимизирующее $f(x)$ на множестве всех решений.

Вектор x^0 есть *локальный оптимум* задачи (P), если существует такая окрестность $V(x^0)$ точки x^0, что точка x^0 является глобальным оптимумом задачи

$$\min f(x),$$
$$g_i(x) \leqslant 0, \qquad i = 1, \ldots, m,$$
$$x \in S \cap V(x^0).$$

На рис. 1 иллюстрируются понятия глобального и локального оптимума для функции одного переменного.

Часто, как мы убедимся в дальнейшем, можно охарактеризовать локальные оптимумы задачи, т. е. сформулировать необходимые и/или достаточные условия того, что решение x есть локальный

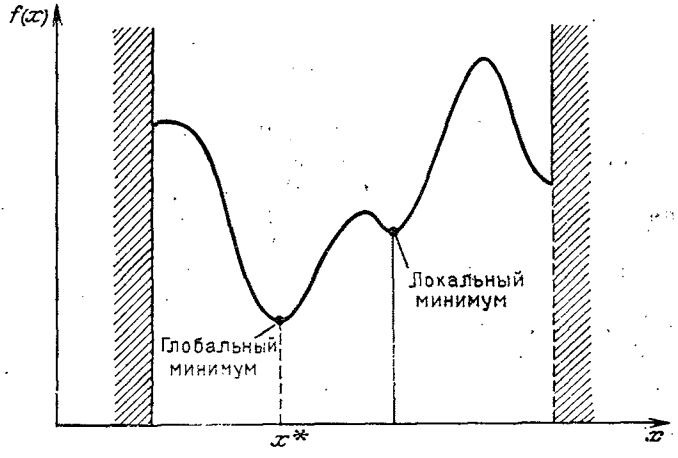


Рис. 1. Локальный оптимум и глобальный оптимум

оптимум. Напротив, глобальный оптимум задачи, вообще говоря, невозможно охарактеризовать, кроме как для очень частных случаев задач выпуклого математического программирования (см. п. 3.3). Этим объясняется трудность решения невыпуклых задач математического программирования, в том числе задач целочисленной оптимизации (см. гл. 7).

В табл. 1 приведена классификация различных типов задач оптимизации, встречающихся на практике, по свойствам функции f, функций g_i и по определению подмножества $S \subset \mathbb{R}^n$. Это позволяет прежде всего прояснить терминологию, принятую для задач оптимизации и кажущуюся несколько необычной на первый взгляд.

Отметим, наконец, что нам часто придется рассматривать задачи типа (P), содержащие лишь ограничения вида

$$g_i(x) \leqslant 0, \qquad i = 1, \ldots, m.$$

Таблица 1

Основные классы задач в математическом программировании

Функция f	Функции g_i	Множество S	Используемая терминология
Непрерывные, произвольные нелинейные		Связное, компактное $S \subset \mathbf{R}^n$	Непрерывное математическое программирование
Произвольные нелинейные (не обязательно непрерывные)		Дискретное (например, множество точек с целыми координатами, принадлежащих некоторому компакту)	Дискретное математическое программирование (если $S \subset \mathbf{Z}^n$, то нелинейное целочисленное программирование)
Непрерывная, произвольная нелинейная	$m = 0$	$S = \mathbf{R}^n$	Непрерывная (нелинейная) оптимизация без ограничений
Произвольная нелинейная (не обязательно непрерывная)	$m = 0$	$S = \mathbf{Z}^n$	Целочисленная (нелинейная) оптимизация без ограничений
Произвольные нелинейные и выпуклые функции		Выпуклое множество $S \subset \mathbf{R}^n$	Нелинейное выпуклое математическое программирование
Линейные		S — подмножество, задаваемое в $\mathbf{R}^n$ системой линейных равенств и неравенств (пример: $\mathbf{R}^{n+}$)	Линейное программирование
Линейные		$S \subset \mathbf{Z}^n$	Целочисленное линейное программирование

Это соответствует случаю, когда S совпадает со всем пространством $\mathbf{R}^n$. Для таких задач мы будем использовать обозначение

$$\begin{gathered} \min f(x), \\ g_i(x) \leqslant 0, \qquad i = 1, \ldots, m, \qquad (P') \\ x \in \mathbf{R}^n. \end{gathered}$$

§ 2. Элементы топологии

Рассмотрим *нормированное векторное пространство* $\mathbf{R}^n$ с евклидовой нормой

$$\|x\| = \left[\sum_{i=1}^{n} x_i^2\right]^{1/2}.$$

Эта норма позволяет ввести расстояние $d(x, y)$ между двумя элементами из $\mathbf{R}^n$:

$$d(x, y) = \|x - y\|.$$

Говорят, что $\mathbf{R}^n$ есть *метрическое векторное* пространство.

Вообще говоря, топологические понятия, вводимые в пп. 2.1, 2.2, равно как и сформулированные свойства, справедливы в метрических пространствах (при этом нормированные векторные пространства представляют собой частный случай метрических).

2.1. Сходимость последовательностей в $\mathbf{R}^n$ или в $\mathbf{R}$. Бесконечная последовательность $x^1, x^2, \ldots, x^k, \ldots$ векторов из $\mathbf{R}^n$ будет обозначаться $\{x^k\}_{k \in N}$ или просто $\{x^k\}$, если не возникнет путаницы относительно множества индексов. Говорят, что последовательность $\{x^k\}$ сходится к x (или что x есть *предел последовательности* $\{x^k\}$), если норма $\|x^k - x\|$ стремится к 0 при $k \to \infty$, или, что равносильно,

$$\forall \varepsilon > 0 \quad \exists N: \; k \geqslant N \Rightarrow \|x^k - x\| < \varepsilon.$$

Если x — предел последовательности $\{x^k\}$, то пишут $x^k \to x$.

Говорят, что x есть *предельная точка* последовательности $\{x^k\}_{k \in N}$, если из этой последовательности можно выбрать подпоследовательность $\{x^l\}_{l \in L} (L \subset N)$, сходящуюся к x. Аналогичным образом, x есть предельная точка последовательности $\{x^k\}$, если

$$\forall \varepsilon > 0, \quad \forall K \quad \exists k > K: \; \|x^k - x\| < \varepsilon.$$

Таким образом, сходящаяся последовательность имеет единственную предельную точку — предел последовательности (очевидно, что последовательность, имеющая единственную предельную точку, не обязательно сходится; можно в качестве упражнения построить противоречащий пример).

Наконец, введем для последовательностей действительных чисел $\{x^k\}$ $(x^k \in \mathbf{R})$ понятия верхнего предела и нижнего предела.

Верхним пределом (когда он существует) последовательности $\{x^k\}_{k \in N}$ называется такое действительное число, что:

1) $\forall \varepsilon > 0 \quad \exists K: \; k > K \Rightarrow x^k \leqslant y + \varepsilon$;
2) $\forall \varepsilon > 0, \quad \forall k > 0 \quad \exists k > K: \; x^k > y - \varepsilon$.

Тогда пишут

$$y = \limsup x^k.$$

Аналогичным образом, верхним пределом последовательности называют наибольшую предельную точку этой последовательности.

Нижний предел последовательности действительных чисел $\{x^k\}$ (обозначаемый $\liminf x^k$) может быть определен аналогично, и можно записать

$$\liminf x_k = -\limsup(-x^k).$$

Нижний предел последовательности есть наименьшая предельная точка этой последовательности.

2.2. Открытые и замкнутые множества. Пусть $S \subset \mathbf{R}^n$ — некоторое множество из $\mathbf{R}^n$.

Определение 1. Элемент $y \in S$ называется *внутренней точкой* множества S, если существует такое $\varepsilon > 0$, что

$$\|x - y\| < \varepsilon \Rightarrow x \in S$$

(иными словами, если существует шар с центром в y, целиком содержащийся в S).

Множество внутренних точек из S называется *внутренностью множества* S и обозначается $\operatorname{int} S$ (или $\mathring{S}$).

Примеры. Множество $A = \{x \mid \|x\| \leqslant 1\}$ имеет своей внутренностью единичный шар $B = \{x \mid \|x\| < 1\}$. Внутренность множества может оказаться пустой; например, это так для гиперплоскости в $\mathbf{R}^n$.

Определение 2. Множество S называется *открытым*, если оно совпадает со своей внутренностью, т. е. если $S = \operatorname{int} S$.

Примеры. Единичный шар $\{x \mid \|x\| < 1\}$ есть открытое множество.

Пустое множество $\varnothing$ и множество $\mathbf{R}^n$ открыты.

Свойство 1. *Пересечение конечного числа открытых множеств есть открытое множество.*

Объединение конечного или бесконечного числа открытых множеств есть открытое множество (доказательство предоставляется читателю в качестве упражнения).

Определение 3. Множество $V(x)$ есть *окрестность* точки $x \in \mathbf{R}^n$, если оно является открытым множеством, содержащим x. Равносильное определение: множество $V(x)$ есть окрестность точки x, если оно содержит шар с центром x.

Определение 4. Говорят, что $x \in \mathbf{R}^n$ есть *точка прикосновения* или *замыкания* множества S, если любая окрестность точки x пересекается с S, или, иначе,

$$\forall \varepsilon > 0 \quad \exists y \in S\colon \|x - y\| < \varepsilon.$$

Множество всех точек прикосновения множества S называется *замыканием множества* S и обозначается $\operatorname{cl} S$. Очевидно, что $\operatorname{cl} S \supset S$.

Пример. Замыкание единичного шара $B = \{x \mid \|x\| < 1\}$ есть множество $\operatorname{cl} B = \{x \mid \|x\| \leqslant 1\}$.

Определение 5. Множество $S \subset \mathbf{R}^n$ называется *замкнутым*, если оно совпадает со своим замыканием, т. е. $\operatorname{cl} S = S$.

Предложение 1. *Дополнение (в $\mathbf{R}^n$) открытого множества замкнуто. Дополнение (в $\mathbf{R}^n$) замкнутого множества открыто.*

Доказательство. Пусть S — открытое множество и $\bar{S}$ — его дополнение в $\mathbf{R}^n$.

Пусть $x \in S$. Существует шар с центром x, целиком содержащийся в S. Этот шар не пересекается с $\bar{S}$. Значит, x не является точкой прикосновения множества $\bar{S}$. Следовательно, все точки прикосновения множества $\bar{S}$ принадлежат $\bar{S}$, и, значит, $\bar{S}$ замкнуто.

Пусть теперь множество T замкнуто, а $\bar{T}$ — его дополнение в $\mathbf{R}^n$. И пусть $x \in \bar{T}$ — произвольная точка; эта точка не является точкой прикосновения множества T (поскольку T содержит все свои точки прикосновения). Значит, существует шар с центром x, не пересекающийся с T.

Таким образом, этот шар целиком содержится в $\bar{T}$, и, следовательно, множество $\bar{T}$ открыто.

Теперь переходом к дополнениям можем получить из свойства 1

Свойство 2. *Объединение конечного числа замкнутых множеств замкнуто. Пересечение конечного или бесконечного числа замкнутых множеств замкнуто.*

Фундаментальным свойством замкнутых множеств является

Свойство 3. *Множество $S \subset \mathbf{R}^n$ замкнуто в том и только том случае, если любая сходящаяся последовательность элементов из S имеет предел в S.*

Доказательство. Если x — предел последовательности $\{x^k\}$ элементов из S, то для любого $\varepsilon > 0$ существуют элементы последовательности, принадлежащие шару с центром x и радиусом ε. Значит, x есть точка прикосновения множества S. Стало быть, если S замкнуто, то любая сходящаяся последовательность элементов на S имеет предел в S.

Обратно, если S не замкнуто, то существует точка прикосновения x множества S, не принадлежащая S ($x \notin S$). А поскольку x — точка прикосновения, то для любого целого $k \geqslant 1$ найдется такая точка $x^k \in S$, что $\|x - x^k\| < 1/k$. Тем самым получаем последовательность $\{x^k\}$, предел которой не принадлежит S.

2.3. Компактные множества. Теорема Вейерштрасса.

Определение 6. Множество $K \subset \mathbf{R}^n$ называется *компактным*, если из любой последовательности $\{x^k\}_{k \in N}$ элементов из K можно выбрать подпоследовательность $\{x^l\}_{l \in L}$ ($L \subset N$), сходящуюся к элементу из K.

Приводимое ниже свойство дает простую характеристику компактных множеств в $\mathbf{R}^n$ (и, в более общей форме, в конечномерных нормированных векторных пространствах).

Свойство 4. *В пространстве $\mathbf{R}^n$ множество K компактно тогда и только тогда, когда оно замкнуто и ограничено (содержится в шаре радиуса $M < +\infty$).*

Это свойство перестает быть верным в бесконечномерном пространстве, что служит причиной некоторых трудностей решения задач оптимизации для бесконечной размерности. Однако мы увидим в гл. 10, как можно вновь найти эквивалент этого свойства посредством введения в бесконечномерных векторных пространствах топологии, отличной от топологии, ассоциированной с нормой, а именно слабой топологии.

Следующая теорема является фундаментальной и касается существования оптимального решения для задачи оптимизации, будет обобщена в гл. 10 для бесконечномерных оптимизационных задач.

Теорема 1 (Вейерштрасс). *Если f — непрерывная действительная функция на компактном множестве $K \subset \mathbf{R}^n$ (K замкнуто*

и ограничено), то задача

$$\min_{x\in K} f(x)$$

имеет оптимальное решение $x^* \in K$.

Доказательство. Пусть $m = \inf\limits_{x\in K}\{f(x)\}$ (значит, $\forall x \in K$ $m \leqslant f(x)$). Тогда существует такая последовательность $\{x^k\}$ элементов из K, что $f(x^k) \to m$ (m априори может быть $-\infty$).

Поскольку K компактно, то существует подпоследовательность $\{x^l\}_{l\in L}$ ($L \in N$), сходящаяся к $x^* \in K$.

В силу непрерывности функции f имеем $f(x^l) \to f(x^*)$ и, значит,

$$m = \lim_{k\to\infty} f(x^k) = \lim_{\substack{l\to\infty \\ l\in L}} f(x^l) = f(x^*).$$

Но $f(x^*) > -\infty$, поэтому $m > -\infty$, и

$$\forall x \in K \quad f(x^*) = m \leqslant f(x);$$

следовательно, $x^* \in K$ есть оптимальное решение поставленной задачи.

Итак, условие компактности существенно для доказательства существования оптимального решения в задаче оптимизации. В дальнейшем мы покажем, что именно условие компактности и позволяет доказать сходимость алгоритмов оптимизации и в $\mathbf{R}^n$ (гл. 1, п. 4.5), и в бесконечномерных пространствах (гл. 10, п. 3.1).

Приводимое ниже следствие сразу вытекает из теоремы Вейерштрасса, но оно часто используется в приложениях, например в случае оптимизации без ограничений, где отыскивается оптимум на всем пространстве $\mathbf{R}^n$.

Следствие 1. *Пусть* f *— непрерывная действительная функция на* $\mathbf{R}^n$, *удовлетворяющая условию*

$$f(x) \to \infty \quad \textit{при} \quad \|x\| \to +\infty.$$

Тогда задача

$$\min_{x\in\mathbf{R}^n} f(x)$$

имеет оптимальное решение x^*.

Доказательство. Пусть $x^0 \in \mathbf{R}^n$ — произвольная точка. Согласно условию возрастания на бесконечности, найдется такое $M > 0$, что

$$\|x\| \geqslant M \Rightarrow f(x) \geqslant f(x^0).$$

Стало быть, задача сводится к задаче оптимизации на замкнутом шаре $\{x \mid x \in \mathbf{R}^n;\ \|x\| \leqslant M\}$, который компактен, и, значит, применима теорема 1.

Заметим, наконец, что теорема 1 сразу же обобщается на полунепрерывные снизу функции (см. гл. 10, п. 3.2).

2.4. Замечания об обозначениях min и max. Как уже упоминалось в самом начале, обозначение $\min\limits_{\alpha\in X}\{f(x)\}$ будет использоваться

для таких функций f, что если выполняются условия

$$-\infty < \inf_{x \in X} \{f(x)\} < +\infty,$$

то существует элемент $\bar{x} \in X$, для которого

$$f(\bar{x}) = \inf_{x \in X} \{f(x)\}.$$

Тогда полагаем $\min\limits_{x \in X} \{f(x)\} = f(\bar{x})$.

Практически это сводится к предположению, что f удовлетворяет условиям теоремы Вейерштрасса или следствия из нее.

Для функций f, не ограниченных снизу на X (т. е. таких, что не существует $c \in \mathbf{R}$: $f(x) > c \quad \forall x \in X$), удобно принять соглашение

$$\min_{x \in X} \{f(x)\} = -\infty.$$

В этом случае будем говорить, что на X не существует конечного минимума, или что задача $\min\limits_{x \in X} \{f(x)\}$ не ограничена.

Наконец, если $X = \varnothing$, принимаем по соглашению

$$\min_{x \in X} \{f(x)\} = +\infty.$$

Условия использования обозначения $\max\limits_{x \in X} \{f(x)\}$ определяются аналогично.

§ 3. Элементы выпуклого анализа

В задачах оптимизации, как с ограничениями, так и без ограничений, важную роль играет понятие выпуклости. В самом деле, для большинства алгоритмов, которые будут описаны в книге, сходимость к глобальному оптимуму может быть доказана лишь при условиях выпуклости.

Большинство результатов этого параграфа приводится без доказательства. За деталями можно отослать читателя к книге [38]. С другой стороны, основные теоремы об отделимости выпуклых множеств содержатся в приложении 1.

3.1. Выпуклые множества.

Определение. Множество $S \subset \mathbf{R}^n$ называется *выпуклым*, если

$$\left.\begin{array}{l} \forall x \in S \\ \forall y \in S \end{array}\right\} \lambda x + (1-\lambda) y \in S$$
$$\forall \lambda \qquad (0 \leqslant \lambda \leqslant 1),$$

Иначе говоря, множество S *выпукло*, если для любых двух точек x, $y \in S$ отрезок $[x, y]$ целиком содержится в S (рис. 2).

Пусть заданы p точек $x^1, x^2, \ldots, x^p \in \mathbf{R}^n$. Говорят, что $x \in \mathbf{R}^n$ есть *выпуклая комбинация* этих точек, если существуют такие коэффициенты $\mu_1, \mu_2, \ldots, \mu_p \geqslant 0$, что

$$\sum_{i=1}^{p} \mu_i = 1, \quad x = \sum_{i=1}^{p} \mu_i x^i.$$

Легко видеть, что множество $S \subset \mathbf{R}^n$ выпукло в том и только том случае, если любая выпуклая комбинация точек из S принадлежит S.

Пусть задано множество $S \subset \mathbf{R}^n$. Обозначим через $\operatorname{conv} S$ *выпуклую оболочку* множества S, т. е. множество точек из $\mathbf{R}^n$, являющихся выпуклыми комбинациями точек из S. Отсюда следует,

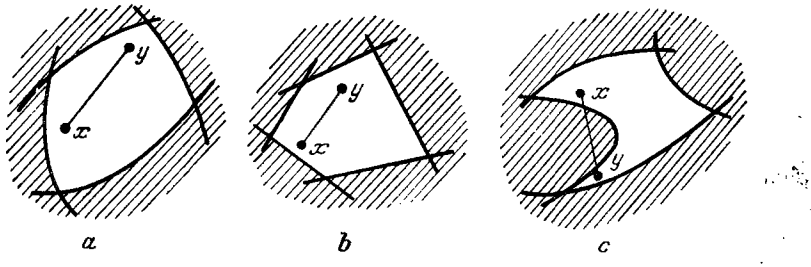


Рис. 2. Выпуклость множества: *a*) выпуклое множество; *b*) выпуклое множество (полиэдр); *c*) невыпуклое множество

что множество S выпукло в том и только том случае, если $S = \operatorname{conv} S$.

Свойство 5. *Пересечение конечного числа выпуклых множеств выпукло.*

Доказательство следует непосредственно из определения.

Определение 8. *Относительной внутренностью* выпуклого множества $C \subset \mathbf{R}^n$, обозначаемой $\operatorname{rint} C$, называется внутренность множества C относительно наименьшего аффинного многообразия, содержащего C.

На рис. 3 представлено выпуклое множество $C \subset \mathbf{R}^n$, лежащее в плоскости π. Здесь $\operatorname{rint} C$ рассматривается как подмножество плоскости π, изоморфной $\mathbf{R}^2$ (при ее определении используются двумерные окрестности).

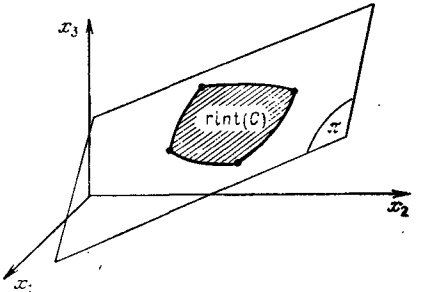


Рис. 3. Относительная внутренность выпуклого множества

Справедлива следующая

Теорема 2. *Всякое непустое выпуклое множество в* $\mathbf{R}^n$ *имеет непустую относительную внутренность.*

3.2. Выпуклые функции.

Определение 9. Говорят, что функция $f\colon \mathbf{R}^n \to \mathbf{R}$, определенная на выпуклом множестве S, *выпукла*, если она удовлетворяет

условиям

$$\left.\begin{array}{l}\forall x \in S \\ \forall y \in S \\ \forall \lambda \in [0, 1]\end{array}\right\} f(\lambda x + (1-\lambda) y) \leqslant \lambda f(x) + (1-\lambda) f(y);$$

функция f называется *строго выпуклой*, если для $x \neq y$ и $\lambda \in (0, 1)$ всегда выполняется строгое неравенство.

Из определения следует, что функция f выпукла на выпуклом множестве $S \subset \mathbf{R}^n$ в том и только том случае, если для любых x, $y \in S$ функция $g(\theta) = f(x + \theta(y - x))$ есть выпуклая функция от θ для $\theta \in [0, 1]$.

Определение 10. *Надграфиком* функции f называется (и обозначается $\operatorname{epi} f$) множество векторов (μ, x) с $n+1$ компонентами:

$$\{(\mu, x) \mid f(x) \leqslant \mu,\ x \in \mathbf{R}^n,\ \mu \in \mathbf{R}\} \subset \mathbf{R}^{n+1}.$$

Свойство 6. *Функция f выпукла в том и только том случае, если $\operatorname{epi} f$ есть выпуклое множество.*

Теорема 3. *Линейная комбинация с положительными коэффициентами от выпуклых функций есть выпуклая функция.*

Теорема 4. *Если функция f непрерывно дифференцируема, то эквивалентны условия* a) *и* b), *приводимые ниже.*

Если функция f дважды непрерывно дифференцируема, то эквивалентны условия a), b) *и* c):

a) *f выпукла*;

b) $\forall x, y$: $f(y) \geqslant f(x) + \nabla f^T(x) \cdot (y - x)$;

c) $\forall x$ *гессиан* $\nabla^2 f(x)$ *есть положительно полуопределенная матрица* $(\forall y\ y^T \cdot \nabla^2 f(x) \cdot y \geqslant 0)$.

Следствие 2. *Положительно полуопределенная квадратичная форма есть выпуклая функция.*

Доказательство сразу следует из теоремы 4, c).

3.3. Выпуклые задачи математического программирования

Определение 11. Говорят, что задача математического программирования *выпукла*, если она состоит в минимизации выпуклой функции (соответственно в максимизации вогнутой функции) на выпуклой области.

Таким образом, задача (P) из § 1 есть выпуклая задача математического программирования (или задача *выпуклого математического программирования*), если, например, f выпукла, g_i $(i = 1, \ldots, m)$ выпуклы, $S \subset \mathbf{R}^n$ выпукло.

Тогда основное свойство выпуклых задач формулирует следующая

Теорема 5. *Для выпуклой задачи всякий локальный оптимум является глобальным оптимумом.*

Доказательство. Можно всегда (без ограничения общности) рассматривать выпуклую задачу в виде

$$\min_{x \in S} f(x),$$

где f — выпуклая функция, S — выпуклое множество.

Пусть x^0 — локальный оптимум. Чтобы показать, что x^0 — глобальный оптимум, возьмем произвольный элемент $y \in S$ $(y \neq x^0)$ и покажем, что $f(x^0) \leqslant f(y)$.

Рассуждая от противного, допустим, что $f(x^0) > f(y)$. Отсюда в силу выпуклости функции f получаем

$$\forall \theta \in [0, 1]: \ f(x^0 + \theta(y - x^0)) \leqslant (1-\theta) f(x^0) + \theta f(y) < f(x^0),$$

что противоречит предположению, что x^0 — локальный оптимум. Значит, для любого $y \in S$ имеем $f(x^0) \leqslant f(y)$, чем доказано, что x^0 — глобальный оптимум.

3.4. Выпуклые функции в расширенном смысле. Иногда приходится рассматривать выпуклые функции, которые могут принимать значения $+\infty$ или $-\infty$. Приведем несколько примеров таких ситуаций.

Пример 1. *Если выпуклая функция определена на выпуклом* множестве $S \subset \mathbf{R}^n$, то можно доопределить ее на всем пространстве $\mathbf{R}^n$, условившись положить

$$f(x) = +\infty \quad \text{для} \quad x \notin S.$$

Тогда минимизация функции f на S эквивалентна минимизации «расширенной» функции без ограничений.

Пример 2. Пусть функция $\Phi(y)$ определена как оптимум задачи оптимизации с правой частью, зависящей от $y = [y_1, y_2, \ldots, y]^T$:

$$\begin{gathered} \min f(x) \quad (= \Phi(y)), \\ g(x) \leqslant y, \qquad (P_y) \\ x \in \mathbf{R}^n. \end{gathered}$$

Задача (P_y) может не иметь решения при некоторых значениях y; для таких y следует положить $\Phi(y) = +\infty$. Точно так же (P_y) может не иметь конечного оптимума для некоторых значений y. В этом случае полагаем $\Phi(y) = -\infty$. (Функция Φ называется *функцией возмущений*. Ее свойства будут изучаться более детально в гл. 5, § 2.)

Итак, если допустить, что $f(x)$ может принимать значения $+\infty$ и $-\infty$, то следует слегка изменить приведенное выше определение выпуклых функций.

Определение 12. Будем говорить, что f — (*расширенная*) *выпуклая функция*, определенная на $\mathbf{R}^n$ и принимающая значение в $\bar{\mathbf{R}}$, если

$$\left.\begin{array}{l} \forall x \quad f(x) \neq +\infty \\ \forall y \quad f(y) \neq +\infty \end{array}\right\} \Rightarrow (\lambda x + (1-\lambda) y) \leqslant \lambda f(x) + (1-\lambda) f(y) \quad \forall \lambda \in [0, 1].$$

Тогда надграфик расширенной выпуклой функции f определяется, как в п. 3.2.

Определение 13. *Эффективной областью* определения выпуклой функции называется (и обозначается $\operatorname{dom} f$) множество

$$\operatorname{dom} f = \{x \in \mathbf{R}^n \mid f(x) < +\infty\}.$$

Заметим, что множество $\operatorname{dom} f$ есть проекция надграфика $\operatorname{epi} f$ на $\mathbf{R}^n$. В самом деле,

$$\operatorname{dom} f = \{x \in \mathbf{R}^n \mid \exists \mu < +\infty:\ (x, \mu) \in \operatorname{epi} f\}.$$

Определение 14. Расширенная выпуклая функция f называется *собственной*, если множество $\operatorname{dom} f$ непусто и если

$$f(x) > -\infty \qquad \forall x \in \operatorname{dom} f.$$

На рис. 4 представлена собственная выпуклая функция одного переменного.

Определение 15. Пусть f: $\mathbf{R}^n \to \mathbf{R}$ — выпуклая функция. *Сопряженной* к функции f называется функция f^*, удовлетворяющая условию

$$\forall y \in \mathbf{R}^n \qquad f^*(y) = \sup_{x \in \mathbf{R}^n} \{y^T \cdot x - f(x)\}.$$

Легко видеть, что f^* — выпуклая функция.

Аналогично, функция, сопряженная к f^* и обозначаемая f^{**} (бисопряженная к f), определяется условием

$$f^{**}(x) = \sup_{x \in \mathbf{R}^n} \{y^T \cdot x - f^*(y)\}.$$

Теперь можно ввести понятие замкнутой выпуклой функции.

Определение 16. Функция f называется *замкнутой выпуклой* в том и только том случае, если $f^{**} = f$.

Функция f^{**} (называемая также *замыканием* функции f) соответствует регуляризации функции f на границе ее эффективной

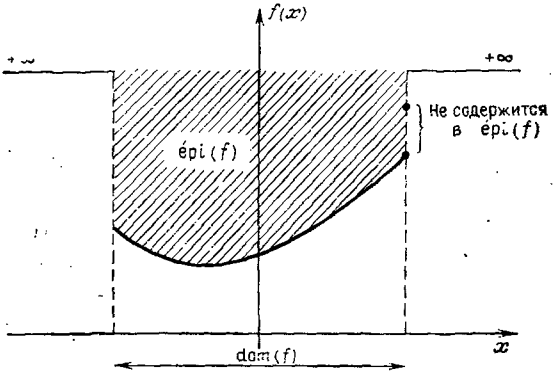


Рис. 4. Собственная выпуклая функция и ее эффективная область

области. Нетрудно доказать, что если f — выпуклая функция, то ее сопряженная f^* — замкнутая выпуклая функция.

3.5. Субградиенты. Субдифференциал. Для дифференцируемой выпуклой функции f *градиент* $\nabla f(x^0)$ в любой точке x^0 удовлетворяет фундаментальному неравенству

$$f(x) \geqslant f(x) + \nabla f^T(x^0)\,(x - x^0).$$

Если выпуклая функция дифференцируема не всюду, то понятие градиента обобщается понятием субградиента.

Определение 17. *Субградиентом* функции f в точке x^0 называется вектор $\gamma=(\gamma_1, \gamma_2, \ldots, \gamma_n)^T \in \mathbf{R}^n$, удовлетворяющий условию

$$f(x) \geqslant f(x^0)+\gamma^T \cdot (x-x^0).$$

В пространстве $\mathbf{R}^{n+1}$ гиперплоскость

$$z=f(x^0)+\gamma^T \cdot (x-x^0),$$

пересекающая надграфик функции f в точке $(x^0, f(x^0))$ и лежащая всюду не выше этой точки, называется *опорной гиперплоскостью* надграфика функции $f(x)$ в точке x^0.

Фундаментальным результатом является следующая [38]

Теорема 6. *Всякая расширенная собственная выпуклая функция f, определенная на $\mathbf{R}^n$, имеет субградиент в любой точке $x^0 \in \operatorname{rint}(\operatorname{dom} f)$.*

Определение 18. *Субдифференциалом* функции f в точке x^0 называется (и обозначается $\partial f(x^0)$) множество всех субградиентов функции f в точке x^0.

Теорема 6 означает, что для любой собственной выпуклой функции

$$\partial f(x^0) \neq \varnothing \quad \forall x^0 \in \operatorname{rint}(\operatorname{dom} f).$$

Если функция f дифференцируема в x^0, то субдифференциал в x^0 сводится к единственному элементу — градиенту функции f в точке x^0:

$$\partial f(x^0)=\{\nabla f(x^0)\}.$$

Теорема 7. *Для любой точки $x^0 \in \operatorname{rint}(\operatorname{dom} f)$ множество $\partial f(x^0)$ замкнуто и выпукло.*

Доказательство. Выпуклость следует непосредственно из определения; в самом деле, если

$$\gamma^1 \in \partial f(x^0), \quad \gamma^2 \in \partial f(x^0),$$

то

$$f(x) \geqslant f(x^0)+\gamma^{1T}(x-x^0) \quad \forall x,$$
$$f(x) \geqslant f(x^0)+\gamma^{2T}(x-x^0) \quad \forall x,$$

откуда

$$\forall \lambda \in [0, 1] \quad f(x) \geqslant f(x^0)+[\lambda\gamma^{1T}+(1-\lambda)\gamma^{2T}](x-x^0) \quad \forall x,$$

и, следовательно,

$$\lambda\gamma^1+(1-\lambda)\gamma^2 \in \partial f(x^0) \quad \forall \lambda \in [0, 1].$$

Чтобы показать, что множество $\partial f(x^0) \neq \varnothing$ замкнуто, рассмотрим сходящуюся к γ последовательность $\gamma^j \in \partial f(x^0)$ и предположим, что $\gamma \notin \partial f(x^0)$. Это значит, что найдется такой элемент $y \in \mathbf{R}^n$, что

$$-f(y)+f(x^0)+\gamma^T(y-x^0)=\varepsilon>0.$$

[illegible] для любого члена γ^j последовательности

$$\gamma^j \in \partial f(x^0) \Rightarrow f(y) - f(x^0) - \gamma^{jT} \cdot (y - x^0) \leqslant \|\gamma - \gamma^j\| \cdot \|y - x^0\|.$$

Отсюда

$$\|\gamma - \gamma^j\| \geqslant \frac{\varepsilon}{\|y - x^0\|} \quad \forall j,$$

и мы пришли к противоречию со сходимостью $\{\gamma^j\}$ к γ.

Минимум выпуклой функции позволяет охарактеризовать

Теорема 8. *Пусть f — собственная выпуклая функция. Точка x^0 минимизирует f в том и только том случае, если $0 \in \partial f(x^0)$.*

Доказательство. В самом деле, $0 \in \partial f(x^0)$ тогда и только тогда, когда

$$f(x) \geqslant f(x^0) + 0 \cdot (x - x^0) \qquad \forall x \in \mathbf{R}^n;$$

а это соотношение в свою очередь выполняется в том и только том случае, если x^0 минимизирует f (рис. 5).

3.6. Субградиенты и производные по направлению.

Определение 19. *Производной по направлению* для функции f в точке x^0 по направлению y называется (и обозначается

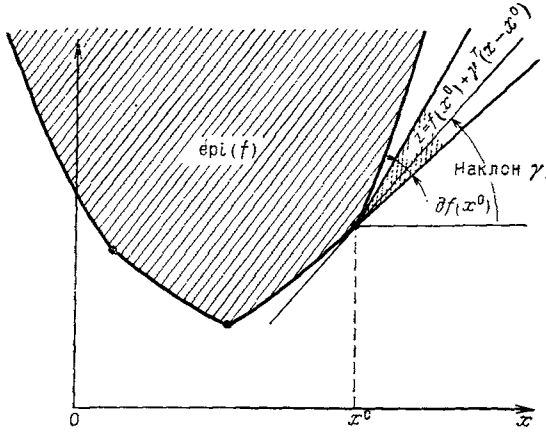


Рис. 5. Выпуклая функция f переменного x, не всюду дифференцируемая; ее надграфик epi(f) и субградиент функции f в точке x^0 (наклон опорной гиперплоскости к надграфику в точке x^0)

$\delta f(x^0, y)$) предел (который может быть равен $\pm\infty$) отношения $\frac{f(x^0 + \lambda y) - f(x^0)}{\lambda}$, когда $\lambda \to 0$, принимая положительные значения. Иначе:

$$\delta f(x^0, y) = \lim_{\lambda \to 0^+} \frac{f(x^0 + \lambda y) - f(x^0)}{\lambda}.$$

Можно заметить, что для выпуклой функции f функция от λ

$$g(\lambda) = \frac{f(x^0 + \lambda y) - f(x^0)}{\lambda}$$

убывает (в широком смысле), когда λ уменьшается, оставаясь положительным. (При этом, очевидно, предполагается, что $f(x^0)$ конечно.)

С другой стороны, если $\partial f(x^0) \neq \varnothing$ и если $\gamma \in \partial f(x^0)$, то

$$\frac{f(x^0 + \lambda y) - f(x^0)}{\lambda} \geqslant \gamma^T \cdot y \quad \forall \lambda.$$

Отсюда выводим, что $g(\lambda)$ имеет конечный предел; следовательно, $\delta f(x^0, y)$ существует в любой точке x^0, для которой $\partial f(x^0) \neq \varnothing$, и, значит,

$$\partial f(x^0, y) \geqslant \gamma^T \cdot y \quad \forall \gamma \in \partial f(x^0).$$

Важный результат о том, что предыдущее неравенство обращается в равенство по крайней мере для некоторого субградиента в x^0, выражает

Теорема 9. *Если множество* $\operatorname{dom} f$ *имеет непустую относительную внутренность, то для любого* $x^0 \in \operatorname{rint}(\operatorname{dom} f)$ *существует* $\delta f(x^0, y)$ *и*

$$\delta f(x^0, y) = \max_{\gamma \in \partial f(x^0)} \{\gamma^T \cdot y\}.$$

§ 4. Исследование сходимости. Глобальная сходимость и асимптотическая сходимость

4.1. Понятие алгоритма в математическом программировании. Большинство методов решения задач оптимизации, которые будут изложены в последующих главах, имеет итерационную природу, т. е. исходя из некоторой начальной точки x^0 они порождают потенциально бесконечную последовательность точек $x^0, x^1, \ldots, x^k, \ldots$, относительно которой мы рассчитываем, что она сходится к искомому оптимуму.

Алгоритм решения есть процесс, позволяющий исходя из заданной начальной точки x^0 строить последовательность $x^1, x^2, \ldots, x^k, \ldots$ Стало быть, алгоритм полностью определяется заданием отображения A, которое ставит в соответствие x^k и $x^{k+1} = A(x^k)$. Это позволит нам в дальнейшем не различать алгоритм и отображение, которое с ним связано. Тем самым исследование сходимости алгоритма сведется к изучению свойств отображения A.

4.2. Общая модель алгоритмов. Многозначные отображения. Заметим, однако, что представленная выше модель, позволяющая хорошо описать конкретный алгоритм, совершенно неприменима для описания класса алгоритмов. Но цель общей теории сходимости состоит скорее в исследовании, по возможности глобальном, поведения класса алгоритмов, чем в изучении всех алгоритмов класса, которые практически часто различаются лишь в деталях их использования.

Пусть $A_1, A_2, \ldots, A_p$ — некоторый набор алгоритмов, приписываемых одному семейству. Стало быть, естественно поставить за-

дачу о возможности глобального исследования их поведения. Тогда мы приходим к глобальному рассмотрению этого семейства алгоритмов как отображения с кратными компонентами, которые точке x^n ставит в соответствие p точек: $A_1(x^k)$, $A_2(x^k)$, ..., $A_p(x^k)$. Иначе говоря, семейство алгоритмов рассматривается как отображение, которое каждой точке x^k соотносит множество $A(x^k) = \{A_1(x^k), \ldots, A_p(x^k)\}$. Это естественно приводит к общей модели, в которой алгоритмы (или классы алгоритмов) представлены многозначными отображениями, т. е. отображениями $\mathbf{R}^n$ в $\mathscr{P}(\mathbf{R}^n)$, которые точке $x \in \mathbf{R}^n$ ставят в соответствие подмножество из $\mathbf{R}^n$.

Мы будем использовать обозначение $A\colon X \xrightarrow{m} Y$ для многозначного отображения X в Y.

Как мы увидим в гл. 3, § 3, эта модель будет также очень удобна для учета кратности локальных оптимумов или погрешностей, вызванных прерыванием итерационного процесса.

4.3. Понятие глобальной сходимости.

Определение 20. Будем говорить, что алгоритм, описываемый многозначным отображением A, *глобально сходится* (или обладает свойством глобальной сходимости), если для любой выбранной исходной точки x^0 последовательность $\{x^k\}$, порожденная точками $x^{k+1} \in A(x^k)$ (или подпоследовательность), сходится к точке, удовлетворяющей необходимому условию оптимальности.

Необходимым условием, входящим в это определение, будет, вообще говоря, стационарность в случае оптимизации без ограничений (гл. 4, п. 1.1) и условия Куна — Таккера в случае оптимизации с ограничениями (гл. 5, § 1).

Свойство глобальной сходимости выражает в некотором смысле *надежность функционирования* алгоритма и соответствует, вообще говоря, *минимальному требованию*, которое можно сформулировать для любого метода решения задач математического программирования. Очень важно подчеркнуть, что это свойство не влечет (в противовес тому, что можно было бы предполагать) сходимость к глобальному оптимуму для любой исходной точки x^0. Речь должна была бы идти об условии гораздо более строгом, а оно не может быть практически выполнено ни при каком из известных алгоритмов. Тем не менее можно отметить, что если только алгоритм обладает свойством глобальной сходимости, то достаточно потребовать условия выпуклости, чтобы получить в точности сходимость алгоритма к глобальному оптимуму задачи независимо от выбранной исходной точки.

Классическим примером не сходящегося глобально алгоритма является метод Ньютона. Для иллюстрации этого явления в одномерном случае обратимся к гл. 3, п. 1.1. Поскольку метод Ньютона обладает, в частности, свойствами, интересными с точки зрения скорости сходимости (см. гл. 4, п. 2.8), то он часто комбинируется с другими методами для обеспечения глобальной сходимости.

Теперь мы покажем, что условия, обеспечивающие свойство глобальной сходимости, могут быть выражены в рамках очень общего формализма, и они сводятся, по существу, к тому, что от отобра-

жения A требуется выполнение условия замкнутости, непосредственно обобщающего понятие непрерывности.

Отображения A, используемые в большинстве алгоритмов, зависят от функции f и часто от ее производных (градиент, возможно, гессиан), поэтому эти условия часто возвращают нас к требованию некоторой регулярности от функции f и/или ее производных.

4.4. Замкнутые многозначные отображения. Здесь мы по всей общности рассмотрим многозначные отображения между двумя топологическими векторными пространствами X и Y. В большинстве случаев $X = Y = \mathbf{R}^n$, но X может быть бесконечномерным векторным пространством, например банаховым пространством (полное нормированное векторное пространство) или гильбертовым пространством (это пространство является частным случаем банахова пространства); относительно изучения бесконечномерных задач оптимизации см. гл. 10.

Для многозначных отображений введем понятие замкнутости, обобщающее понятие непрерывности для обычных отображений.

Определение 21. Многозначное отображение $A\colon X \xrightarrow{m} Y$ называется *замкнутым в точке* X, если

$$\left.\begin{array}{l} x^k \to x \quad \text{в} \quad X \\ y^k \to y \quad \text{в} \quad Y \\ \forall k \quad y^k \in A(x^k) \end{array}\right\} \Rightarrow y \in A(x).$$

Многозначное отображение A называется замкнутым на $S \subset X$, если оно замкнуто в любой точке из S.

Замечание. Если A — обычное непрерывное отображение $X \to Y$, то

$$x^k \to x \ (\text{в } X) \Rightarrow A(x^k) \to A(x) \quad (\text{в } Y),$$

и, следовательно, непрерывность влечет замкнутость. Обратное, вообще говоря, неверно.

Определение 22. Если A и B — два многозначных отображения:

$$(A\colon X \xrightarrow{m} Y \quad \text{и} \quad B\colon Y \xrightarrow{m} Z),$$

то *произведение отображений* A, B (обозначаемое $B \circ A$) есть многозначное отображение $C\colon X \xrightarrow{m} Z$, которое элементу $x \in X$ ставит в соответствие

$$C(x) = \bigcup_{y \in A(x)} B(y).$$

Многие из алгоритмов, которые мы будем изучать, могут рассматриваться как произведение нескольких многозначных отображений. Приводимые ниже результаты дают достаточные условия для того, чтобы произведение отображений было замкнуто.

Предложение 2. *Пусть* $A\colon X \xrightarrow{m} Y$ *и* $B\colon Y \xrightarrow{m} Z$ — *два многозначных отображения. Предположим, что выполняются условия*:

1) A *замкнуто в* $x \in X$, B *замкнуто на* $A(x)$;

2) *Y есть замкнутое множество, и любая последовательность $\{y^k\}$, удовлетворяющая условию $y^k \in A(x^k)$ при $x_k \to x$, имеет сходящуюся подпоследовательность (в частности, условие 2) выполняется, если Y — компактное множество).*

Тогда многозначное отображение $C = B \circ A$ замкнуто в x.

Доказательство. Предположим, что $x^k \to x$ в X и $z^k \in C(x^k)$, $z^k \to z$ в Z. Покажем, что $z \in C(x)$.

Поскольку $z^k \in C(x^k)$, то для любого k существует такое $y^k \in A(x^k)$ (значит, $y^k \in Y$), что $z^k \in B(y^k)$. Согласно условию 2) из последовательности $\{y^k\}$ можно выбрать подпоследовательность, сходящуюся к $y \in Y$.

Пусть $\{y^l\}_{l \in L}$ — эта подпоследовательность. Значит, $y^l \to y$ в Y и $\forall l \quad y^l \in A(x^l)$. Отсюда в силу замкнутости отображения A в x выводим, что $y \in A(x)$.

Кроме того, $y^l \to y$ в Y, $z^l \to z$ в Z, а поскольку B замкнуто на $A(x)$, в частности, в y, то имеем $z \in B(y)$. Но тогда $y \in A(x)$ и $z \in B(y) \Rightarrow z \in C(x)$, откуда и следует результат.

Тот факт, что условие 2) автоматически выполняется, если Y — компактное множество, вытекает из определения 6. п. 2.3.

Многозначное отображение может быть также получено как произведение обычного отображения и многозначного отображения. Приводимый ниже результат уточняет для этого случая условия, при которых произведение отображений замкнуто.

Предложение 3. *Пусть $A\colon X \to Y$ — обычное отображение и $B\colon Y \xrightarrow{m} Z$ — многозначное отображение. Тогда если A непрерывно в точке x и B замкнуто в $A(x)$, то произведение $C = B \circ A$ есть многозначное отображение, замкнутое в x.*

Доказательство следует непосредственно из предложения 2, ибо если A непрерывно в точке x и если $x^k \to x$, то последовательность $y^k = A(x^k)$ сходится к $y = A(x)$.

4.5. Теорема о глобальной сходимости. Рассмотрим задачу оптимизации на X, и пусть Ω есть множество точек из X, удовлетворяющих некоторому необходимому условию оптимальности (например, стационарность в случае оптимизации без ограничений — см. гл. 4, п. 1.1).

Предположим, что для решения этой задачи используется алгоритм, представленный многозначным отображением $A\colon X \xrightarrow{m} X$.

Приводимый ниже результат показывает, что глобальная сходимость алгоритма (т. е. сходимость к точке из Ω для любой исходной точки x^0) существенно зависит:

— от замкнутости отображения A во всех точках из $X - \Omega$;

— от существования непрерывной функции $z\colon X \to \mathbf{R}$, называемой функцией спуска.

Определение 23. Говорят, что $z\colon X \to \mathbf{R}$ есть *функция спуска* (относительно алгоритма A), если она непрерывна и удовлетворяет следующим условиям:

1) $x \notin \Omega \Rightarrow z(y) < z(x) \quad \forall y \in A(x)$,

2) $x \in \Omega \Rightarrow z(y) \leqslant z(x) \quad \forall y \in A(x)$.

Теперь можно сформулировать следующую теорему.

Теорема 10 (Зангвилл, 1969). *Пусть дана задача оптимизации на X и пусть Ω — множество точек, которые удовлетворяют некоторому необходимому условию оптимальности.*

Пусть A: $X \xrightarrow{m} X$ — многозначное отображение (алгоритм) и пусть последовательность $\{x^k\}$ порождена алгоритмом, т. е. удовлетворяет условию $x^{k+1} \in A(x^k)$.

Пусть выполняются три условия:

C1) *все точки x^k содержатся в компактном множестве $K \subset X$;*

C2) *существует функция спуска z;*

C3) *многозначное отображение A замкнуто на $X - \Omega$ и $A(x) \neq \varnothing$ $\forall x \in X - \Omega$.*

Тогда предел x любой сходящейся подпоследовательности последовательности $\{x^k\}$ принадлежит Ω.

Доказательство. Все точки x^k содержатся в компактном множестве $K \subset X$, поэтому можно выбрать из $\{x^k\}_{k \in N}$ сходящуюся подпоследовательность. Пусть $\{x^l\}_{l \in L}$ — такая подпоследовательность и x — ее предел ($x \in K$).

В силу непрерывности функции z имеем $z(x^l) \to z(x)$ для $l \to \infty$ ($l \in L$).

Покажем, что $z(x^k) \to z(x)$ $(k \to \infty,\ k \in N)$.

Для любого заданного $\varepsilon > 0$ найдется такое l_ε, что

$$\forall l \geqslant l_\varepsilon \quad (l \in L), \quad z(x^l) - z(x) < \varepsilon.$$

Значит, для любого $k \geqslant l_\varepsilon$ $(k \in N)$ имеем

$$z(x^k) - z(x) = z(x^k) - z\left(x^{l_\varepsilon}\right) + z\left(x^{l_\varepsilon}\right) - z(x).$$

В силу монотонности z имеем $z(x^k) - z\left(x^{l_\varepsilon}\right) \leqslant 0$. Стало быть, для $k \geqslant l_\varepsilon$

$$z(x^k) - z(x) \leqslant z\left(x^{l_\varepsilon}\right) - z(x) < \varepsilon.$$

Покажем, наконец, что $x \in \Omega$. Для этого рассмотрим последовательность $\{x^{l+1}\}_{l \in L}$. Все ее элементы принадлежат компактному множеству K, поэтому можно выбрать сходящуюся подпоследовательность

$$\{x^{l+1}\}_{l \in L' \subset L}, \qquad x^{l+1} \to x'.$$

Значит, для $l \to \infty$ и $l \in L'$ имеем

$$x^l \to x, \quad x^{l+1} \to x', \quad x^{l+1} \in A(x^l).$$

При $x \notin \Omega$ отображение A замкнуто в x, откуда $x' \in A(x)$. А поскольку $z(x^k) \to z(x)$ для $k \to \infty$, $k \in N$, то, в частности, $z(x^{l+1}) \to z(x') = z(x)$ для $l \to \infty$, $l \in L'$. Тогда строгое неравенство $z(y) < z(x)$ не выполняется для $y = x' \in A(x)$, что противоречит предположению, что z — функция спуска.

Следовательно, $x \in \Omega$.

Как мы увидим, этот весьма общий результат позволяет установить глобальную сходимость большинства алгоритмов, которые

будут изучаться в книге, и, в частности, в гл. 4—6. (Относительно обобщений этой теоремы см., например, [3, 27].)

Сделаем несколько замечаний, касающихся практического применения теоремы 10.

1. Функция спуска z будет, как правило, приниматься равной функции f, которую мы пытаемся минимизировать. Стало быть, чаще всего для установления сходимости алгоритмов мы будем требовать непрерывности функции f.

2. Для выполнения условия C1) и принимая во внимание выбор функции f как функции спуска, очень часто будем налагать условие, что множество

$$X_\alpha = \{x | x \in X,\ f(x) \leqslant \alpha = f(x^0)\}$$

ограничено. Тем самым обеспечивается условие, что все элементы последовательности $\{x^k\}$, порожденной алгоритмом, содержатся в замкнутом ограниченном множестве K.

Если рассматриваемое пространство есть $\mathbf{R}^n$, то K — компактное множество.

Другой способ удовлетворить условию C1) заключается в требовании, чтобы функция f удовлетворяла условию $f(x) \to \infty$ при $\|x\| \to \infty$.

3. В противовес условиям C1) и C2), которые обычно удовлетворяются (во всех методах спуска целевая функция в процессе итераций монотонно убывает), условие C3) является критическим. Отсутствие сходимости алгоритма чаще всего проистекает из того, что это условие не выполняется.

Пример. Приведем сразу же очень простой пример [31].

Рассмотрим для $x \in X \in [0, 1]$ многозначное отображение

$$A(x) = \begin{cases} [0, x) & \text{для } 0 < x \leqslant 1, \\ \{0\} & \text{для } x = 0. \end{cases}$$

Для $\Omega = \{0\}$ функция $z(x) = x$ есть функция спуска, ибо для $x \neq 0$

$$z(y) < z(x) = x \quad \forall y \in A(x).$$

Последовательность

$$x^0 = 1, \quad x^{k+1} = x^k - 1/2^{k+2}$$

при любом k удовлетворяет условию

$$x^{k+1} \in A(x^k),$$

но $x^k \to 1/2 \notin \Omega$. Стало быть, x^k не сходится в Ω, и многозначное отображение A не замкнуто на $X - \Omega$.

Можно указать другие примеры этого явления, в частности, это будет сделано в гл. 5, § 3, в связи с методами возможных направлений.

4.6. Асимптотическая сходимость. Скорость сходимости. Если глобальная сходимость установлена, то нас интересует оценка ее эффективности. С практической точки зрения эффективность алго-

ритма зависит от числа итераций, необходимых для получения приближения оптимума x^* с заданной точностью ε. Если сравнить между собой большое количество алгоритмов и допустить, что время вычисления итераций одинаково для всех алгоритмов, то наилучшим среди них будет тот, который требует наименьшего числа итераций.

К сожалению, оказывается невозможным сформировать общие правила такого рода сравнений. В соответствии с выбранной нами отправной точкой природа оптимизируемой функции, значение выбранной точности, иерархия алгоритмов могут сильно различаться.

Если мы хотим получить критерий с некоторым абсолютным значением, то следует прибегнуть к другому типу анализа, взяв за объект исследования *асимптотическую сходимость*, т. е. поведение последовательности $\{x^k\}$ в окрестности предельной точки x^*.

Это приводит к тому, что каждому алгоритму приписывается некоторый индекс эффективности, называемый *скоростью сходимости*. Здесь мы приведем основные определения, которые будут широко использоваться в дальнейшем.

Пусть евклидова норма в $\mathbf{R}^n$ обозначается $\|\cdot\|$ и пусть последовательность $\{x^k\}$ сходится к x^*.

Если выполняется неравенство

$$\limsup \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = \alpha < 1,$$

то говорят, что имеет место линейная сходимость и что α — соответствующий коэффициент сходимости.

Если

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \to 0 \quad \text{при} \quad k \to \infty,$$

то говорят, что имеет место суперлинейная сходимость. Более точно, если существует такое $\gamma > 1$, что

$$\limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^\gamma} < +\infty,$$

то говорят, что имеет место сходимость порядка $\gamma > 1$. В частности, если

$$\limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < +\infty,$$

то говорят, что имеет место квадратичная сходимость (сходимость порядка 2).

Как мы увидим ниже, исследование скорости сходимости алгоритма позволяет оценить его эффективность и осуществить его сравнение (в некотором смысле) с другими алгоритмами. Однако при этом необходима определенная осторожность и следует воздержаться от поспешных выводов.

Например, нельзя утверждать, что алгоритм A, сходящийся в смысле квадратичной сходимости, должен быть существенно лучше алгоритма B, сходящегося линейно.

Предположим, что квадратичная сходимость алгоритма A получена лишь в малой окрестности точки x^*:

$$\mathscr{V} = \{x \mid \|x - x^*\| \leqslant 10^{-2}\|x^0 - x^*\|\},$$

и что требуется $k = 100$ итераций для достижения этой окрестности из точки x^0; и пусть, кроме того, алгоритм B сходится линейно с первой итерации с коэффициентом сходимости $\alpha = 0{,}9$. Если заданная точность есть $\varepsilon_1 = 10^{-3}\|x^0 - x^*\|$, то для алгоритма B понадобится только около $k = 60$ итераций, тогда как для алгоритма A необходимо более 100 итераций.

Если же заданная точность равна $\varepsilon_2 = 10^{-9}\|x^0 - x^*\|$, то очевидно, что алгоритм A потребует значительно меньше 100 итераций, зато алгоритм B будет сходиться лишь приблизительно за $3 \times 60 = 180$ итераций.

Приведенный пример хорошо иллюстрирует предосторожности, которые следует соблюдать при сравнении алгоритмов по критерию скорости сходимости. Следует подчеркнуть также, что этот критерий тем лучше отражает практическую эффективность сравнения алгоритмов, чем больше конечная требуемая точность.

Замечание. Иногда к выражению скорости сходимости последовательности $\{x^k\}$ приходят при исследовании не способа сходимости $\|x^k - x^*\|$ к 0, а способа сходимости последовательности $f(x^k)$ к $f(x^*)$, где f — минимизируемая функция.

Рассмотрим сначала наиболее часто встречающийся частный случай, когда функция f имеет положительно определенный гессиан $\nabla^2 f(x^*)$ в точке оптимума x^*. Стало быть, в окрестности точки x^*

$$f(x) \simeq f(x^*) + (x - x^*)^{\mathrm{T}} \cdot \nabla^2 f(x^*)\,(x - x^*),$$

и если обозначить через λ и Λ соответственно наименьшее и наибольшее собственное значение гессиана $\nabla^2 f(x^*)$, то найдется такая окрестность $\mathscr{V}$ точки x^*, что для любого $x \in \mathscr{V}$

$$\lambda' \|x - x^*\|^2 \leqslant |f(x) - f(x^*)| \leqslant \Lambda' \|x - x^*\|^2, \qquad (1)$$

где $\lambda' = \lambda - \varepsilon > 0$, $\Lambda' = \Lambda + \varepsilon$ $(\varepsilon > 0)$.

При этих условиях легко показать, что сходимость последовательности x^k и x линейна (соответственно более высокого порядка, порядка λ) в смысле абсолютного значения $|f(x^k) - f(x^*)|$, в том и только том случае, если сходимость x^k к x^* в смысле нормы $\|x^k - x^*\|$ линейна (соответственно более высокого порядка, порядка λ).

Исследуем, например, случай линейной сходимости. Предположим, что

$$\limsup_{k \to \infty} \frac{|f(x^{k+1}) - f(x^*)|}{|f(x^k) - f(x^*)|} = \alpha < 1.$$

Для любого α' $(\alpha < \alpha' < 1)$ (например, $\alpha' = (1+\alpha)/2$) найдется такое K, что для $k \geqslant K$

$$\frac{|f(x^{k+1}) - f(x^*)|}{|f(x^k) - f(x^*)|} \leqslant \alpha' < 1,$$

и, значит, для любого $p \geqslant 1$

$$|f(x^{K+p}) - f(x^*)| \leqslant \mu(\alpha')^p, \quad \mu = |f(x^K) - f(x^*)|.$$

С другой стороны, всегда можно предположить, что K выбрано столь большим, что $k \geqslant K = x^k \in \mathscr{V}$. Отсюда для $p \geqslant 1$ имеем

$$\|x^{K+p} - x^*\|^2 \leqslant \frac{1}{\lambda} |f(x^{K+p}) - f(x^*)| \leqslant \frac{\mu}{\lambda} (\alpha')^p,$$

откуда следует линейная сходимость x^k к x^* в смысле сходимости по норме $\|x^k - x^*\|$.

Обратное доказывается аналогично. Результаты, относящиеся к сходимости более высокого порядка и к сходимости порядка λ, устанавливаются точно так же.

Теперь мы рассмотрим проблему во всей общности (т. е. при отсутствии соотношения типа (1)); ясно, что не существует полной эквивалентности между двумя понятиями скорости сходимости.

Например, легко представить себе последовательность $\{x^k\}$, для которой норма $\|x^k - x^*\|$ линейно сходится к 0, но последовательность $\{f(x^k)\}$ даже не является монотонно убывающей (один из таких случаев представляют алгоритмы субградиентов, изучаемые в гл. 4, § 3). Обратно, $f(x^k) - f(x^*)$ могут линейно сходиться к 0, когда расстояние $\|x^k - x^*\|$ не является монотонным.

Однако можно показать, что благодаря понятию средней скорости сходимости в большинстве случаев оценка скорости сходимости по норме $\|x^k - x^*\|$ равносильна оценке относительно разности $|f(x^k) - f(x^*)|$ (см., например, [31]).

СПИСОК ЛИТЕРАТУРЫ *)

1. Abadie J. (1967), *Nonlinear Programming*, North Holland, Amsterdam.
2. Abadie J. (1970), *Integer and Nonlinear Programming*, North Holland, Amsterdam.
3. Adhigama S. T., Polak E., Klessig R. (1979), A comparative study of several general convergence conditions for algorithms modeled by point-to-set-maps, in: *Point-to-set Maps and Mathematical Programming*, Mathematical Programming Study 10 (P. Huard ed.), North Holland, p. 172—190.
4. Aoki M. (1971), *Introduction to optimization techniques*, MacMillan, New York.
5. Auslender A. (1976), *Optimisation-Méthodes numériques*, Masson, Paris.
6. Avriel M. (1976), *Nonlinear Programming: Analysis and Methods*, Prentice Hall.
7. Balakrishnan A. V. (1972), *Techniques of optimization*, Academic Press, New York.

*) Знак * при номере означает, что соответствующий источник имеется на русском языке.— *Примеч. пер.*

8. Bazaraa M. S., Shetty C. M. (1979), *Nonlinear Programming. Theory and Algorithms*, Wiley, New York.
9. Bellman R. (1963), *Mathematical Optimization Techniques*, University of California Press, Berkeley.
10. Beltrami E. J. (1970), *An Algorithmic Approach to Nonlinear Analysis and Optimization*, Academic Press, New York.
11. Berge C. (1966), *Espaces topologiques. Fonctions multivoques.* Dunod, Paris.
12. Berge C., Ghouila-Houri A. (1962), *Programmes, jeux et réseaux de transport*, Dunod, Paris.
13. Beveridge G., Schechter R. (1970), *Optimization: Theory and Practice*, McGraw-Hill, New York.
14. Bradley S. P., Hax A. C., Magnanti T. L. (1977), *Applied Mathematical Programming*, Addison Wesley.
15. Dano S. (1975), *Nonlinear and Dynamic programming*, Springer Verlag.
16*. Dantzig G. B. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton.
17. Dixon L. C. W. (1972), *Nonlinear Optimization*, The English Universities Press, London.
18. Fletcher R. (1969), *Optimization*, Academic Press, London.
19. Gottfred B. S., Weisman J. (1973), *Introduction to Optimization Theory*, Prentice Hall.
20. Grünbaum B. (1967), *Convex Polytopes*, Wiley, New York.
21. Hadley G. (1964), *Nonlinear and Dynamic programming*, Addison Wesley.
22. Hestenes M. R. (1975), *Optimization Theory: The Finite Dimensional case*, Wiley, New York.
23*. Himmelblau D. M. (1972), *Applied Nonlinear Programming*, McGraw-Hill, New York.
24. Hogan W. W. (1973), Point-to-Set-Maps in Mathematical Programming, *S. I. A. M. Review*, 15, p. 591-603.
25. Huard P. (1972), *Optimisation dans R^n (Programmation Mathématique)* (2 parties), Cours de DEA Traitement de l'Information, Université Lille.
26. Huard P. (1975), Optimization Algorithms and Point-to-Set maps, *Mathematical Programming* 8, p. 308-331.
27. Huard P. (1979), Extensions of Zangwill's theorem, in: *Point-to-set Maps*, Mathematical programming Study 10 (P. Huard ed.), North Holland, p. 98-103.
28*. Karlin S. (1959), *Mathematical Methods and Theory in Games, Programming, and Economics*, Addison Wesley, Vol. 1 et 2.
29. Kuhn H. W. (1976), Nonlinear Programming: A Historical View, in *Nonlinear Programming* (Cottle R. W. & Lemke C. E. eds.), American Mathematical Society, Providence.
30. Kunzi H. P., Krelle W., Oettli W. (1966), *Nonlinear Programming*, Blaisdell, Amsterdam.
31. Luenberger D. G. (1973), *Introduction to linear and non linear programming*, Addison-Wesley.
32. Mangasarian O. L. (1969), *Nonlinear Programming*, McGraw-Hill, New York.
33. Martos B. (1975), *Nonlinear Programming: Theory and Methods*, American Elsevier, New York.
34. McMillan C. Jh. (1970), *Mathematical Programming*, Wiley, New York.
35. Neustadt L. W. (1974), *Optimization*, Princeton University Press, Princeton.
36. Pierre D. A. (1969), *Optimization Theory with Applications*, Wiley, New York.
37*. Polak E. (1971), *Computational methods in Optimization*, Academic Press, New York.
38*. Rockafellar R. T. (1970), *Convex analysis*, Princeton University Press, Princeton.
39. Rockafellar R. T. (1981), *The Theory of Subgradients and its Applications to Problems of Optimization. Convex and Nonconvex Functions*, Heldermann Verlag, Berlin.

40. Shapiro J. F. (1979), *Mathematical Programming; Structures and Algorithms*, Wiley, New York.
41. Stoer J., Witzgall C. (1970), *Convexity and Optimization in Finite Dimensions I*, Springer-Verlag.
42. Vajda S. (1961), *Mathematical Programming*, Addison Wesley.
43. Vajda S. (1974), *Theory of Linear and Nonlinear Programming*, Longman, London.
44. Walsh G. R. (1975), *Methods of Optimization*, Wiley, New York.
45. Wilde D. J., Beightler C. S. (1967), *Foundations of Optimization*, Prentice Hall, Englewood Cliffs, N. J.
46. Wolfe P. (1970), Convergence Theory in Nonlinear programming, in *Integer and Nonlinear programming* (Abadie J. ed.), North Holland.
47*. Zangwill W. I. (1969), *Nonlinear programming: an unified approach*, Prentice Hall.
48. Zoutendijk G. (1976), *Mathematical Programming Methods*, North Holland.

ГЛАВА 2
ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

§ 1. Основные определения и результаты

Задача линейного программирования состоит в минимизации (или максимизации) линейной функции при линейных ограничениях; следовательно, речь идет о математической задаче:

найти $\min f(x)$ при условиях:

$$\left.\begin{aligned} &g_i(x) = 0, \quad i \in I^0 \qquad \text{(ограничения типа равенства);}\\ &\left.\begin{aligned} &g_i(x) \leqslant 0, \quad i \in I^-\\ &g_i(x) \geqslant 0, \quad i \in I^+ \end{aligned}\right\} \text{(ограничения типа неравенства);}\\ &x = (x_1, \ldots, x_n)^T \geqslant 0, \end{aligned}\right\} \qquad (P_0)$$

где f, g_i $(i \in I = I^0 \cup I^+ \cup I^-)$ — линейные (аффинные) функции от переменных $x_1, \ldots, x_n$.

Замечание. Без ограничения общности можем предполагать, что x_j $(j = 1, \ldots, n)$ могут быть только неотрицательными. В самом деле, если существует переменная x_j, которая может принимать любое действительное значение — положительное, отрицательное или нуль, то можно заменить x_j разностью $x_j^+ - x_j^-$ двух переменных, принимающих лишь неотрицательные значения.

Если имеется несколько переменных с неопределенным знаком, то мы осуществим замену столько раз, сколько нужно; после этого задача, очевидно, сведется к линейной задаче вида (P_0).

1.1. Стандартная форма линейной задачи. Говорят, что линейная задача имеет стандартную форму, если все ее ограничения имеют форму равенств (кроме ограничений неотрицательности).

Любую линейную задачу всегда можно представить в стандартной форме, введя дополнительные переменные, называемые переменными скачков.

Пример.

$$\left.\begin{aligned} &\min(5x_1 - 3x_2),\\ &x_1 - x_2 \geqslant 2,\\ &2x_1 + 3x_2 \leqslant 4,\\ &-x_1 + 6x_2 = 10,\\ &x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{aligned}\right\} \qquad (P_1)$$

Введя переменные скачков $x_2 \geqslant 0$ и $x_4 \geqslant 0$ в первом и втором ограничениях, получим (P_1) в эквивалентной форме:

$$\begin{gathered} \min(5x_1 - 3x_2 + 0 \cdot x_3 + 0 \cdot x_4), \\ x_1 - x_2 - x_3 = 2, \\ 2x_1 + 3x_2 + x_4 = 4, \\ -x_1 + 6x_2 = 10, \\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_4 \geqslant 0; \end{gathered} \tag{P_1'}$$

(P_1') есть стандартная форма задачи (P_1).

На основании сказанного выше мы будем рассматривать в дальнейшем лишь линейные задачи в стандартной форме типа

$$\begin{gathered} z = c \cdot x \to \min, \\ A \cdot x = b, \\ x \geqslant 0. \end{gathered} \tag{P}$$

Здесь:

n — число переменных;

m — число ограничений;

A — действительная $m \times n$-матрица (матрица ограничений);

$c = (c_1, c_2, \ldots, c_n)$ — вектор-строка;

$b = (b_1, b_2, \ldots, b_m)^T$ — вектор правой части;

$z = c \cdot x = \sum_{j=1}^{n} c_j \cdot x_j$ — минимизируемая функция (целевая функция или экономическая функция).

Заметим, что мы всегда можем предположить $\operatorname{rang}(A) = m$. В самом деле, если $\operatorname{rang}(A) < m$, то одна или несколько строк матрицы A могут быть представлены в виде линейной комбинации остальных. В зависимости от значений коэффициентов b_i соответствующие ограничения окажутся излишними (и в этом случае они могут быть исключены из задачи) либо не согласованными с остальными (в этом случае система $Ax = b$ не имеет решения).

1.2. Решения линейной задачи и выпуклые полиэдры. Обозначим через $X = \{x | Ax = b,\ x \geqslant 0\}$ множество решений задачи (P). Заметим, что X — пересечение аффинного многообразия

$$\{x | Ax = b,\ x \in \mathbf{R}^n\}$$

с положительным ортантом — есть выпуклое множество.

Определение 1. Выпуклое множество вида $X = \{x | Ax = b, x \geqslant 0\}$ называется *выпуклым политопом.*

Ограниченный выпуклый политоп называется *выпуклым полиэдром* или *выпуклым многогранником.*

Как мы увидим ниже, некоторые точки в множестве решений линейной задачи будут играть специальную роль.

Определение 2. *Крайней точкой* выпуклого политопа или полиэдра X называется точка $x \in X$, которая не может быть выражена в виде выпуклой комбинации других точек $y \in X$ $(y \neq x)$.

Понятия выпуклого политопа и крайней точки иллюстрируются на рис. 1.

Теперь для алгебраического описания понятий крайних точек выпуклого политопа и выпуклого полиэдра мы введем понятия базиса и базисного решения.

1.3. Базисы, реализуемые базисы, базисные решения.

Определение 3. Базисом называется всякая регулярная квадратная подматрица порядка m матрицы A (существует по крайней мере одна такая матрица, поскольку $rang\, A = m$).

Пусть B — некоторый базис. Тогда перестановкой столбцов матрицы A всегда можно привести A к виду $A = [B, N]$, где N — подматрица, состоящая из столбцов матрицы A, не принадлежащих

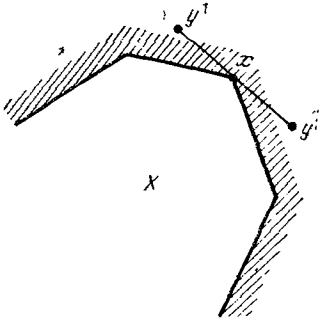


Рис. 1. Выпуклый политоп в R^2. Точка x, которая не может быть представлена как выпуклая комбинация других точек X, есть крайняя (экстремальная) точка

базису. Точно так же возможно разбиение вектора x на $[x_B, x_N]^T$, а вектора c — на $[c_B, c_N]$ (см. рис. 2).

Всякое решение задачи (P) удовлетворяет условию $Ax = b$, и, следовательно,

$$Bx_B + Nx_N = b. \qquad (1)$$

Базисным решением (относительно базиса B) называется частное решение задачи (1), полученное при условии $x_N = 0$. Тогда

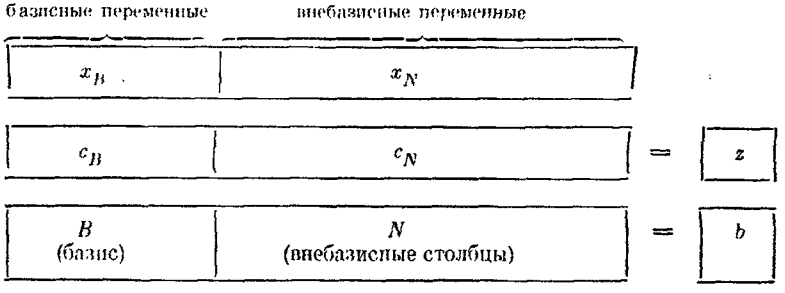


Рис. 2

x_B определяется единственным образом в результате решения системы (*Крамера*)

$$Bx_B = b, \quad \text{или} \quad x_B = B^{-1}b.$$

Базисное решение называется *реализуемым*, если $x_B \geqslant 0$, или, иначе, если

$$B^{-1}b \geqslant 0.$$

Базис, соответствующий реализуемому базисному решению, называется *реализуемым базисом.*

Базисное решение называется *вырожденным*, если вектор $x_B = B^{-1}b$ имеет нулевые компоненты. Вырожденность есть частое явление в некоторых задачах (в задачах переноса, течений, кратчайших путей, ...).

Исследование этого явления носит относительно тонкий характер, и мы его не будем рассматривать. Стало быть, в дальнейшем мы будем предполагать, вообще говоря, что реализуемые базисные решения не являются вырожденными (это используется, в частности, при доказательстве теоремы 4).

1.4. Алгебраическая характеристика крайних точек.

Теорема 1. *Множество крайних точек полигона* $X = \{x \in \mathbf{R}^n | Ax = b,\ x \geqslant 0\}$ *соответствует множеству реализуемых базисных решений.*

Доказательство. а) Если x — реализуемое базисное решение, то x — крайняя точка.

Действительно, имеем $x = (x_1, \ldots, x_m, 0, \ldots, 0)$, и если предположить, что $x = \lambda\alpha + (1-\lambda)\beta$ $(0 < \lambda < 1,\ \alpha \in X,\ \alpha \neq \beta \neq x)$, где

$$\alpha = (\alpha_1, \ldots, \alpha_m, \alpha_{m+1}, \ldots, \alpha_n),$$

$$\beta = (\beta_1, \ldots, \beta_m, \beta_{m+1}, \ldots, \beta_n),$$

то должны выполняться соотношения

$$\lambda\alpha_j + (1-\lambda)\beta_j = 0 \qquad \forall j = m+1, \ldots, n,$$

а поскольку $0 < \lambda < 1$ $(\alpha \geqslant 0,\ \beta \geqslant 0)$, отсюда следует, что

$$\alpha_j = \beta_j = 0 \qquad \forall j = m+1, \ldots, n.$$

Но тогда m первых компонент векторов α, β определяются однозначно при помощи решения системы Крамера $Bx_B = b$, откуда $x = \alpha = \beta$, и мы приходим к противоречию.

b) Если x — крайняя точка, то x есть реализуемое базисное решение.

Определим носитель вектора x (обозначается $\operatorname{supp}(x)$) как множество индексов его ненулевых компонент.

Прежде всего покажем, что если x — крайняя точка в множестве X, то носитель $\operatorname{supp} x$ минимален (относительно включения).

Если носитель $\operatorname{supp}(x)$ не минимален (относительно включения), то существует такое возможное решение y $(y \geqslant 0,\ Ay = b)$, что $\operatorname{supp}(y) \not\subseteq \operatorname{supp}(x)$ (значит, $y \neq x$) и $\operatorname{supp} y$ минимально. И пусть $j \in \operatorname{supp}(y)$ таково, что

$$\frac{x_j}{y_j} = \min_{i \in \operatorname{supp}(y)} \left\{\frac{x_i}{y_i}\right\} = \lambda > 0.$$

Тогда имеем $x - \lambda y \geqslant 0$ и $A(x - \lambda y) = (1-\lambda)b$. Допустим, что $\lambda \geqslant 1$; тогда вектор $x - y$ удовлетворяет равенству

$$A(x - y) = 0.$$

А поскольку $x \neq y$, то $x - y$ есть вектор $x - y = t \geqslant 0$, удовлетворяющий равенству $At = 0$; значит, x может быть представлен в виде

$$x = \frac{1}{2}\left(y + \frac{2}{3}t\right) + \frac{1}{2}\left(y + \frac{4}{3}t\right) = \frac{y^1 + y^2}{2},$$

где $y^1 \neq y^2 \neq x$ и $Ay^1 = Ay^2 = b$ $(y^1 \geqslant 0,\ y^2 \geqslant 0)$. Стало быть, x есть выпуклая комбинация различных решений, и мы приходим к противоречию. Следовательно, $\lambda < 1$.

Вектор $z = (x - \lambda y)/(1 - \lambda)$ удовлетворяет равенству $Az = b$ $(z \geqslant 0)$. С другой стороны, он отличен от x (поскольку его носитель строго включается в носитель вектора x) и отличен от y (ибо $y_i \neq 0$, когда $z_i = 0$). Но тогда x можно представить в виде $x = \lambda y + (1 - \lambda)z$, и, значит, x есть выпуклая комбинация двух различных решений, и мы приходим к противоречию.

Следовательно, носитель вектора x минимален (относительно включения).

Покажем теперь, что $|\mathrm{supp}(x)| \leqslant m$. Если бы $|\mathrm{supp}(x)| > m$, то столбцы матрицы A, соответствующие ненулевым компонентам вектора x, были бы линейно зависимы; значит, найдется такой вектор y, что $\mathrm{supp}(y) \subset \mathrm{supp}(x)$ и $Ay = 0$ (и, стало быть, $y \neq x$). Выберем

$$\lambda = -\min_{i|y_i<0}\{-x_i/y_i\} \quad \text{или} \quad \lambda = -\min_i\{x_i/y_i\}, \qquad y \geqslant 0;$$

допущение, что вектор $x + \lambda y$ есть реализуемое решение $(x + \lambda y \geqslant 0)$ со строгим включением носителя в $\mathrm{supp}(x)$, приводит к противоречию. Значит, $|\mathrm{supp}(x)| \leqslant m$, и вектор x есть реализуемое решение.

Следствие 1. *Выпуклое множество (политоп или полиэдр)*

$$X = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geqslant 0\}$$

имеет конечное число $\nu(X)$ *крайних точек, и*

$$\nu(X) \leqslant C_n^m.$$

Доказательство. Максимальное число базисов, извлеченных из матрицы A, равно C_n^m (число возможностей выбора m столбцов из n), и не все они — обязательно реализуемые базисы. Отсюда следует утверждение.

Следствие 2. *Любая точка выпуклого полиэдра* $x \subset \mathbf{R}^n$ *есть выпуклая комбинация его крайних точек.*

Доказательство. Рассмотрим выпуклый полиэдр (т. е. ограниченный политоп)

$$X = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geqslant 0\}$$

и пусть $x \in X$.

Если x имеет минимальный носитель, то это есть крайняя точка полиэдра X (см. доказательство теоремы 1), и следствие доказано.

Если носитель x не является минимальным, то найдется такой вектор y^1 ($y^1 \geqslant 0$, $Ay = b$) с минимальным носителем, что

$$\operatorname{supp}(y^1) \not\subseteq \operatorname{supp}(x),$$

и, значит, $y^1 \neq x$.

Пусть

$$\lambda_1 = \min_{j \in \operatorname{supp}(y^1)} \{x_j / y_j^1\} > 0.$$

Имеем $|\operatorname{supp}(x - \lambda_1 y^1)| < |\operatorname{supp}(x)|$ и $x - \lambda_1 y^1 \geqslant 0$.

Действуя точно так же относительно $x - \lambda_1 y^1$, найдем вектор y^2 с минимальным носителем ($y^2 \geqslant 0$, $Ay^2 = b$, $y^2 \neq y^1$) и число λ_2 такие, что

$$|\operatorname{supp}(x - \lambda_1 y^1 - \lambda_2 y^2)| < |\operatorname{supp}(x - \lambda_1 y^1)|,$$
$$x - \lambda_1 y^1 - \lambda_2 y^2 \geqslant 0.$$

И так далее. Через конечное число p шагов ($p \leqslant n$) получим p крайних точек $y^1, y^2, \ldots, y^p$ и чисел $\lambda_1, \lambda_2, \ldots, \lambda_p > 0$, удовлетворяющих равенству

$$x - \lambda_1 y^1 - \lambda_2 y^2 - \ldots - \lambda_p y^p = 0.$$

Стало быть, $x = \lambda_1 y^1 + \lambda_2 y^2 + \ldots + \lambda_p y^p$, и $\sum_{i=1}^{p} \lambda_i = 1$, поскольку $Ax = b = (\lambda_1 + \lambda_2 + \ldots + \lambda_p) b$.

Исследуем теперь более подробно случай, когда $X = \{x | Ax = b, x \geqslant 0\}$ есть неограниченный политоп.

Говорят, что вектор $y \geqslant 0$ есть *бесконечный луч*, если для любого $x \in X$ вектор $x + \lambda y$ принадлежит X при любом $\lambda \geqslant 0$.

Для того чтобы вектор $y \geqslant 0$ был бесконечным лучом, необходимо и достаточно, чтобы y был неотрицательным решением системы $Ay = 0$.

С другой стороны, можно показать, что множество $Y = \{y | Ay = 0, y \geqslant 0\}$ бесконечных лучей есть конус.

Рассмотрим гиперплоскость уравнения

$$\sum_{j=1}^{n} y_j = 1.$$

Множество $H \cap Y$ есть ограниченный политоп, а значит, выпуклый полиэдр. Стало быть, он в силу теоремы 1 имеет конечное множество крайних точек $y^1, y^2, \ldots, y^k$.

Поскольку любая точка множества Y получается при помощи гомотетии из точки множества $H \cap Y$, а любая точка из $H \cap Y$ есть выпуклая комбинация точек $y^1, y^2, \ldots, y^k$, из сказанного выше выводим, что всякая точка из Y есть линейная комбинация точек $y^1, y^2, \ldots, y^k$ с положительными или нулевыми коэффициентами. Говорят, что Y есть *выпуклый полиэдральный конус*. Векторы y^1,

$y^2, \ldots, y^k$, составляющие *порождающее* множество для Y, называются экстремальными *или крайними лучами* политопа X.

Из их определения следует, что крайние лучи соответствуют реализуемым базисным решениям системы

$$\begin{aligned} Ay &= 0, \\ \sum_j y_j &= 1, \\ y &\geqslant 0. \end{aligned} \tag{1}$$

Для получения крайних лучей заметим, что система (1) определяется $m+1$ уравнениями.

Значит, достаточно рассмотреть все базисные $(m \times m)$-матрицы, извлеченные из A, и для каждой базисной матрицы B взять все столбцы A^j (j — вне базиса). Всякий раз, когда система $By = -A^j$ имеет неотрицательное решение $\bar{y}$, то вектор $\bar{y}$ определяет крайний луч множества X.

Из сказанного выше и из доказательства следствия 2 получаем

Следствие 3. *Всякая точка выпуклого политопа $X \subset \mathbf{R}^n$ есть выпуклая комбинация крайних точек из X, к которой в случае необходимости добавляется линейная комбинация крайних лучей с положительными коэффициентами.*

1.5. Теорема 2 (оптимальность в крайней точке). *Оптимум линейной функции z на выпуклом полиэдре $X \subset \mathbf{R}^n$ достигается по крайней мере в одной крайней точке. Если он достигается в нескольких крайних точках, то он достигается в любой точке, являющейся выпуклой комбинацией крайних точек.*

Доказательство. Пусть $y^1, y^2, \ldots, y^K$ — крайние точки полиэдра X. Положим $z^* = \min\limits_{k=1,\ldots,K} \{z(y^k)\}$ и покажем, что z^* есть минимум для z на X. На основании следствия 1 любая точка $x \in X$ может быть представлена в виде

$$x = \sum_{k=1}^{K} \lambda_k y^k \qquad (\lambda_k \geqslant 0,\ \textstyle\sum \lambda_k = 1);$$

тогда $z(x) = \sum\limits_{k=1}^{K} \lambda_k z(y^k)$ (линейность функции z), откуда

$$z(x) \geqslant z^* \sum_{k=1}^{K} \lambda_k = z^*.$$

Стало быть, z^* есть минимум функции z на X и достигается по крайней мере в одной крайней точке.

Вторая часть утверждения следует непосредственно из линейности функции z.

Из теорем 1 и 2 выводим, что когда линейная задача имеет оптимум на конечном расстоянии, то существует реализуемый базис B^*, при котором соответствующее базисное решение x^* оптимально. Теперь задача состоит в определении алгоритмической процедуры, позволяющей находить оптимальный базис B^*. Как мы по-

кажем, такая процедура может быть выведена непосредственно из следующего далее результата.

1.6. Характеризация базисов и оптимальных базисных решений.

Теорема 3. *Пусть* m*-вектор-строка* $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ — *имеет вид*

$$\pi = c_B B^{-1}.$$

Для того чтобы матрица B *была оптимальным реализуемым базисом, необходимо и достаточно* *), *чтобы*

$$\bar{c}_N = c_N - \pi N = c_N - c_B B^{-1} N \geqslant 0.$$

Вектор π называется вектором *симплексных множителей*. Компоненты $\bar{c}_j$ вектора $\bar{c}_N$ называются *приведенными значениями*, или *приведенными ценами*, внебазисных переменных.

Достаточность. Пусть $x = [x_B, x_N]$ — любое решение задачи (P) (не обязательно базисное) и $z(x)$ — соответствующее значение функции z. Произведем замену переменных

$$x_B = B^{-1}b - B^{-1}Nx_N;$$

теперь можно выразить z в виде функции только внебазисных переменных:

$$z(x) = c_B B^{-1} b + (c_N - c_B B^{-1} N) x_N = z_B + \bar{c}_N x_N.$$

Поскольку $\bar{c}_N \geqslant 0$ и $x_N \geqslant 0$, то $z(x) \geqslant z_B = c_B B^{-1} b$.

С другой стороны, значение z_B функции z достигается на реализуемом базисном решении

$$x^0 = [x_B^0, x_N^0] = [B^{-1}b, 0].$$

Следовательно, $z_B = z(x^0)$ есть оптимальное значение для (P), а x^0 — оптимальное решение.

Необходимость. Покажем, что если существует внебазисное переменное с индексом s, удовлетворяющее условию $\bar{c}_s < 0$, то тогда можно выбрать решение с ценой, меньшей z_B.

Исходя из точки $x^0 = [x_B^0, x_N^0] = [B^{-1}b, 0]$, рассмотрим замену, в которой переменное x_s, и только оно, среди внебазисных переменных (x_N), изменяет свое значение:

$$x_s = \begin{cases} 0 \text{ в решении } x^0, \\ \theta > 0 \quad \text{в точке } x. \end{cases}$$

Полученная таким образом точка $x = [x_B, x_N]$ удовлетворяет равенству

$$x_N = x_N^0 + \theta e_s = \theta e_s$$

(e_s — вектор той же размерности, что и x_N, причем все его компоненты — нули, кроме s-й, равной 1).

---

*) При отсутствии вырожденности. Стало быть, предполагаем $x_B = B^{-1}b > 0$.

Число θ следует выбрать так, чтобы вектор $x = [x_B, x_N]$ оставался решением, т. е.

$$x_B = B^{-1}b - B^{-1}Nx_N = B^{-1}b - \theta B^{-1}Ne_s,$$
$$x_B \geqslant 0.$$

Заметив, что произведение Ne_s есть не что иное, как столбец A_s матрицы A, соответствующий переменному x_s, и обозначив $\bar{A}_s = B^{-1}A_s$, $\bar{b} = B^{-1}b$, мы должны, стало быть, выбрать θ удовлетворяющим условиям

$$x_B = \bar{b} - \theta\bar{A}_s,$$
$$x_B \geqslant 0.$$

Если $\bar{b} > 0$ (**предположение невырожденности**), то θ всегда можно выбрать достаточно малым для того, чтобы оставалось $x_B \geqslant 0$. Тогда

$$z(x) = z(x^0) + \bar{c}_N x_N = z(x^0) + \theta\bar{c}_N e_s = z(x^0) + \theta\bar{c}_s < z(x^0)$$

и, значит, x^0 не будет оптимальным решением, а B не будет оптимальным базисом.

Следствие 4. *Пусть B — произвольный реализуемый базис, x^0 — соответствующее базисное решение и $\bar{c}_N = c_N - \pi N$.*

Если существует такое внебазисное переменное x_s, что $\bar{c}_s < 0$, то:

а) *либо можно неограниченно увеличивать значение x_s, не выходя из множества реализуемых решений, и в этом случае оптимум функции z не ограничен $(-\infty)$;*

b) *либо берется другой базис $\widehat{B}$ и другое реализуемое базисное решение $\widehat{x}$, удовлетворяющее условию $z(\widehat{x}) < z(x^0)$.*

Доказательство. Как и при доказательстве теоремы 3, рассмотрим для $\theta > 0$ решение $x = [x_B, x_N]$, определяемое равенствами

$$x_N = x_N^0 + \theta e_s,$$
$$x_B = \bar{b} - \theta\bar{A}_s.$$

Обозначим через $\bar{a}_{is}$ $(i = 1, 2, \ldots, m)$ компоненты вектора $\bar{A}_s$. Могут представиться два случая.

Случай 1: $\bar{A}_s \leqslant 0$ (все компоненты столбца $\bar{A}_s = B^{-1}A_s$ отрицательны). Тогда x_s может принимать значение θ сколь угодно много раз, и всегда будет $x_B \geqslant 0$. Поскольку

$$z(x) = z(x^0) + \theta\bar{c}_s, \quad \bar{c}_s < 0,$$

то соответствующее значение функции z может быть сколь угодно мало: задача имеет *неограниченный оптимум* $(-\infty)$.

Случай 2: существует такое i $(1 \leqslant i \leqslant m)$, что $a_{is} > 0$ (не все компоненты столбца $\bar{A}_s$ отрицательны). В этом случае значение величины θ не может возрастать бесконечно: наибольшее зна-

чение $\widehat{\theta}$, которое может принимать величина 0, определяется равенством

$$\widehat{\theta} = \min_{i|\bar{a}_{is}>0} \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} \right\} = \frac{\bar{b}_r}{\bar{a}_{rs}}. \tag{2}$$

Полученное новое решение $\widehat{x}$ имеет компоненты

$$\widehat{x}_N = \widehat{\theta} e_s,$$
$$\widehat{x}_B = \bar{b} - \widehat{\theta} \bar{A}_s.$$

Заметим, что если минимум в (2) единственный (т. е. при отсутствии вырожденности), то это решение имеет в точности m ненулевых компонент. В самом деле, переменное x_s, равное нулю в решении x^0, становится строго положительным; переменное x_r, которое было строго положительным, имеет теперь значение

$$\widehat{x}_r = \bar{b}_r - \theta a_{rs} = 0.$$

Стало быть, новое решение $\widehat{x}$ является базисным. Оно соответствует базису $\widehat{B}$, полученному из B заменой столбца r столбцом s.

(Базисы B и $\widehat{B}$ называют *сопряженными*; они соответствуют *сопряженным крайним точкам* множества решений X.)

С другой стороны, в силу неравенства $\bar{c}_s < 0$ имеем $z(\widehat{x}) = z(x^0) + \widehat{\theta} c_s < z(x^0)$.

Следствие 4 представляет интерес тем, что из него непосредственно вытекает один из методов решения линейной задачи: *прямой* (или *исходный*) *симплекс-алгоритм.*

§ 2. Решение линейных задач

2.1. Прямой симплекс-алгоритм (пересмотренная форма). Предположим, что мы имеем реализуемый исходный базис B^0. Перечислим последовательно шаги алгоритма:

a) B^0 — исходный реализуемый базис. Итерация $k = 0$.

b) $k \leftarrow k + 1$.

c) Пусть на k-й итерации B — текущий базис, $x = [x_B, x_N]$ — соответствующее базисное решение. Вычислить:

$$\bar{b} = B^{-1} b \qquad \text{(значения переменных базиса)},$$
$$\bar{u} = c_B B^{-1} \qquad \text{(симплексные множители)},$$
$$\bar{c}_N = c_N - \pi N \qquad \text{(приведенные значения)}.$$

d) Если $c_N \geqslant 0$, СТОП: оптимум достигнут.

Если существует такое s, что $\bar{c}_s < 0$, то

e) Пусть A_s — s-й столбец матрицы A.

Вычислить $\bar{A}_s = B^{-1} A_s$.

Если $\bar{a}_{is} \leqslant 0 \quad \forall i = 1, 2, \ldots, m$, то СТОП: неограниченный оптимум $(-\infty)$. Иначе вычислить

$$\hat{x}_s = \frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{i|\bar{a}_{is}>0} \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} \right\}.$$

f) Пусть x_t — переменное, соответствующее r-й строке базиса, т. е. такое, что $B^{-1}A_t = e_r$ (m-вектор, все компоненты которого нули, кроме r-й, равной $+1$); тогда s-е переменное принимает значение $\hat{x}_s > 0$ (возвращение в базис); t-е переменное обращается в нуль $(\hat{x}_t = 0)$ (выход из базиса); новое текущее решение $\hat{x}$ соответствует новому реализуемому базису: $\hat{B} = B + \{A_s\} - \{A_t\}$. Вычислить базис $\hat{B}^{-1}$, обратный к новому, и возвратиться к b).

Геометрически процедура интерпретируется как переход от крайней точки до сопряженной крайней точки вдоль границы множества X (решений задачи).

Алгебраически алгоритм интерпретируется как определение последовательности сопряженных базисов $B^0, B^1, \ldots, B^q, \ldots$ и базисных решений $x^0, x^1, \ldots, x^q, \ldots$, удовлетворяющих условиям $z(x^0) > z(x^1) > \ldots > z(x^q) \ldots$

Замечание 1. Если есть несколько таких переменных x_s, что $\bar{c}_s < 0$, и если вырожденность отсутствует, то выбор в (d) безразличен. Хороший эвристический критерий — выбирать переменное x_s с наименьшей приведенной ценой $\bar{c}_s$. Проблемы, порожденные вырожденностью, изучаются в 2.3.

Замечание 2. Если матрица базиса B состоит из первых m столбцов матрицы A, взятых в естественном порядке, то $B^{-1}A = [I, B^{-1}N] = [I, \bar{N}]$, и в этом случае переменное, которое выходит из базиса в f), есть переменное x_r $(t = r)$. Однако при замене последовательных базисов это, вообще говоря, не имеет места; тогда следует установить соответствие между порядковым номером переменного в базисе и его индексом.

2.2. Теорема 4 (конечная сходимость). *При условии невырожденности прямой симплекс-алгоритм сходится за конечное число итераций.*

Доказательство. Достаточно заметить, что существует конечное число крайних точек (см. следствие 1 теоремы 1) и что строгое убывание функции z запрещает дважды проходить через одну и ту же крайнюю точку.

2.3. Проблемы, порожденные вырожденностью. В вырожденном случае, когда $\bar{b}_r = 0$, имеем

$$\hat{x}_s = \bar{b}_r / \bar{a}_{rs} = 0.$$

Тогда значение функции z после замены базиса не изменится (действительно, в этом случае $z(\hat{x}) = z_B + \bar{c}_s \hat{x}_s = z_B$).

Таким образом, в результате некоторого числа замен базисов мы можем прийти к базису, уже встречавшемся ранее, и создать бесконечную «зацикленность». Предотвратить такое зацикливание

можно разными способами. К ним относятся, в частности, следующие.

1. Бесконечно малые возмущения исходных данных задачи.

2. Применение лексикографической процедуры: располагают базисы в лексикографическом порядке, индуцируемом расположением столбцов. Каждый раз, как появляется вырожденность, производят замены базиса, соответствующие лексикографически возрастающим последовательностям базисов. Таким образом, никогда не появится базис, уже встречавшийся ранее, и это обеспечивает ситуацию, когда после конечного числа итераций функция z снова начнет строго убывать.

Практически лексикографическая процедура — весьма дорогостоящая для вычислений, и большинство машинных кодов линейного программирования не предохраняют против зацикливания. Однако оно встречается достаточно редко.

3. Правила выбора поворота. Рассмотрим, например, очень простое правило, предложенное в [6, 7].

Для каждой итерации симплекс-алгоритма:

— среди всех переменных s, пригодных для вхождения в базис (т. е. таких, что $\bar{c}_s < 0$), выбрать переменное с наименьшим индексом;

— среди всех переменных, пригодных к выходу из базиса (т. е. всех переменных x, для которых $\bar{b}_r/\bar{a}_{rs} = \min\limits_{i|\bar{a}_{is}>0} \{\bar{b}_i/\bar{a}_{is}\}$), выбрать переменное с наименьшим индексом.

Можно показать (см. [6, 7]), что даже в вырожденном случае это правило обеспечивает конечную сходимость симплекс-метода. Опыт показывает, однако, что его применение существенно снижает преимущества «естественного» метода, состоящего, в соответствии с замечанием 1 п. 2.1, в выборе в качестве входящего переменного минимального значения [2].

2.4. Алгоритмическая сложность и практическая эффективность симплекс-метода. Оценка сложности алгоритма состоит в исследовании максимального числа элементарных операций, необходимых в наиболее неблагоприятных случаях.

Но, как показано в [17], можно сформулировать задачи, для которых симплекс-метод требует исследования возрастающего по экспоненте числа крайних точек как функции размера задачи (числа переменных и ограничений). Стало быть, сложность симплекс-алгоритма имеет экспоненциальный характер.

Однако если симплекс-алгоритм показывает плохой результат в самых неблагоприятных случаях, то в среднем он оказывается весьма эффективным в большинстве практических задач. В самом деле, статистика свидетельствует (см. [11]), что если он применяется в соответствии с «естественным» правилом (см. замечание 2 п. 2.1), то в среднем требуется от m до $3m$ итераций (где m — число ограничений). Таким образом, число исследуемых крайних точек в общем случае много меньше C_n^m, которое имеет порядок числа крайних точек.

Начиная с работ Хачияна [16], известно, что линейные задачи могут быть решены за время, полиномиально зависящее от размера задачи. Полиномиальный алгоритм, изучавшийся в [16] (вариант метода дилатации пространства — метода Шора; см. гл. 4, п. 3.12), составляет предмет многих числовых исследований, но, несмотря на некоторое улучшение в деталях, представляется в настоящее время еще очень далеким от возможности конкурировать с симплекс-методом. За более полным изложением алгоритма Хачияна — Шора можно обратиться к гл. 4, п. 3.12 и к [1, 13].

2.5. Матрицы замены базиса в форме обратного произведения. Поскольку базисы B и $\widehat{B}$ — сопряженные (т. е. отличаются лишь одним столбцом), легко вычислить $\widehat{B}^{-1}$ исходя из B^{-1}. Для этого достаточно предварительно умножить обратную к прежнему базису матрицу B^{-1} на матрицу (называемую *заменой базиса* или *матрицей замены базиса*)

$$\eta = \begin{bmatrix} 1 & & -\frac{\bar{a}_{1s}}{\bar{a}_{rs}} & & \\ & 1 & \vdots & & \\ & & \frac{1}{\bar{a}_{rs}} & & 0 \\ & & \vdots & & \\ & 0 & -\frac{\bar{a}_{ms}}{\bar{a}_{rs}} & & 1 \end{bmatrix}.$$

Ясно, что η состоит из единичной $m \times m$-матрицы, в которой r-й столбец заменен на m-вектор с координатами

$$-\bar{a}_{is}/\bar{a}_{rs}, \quad i \neq r,$$
$$1/\bar{a}_{rs} \text{ в } r\text{-й компоненте.}$$

Если исходный базис представляет собой **единичную** матрицу, то после некоторого числа замен базиса

$$B^0 \to B^1 \to \ldots \to B^q,$$

соответствующих матрицам замены базиса $\eta^1, \eta^2, \ldots, \eta^q$, имеем

$$\eta^q \times \eta^{q-1} \times \ldots \times \eta^2 \times \eta^1 = [B^q]^{-1}.$$

Практически получение в явном виде матрицы $[B^q]^{-1}$ не представляет интереса (из соображений перегрузки памяти вычислительного устройства): достаточно зафиксировать последовательность $\eta^1, \eta^2, \ldots, \eta^q$.

Для сохранения в памяти матрицы η^t достаточно знать индекс r и ненулевые члены r-го столбца. Это есть *форма обратного произведения.*

2.6. Исходный базис. В принципе можно в алгоритме п. 2.1 взять какой угодно реализуемый исходный базис. На практике не

всегда бывает просто указать такой реализуемый базис (например, как при ограничениях типа равенства), поэтому всегда пытаются получить единичную матрицу, при необходимости преобразуя исходную задачу введением дополнительных переменных: *переменных скачка* или *искусственных переменных*.

Пример 1 (случай ограничений типа равенства).

$$\begin{gathered}\min z_1 = x_1 - 2x_2 + 2x_3,\\ x_1 + x_2 - x_3 = 3,\\ -x_1 + 3x_2 = -4,\\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0.\end{gathered} \tag{P_1}$$

Добавим два переменных y_1, y_2, называемых *искусственными*, с очень большим коэффициентом M и будем решать новую задачу:

$$\begin{gathered}\min z_1' = x_1 - 2x_2 + 2x_3 + My_1 + My_2,\\ x_1 + x_2 - x_3 + y_1 = 3,\\ -x_1 + 3x_2 - y_2 = -4,\\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad y_1 \geqslant 0, \quad y_2 \geqslant 0,\end{gathered} \tag{P_1'}$$

где исходный базис переменных (y_1, y_2) является реализуемым (решение в соответствующем базисе есть $y_1 = 3$, $y_2 = 4$, $x_1 = 0$, $x_2 = 0$, $x_3 = 0$) и соответствует (с точностью до знака) единичной матрице. Добавление к искусственным переменным y_1, y_2 большого положительного множителя гарантирует оптимальность решения при $y_1 = y_2 = 0$ (если (P_1) имеет решения), и тогда оптимальное решение задачи (P_1') будет оптимальным решением задачи (P_1).

Пример 2 (случай ограничений типа $\leqslant$).

$$\begin{gathered}\min z_2 = x_1 - 2x_2,\\ x_1 + x_2 \leqslant 3,\\ -x_1 + 3x_2 \leqslant -4,\\ x_1 \geqslant 0, \; x_2 \geqslant 0.\end{gathered} \tag{P_2}$$

Добавим в первом ограничении переменное скачка $s_1 \geqslant 0$, во втором — переменное скачка s_2 и искусственное переменное y_2. Припишем y_2 множитель M ($M > 0$ велико) и решим новую задачу:

$$\begin{gathered}\min z_2' = x_1 - 2x_2 + My_2,\\ x_1 + x_2 + s_1 = 3,\\ -x_1 + 3x_2 + s_2 - y_2 = -4,\\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad s_1 \geqslant 0, \quad s_2 \geqslant 0, \quad y_2 \geqslant 0,\end{gathered} \tag{P_2'}$$

где исходный базис переменных (s_1, y_2) является реализуемым (соответствующее базисное решение имеет вид $s_1 = 3$, $y_2 = 4$, $x_1 = 0$, $x_2 = 0$, $s_2 = 0$) и соответствует (с точностью до знака) единичной матрице. Как и в примере 1, в случае, если (P_2) имеет решения, оптимумы для (P_2) и (P_2') совпадают, ибо y_2 с большим

положительным множителем M дает при $y_2 = 0$ оптимум (минимум).

2.7. Каноническая форма и симплекс-таблица.

Определение 4. Говорят, что линейная задача представлена в канонической форме относительно базисных переменных $(x_{i1}, x_{i2}, \ldots, x_{im})$, если:

— z выражается в виде функции внебазисных переменных (рассматриваемых как независимые);

— столбцы матрицы ограничений, соответствующие базисным переменным, образуют (с точностью до перестановки) единичную матрицу.

Ниже мы увидим, как можно получить каноническую форму линейной задачи относительно произвольного базиса и к какому представлению можно при этом прийти.

Рассмотрим в самом общем виде следующую линейную задачу (в стандартной форме):

$$\min z = c_1x_1 + \ldots + c_nx_n,$$
$$a_{i1}x_1 + \ldots + a_{in}x_n = b_i \quad (i = 1, \ldots, m),$$
$$x \geqslant 0,$$

или, в эквивалентной записи,

$$\min z,$$
$$c_1x_1 + \ldots + c_nx_n - z = 0,$$
$$a_{11}x_1 + \ldots + a_{1n}x_n = b_1,$$
$$a_{21}x_1 + \ldots + a_{2n}x_n = b_2,$$
$$\ldots\ldots\ldots\ldots$$
$$a_{m1}x_1 + \ldots + a_{mn}x_n = b_m,$$
$$x_1, \ldots, x_n \geqslant 0.$$

Ее можно записать в виде таблицы размера $(m+1)\times(n+2)$, **называемой** *симплекс-таблицей*:

x_1	x_2	...	x_n	z	↓ правые части
c_1	c_2	...	c_n	-1	0
a_{11}	a_{12}	...	a_{1n}	0	b_1
a_{21}				0	b_2
...	....	....	....	....	...
a_{m1}	a_{m2}	...	a_{mn}	0	b_m

(Двойная вертикальная черта расположена на месте знака равенства левой и правой части.)

Предположим для упрощения, что базисными переменными являются $x_1, \ldots, x_m$.

Умножив предыдущую таблицу на регулярную $(m+1)\times(m+1)$-матрицу

1	$-\pi = -c_B B^{-1}$
0 0 0	B^{-1}

получим каноническую форму относительно базисных переменных:

Базисные переменные			Внебазисные переменные		
x_1	$x_2 \cdots$	x_m	$x_{m+1}\ x \ldots\ x_n$	z	
0 0 ... 0			$\bar{C}_N = C_N - \pi N$	-1	$-z_B$
1			$\bar{N} = B^{-1}N$	0 ⋮ 0	$\bar{b} = B^{-1}b$
	1	0			
	0	1			

Симплекс-таблица представляет интерес тем, что она в конденсированной форме собирает воедино все элементы, необходимые для осуществления симплекс-алгоритма. Действительно:

— базисное решение получается при прямом чтении: поскольку $x_N = 0$, то

$x_1 = \bar{b}_1,\ x_2 = \bar{b}_2,\ \ldots,\ x_m = \bar{b}_n$ (для базисных переменных),

причем

$\bar{b}_1 \geqslant 0,\ \bar{b}_2 \geqslant 0,\ \ldots,\ \bar{b}_m \geqslant 0$, если базис допустимый (реализуемый);

— значение z_B целевой функции содержится (со знаком минус) в правой верхней клетке.

В самом деле, первая строка записывается в виде

$$\bar{c}_N x_N - z = -c_B B^{-1} b,$$

и в силу $x_N = 0$ имеем

$$z_B = c_B B^{-1} b = [c_B c_N] \begin{bmatrix} \bar{b} \\ 0 \end{bmatrix}.$$

— наконец, приведенные значения $c_N = (\bar{c}_j)_{j=m+1,\ldots,n}$ внебазисных переменных получаются прямым чтением первой строки симплекс-таблицы. Они позволяют, в частности, сразу же увидеть, будет ли текущее базисное решение оптимальным (это происходит, когда $\bar{c}_j \geqslant 0$ для всех внебазисных s).

Замечание. Для простоты предположим, что базисными будут m первых переменных $x_1, \ldots, x_m$. Если базисные переменные расположить в произвольном порядке, их легко идентифицировать при помощи того факта, что соответствующие столбцы являются с точностью до перестановки столбцами единичной матрицы.

В результате замены базиса, в которой переменное x_s ($\bar{c}_s < 0$) заменяется на переменное x_r, каноническая форма, соответствующая новому базису, получается посредством перемножения симплекс-таблицы и матрицы, где η — матрица замены базиса, введенная в п. 2.5:

$$\begin{array}{|c|c|}\hline 1 & 0 \ldots 0 - \dfrac{\bar{c}_s}{\bar{a}_{rs}} \; 0 \ldots 0 \\ \hline 0 & \begin{array}{ccccccc} 1 & & -\dfrac{\bar{a}_{1s}}{\bar{a}_{rs}} & & & & \\ & 1 & \vdots & & 0 & & \\ & & 1 & & & & \\ & & \dfrac{}{\bar{a}_{rs}} & & & & \\ & 0 & \vdots & 1 & & & \\ & & \dfrac{\bar{a}_{ms}}{\bar{a}_{rs}} & & \ddots & & \\ & & & & & 1 & \end{array} \\ \hline \end{array} = \begin{array}{|c|c|}\hline 1 & 0 \ldots 0 - \dfrac{\bar{c}_s}{\bar{a}_{rs}} \; 0 \ldots 0 \\ \hline 0 & \eta \\ \hline \end{array}$$

Элемент $\bar{a}_{rs}$ часто называют разрешающим элементом, r-я строка называется разрешающей строкой, а s-й столбец — разрешающим столбцом.

2.8. **Пример.** Приведем небольшой пример использования симплекс-таблицы для решения линейных задач.

Пусть имеется задача

$$\begin{aligned} &\max x_1 + 2x_2, \\ &-3x_1 + 2x_2 \leqslant 2, \\ &-x_1 + 2x_2 \leqslant 4, \\ &x_1 + x_2 \leqslant 5, \\ &x_1 \geqslant 0,\ x_2 \geqslant 0. \end{aligned}$$

Она может быть представлена как задача на минимизацию, достаточно изменить знак при коэффициентах целевой функции, и в стандартной форме (с ограничения типа равенства), если ввести переменные скачка $x_3, x_4, x_5 \geqslant 0$. Получаем задачу

$$\begin{aligned} &\min z = -x_1 - 2x_2, \\ &-3x_1 + 2x_2 + x_3 = 2, \\ &-x_1 + 2x_2 + x_4 = 4, \\ &x_1 + x_2 + x_5 = 5, \\ &x_1,\ x_2,\ x_3,\ x_4,\ x_5 \geqslant 0. \end{aligned}$$

Она может быть представлена в виде следующей симплекс-таблицы:

Таблица 1

x_1	x_2	x_3	x_4	x_5	z		
−1	−2	0	0	0	−1	0	(L1)
−3	2	1	0	0	0	2	(L2)
−1	2	0	1	0	0	4	(L3)
1	1	0	0	1	0	5	(L4)

внебазисные переменные — базисные переменные

Заметим, что речь идет о канонической форме относительно базиса переменных скачка (x_3, x_4, x_5).

Поскольку не все коэффициенты (приведенные значения) внебазисных переменных в первой строке неотрицательны, то этот базис не оптимален. Стало быть, мы сейчас осуществим перемещение посредством замены базиса.

Выберем в качестве переменного, входящего в новый базис (переменное x_s), внебазисное переменное, имеющее строго отрицательное приведенное значение — тем самым переменное x_2; x_2 может принимать значение $\widehat{\theta}$, являющееся максимальным значением величины θ, удовлетворяющей условиям

$$2\theta + x_3 = 2,$$

$$2\theta + x_4 = 4,$$

$$\theta + x_5 = 5,$$

$$x_3, \quad x_4, \quad x_5 \geqslant 0.$$

Получаем $\widehat{\theta} = \min\{2/2;\ 4/2;\ 5/1\} = 1$.

Таким образом, переменное x_3 обращается в нуль и покидает базис. Чтобы получить новую каноническую форму относительно базиса из переменных (x_2, x_4, x_5), достаточно осуществить со строками (L1), (L2), (L3), (L4) следующие элементарные действия:

$$(\text{L2})' = \frac{1}{2}(\text{L2}),$$

$$(\text{L1})' = (\text{L1}) - (\text{L2}),$$

$$(\text{L3})' = (\text{L3}) - (\text{L2}),$$

$$(\text{L4})' = (\text{L4}) - \frac{1}{2}(\text{L2}).$$

Отсюда получаем табл. 2:

Таблица 2

x_1	x_2	x_3	x_4	x_5	z		
−4	0	1	0	0	−1	2	(L1)′
−3/2	1	1/2	0	0	0	1	(L2)′
2	0	−1	1	0	0	2	(L3)′
5/2	0	−1/2	0	1	0	4	(L4)′
	↑		↑	↑			

базисные переменные

Заметим, что к тому же результату мы пришли бы, перемножив табл. 1 и матрицу замены базиса:

1	−1	0	0
0	1/2	0	0
0	−1	1	0
0	−1/2	0	1

Прямое чтение табл. 2 показывает, что соответствующее новое базисное решение имеет вид

$$x_2 = 1, \quad x_4 = 2, \quad x_5 = 4,$$

$$x_1 = 0, \quad x_3 = 0$$

при значении $z = -2$.

Это решение не оптимально, так как переменное x_1 имеет строго отрицательное приведенное значение (−4); стало быть, x_1 вернется в базис и примет значение $\hat{\theta}$ — максимальное значение величины θ, удовлетворяющей условиям

$$-30/2 + x_2 = 1,$$

$$20 + x_4 = 2,$$

$$50/2 + x_5 = 4,$$

$$x_2, \quad x_4, \quad x_5 \geqslant 0;$$

получаем $\hat{\theta} = \min\{2/2;\ 8/5\} = 1$.

Значит, переменное x_4 обращается в нуль и покидает базис. Тогда новая каноническая форма относительно базиса из переменных (x_1, x_2, x_5) получается в результате осуществления над

строками (L1), (L2), (L3), (L4) следующих операций:

$$(L3)'' = \frac{1}{2}(L3)',$$

$$(L1)'' = (L1)' - 2(L3)',$$

$$(L2)'' = (L2)' + \frac{3}{4}(L3)',$$

$$(L4)'' = (L4)' - \frac{5}{4}(L3)'.$$

Имеем

Таблица 3

x_1	x_2	x_3	x_4	x_5	z		
0	0	−1	2	0	−1	6	(L1)″
0	1	−1/4	3/4	0	0	5/2	(L2)″
1	0	−1/2	1/2	0	0	1	(L3)″
0	0	3/4	−5/4	1	0	3/2	(L4)″

↑ ↑ ↑

базисные переменные

Соответствующее базисное решение получаем прямым чтением таблицы:

$$x_1 = 1, \quad x_2 = 5/2, \quad x_5 = 3/2, \quad x_3 = 0, \quad x_4 = 0;$$

его значение равно −6.

Это решение не оптимально, ибо переменное x_3 имеет строго отрицательное приведенное значение (−3). Значит, прибегаем к новой замене базиса, и переменное x_3 возвращается в базис. Значение, принимаемое x_3, будет равно (3/2)/(3/4) = 2, и переменное x_5 покинет базис.

Новая каноническая форма относительно базиса $\{x_1, x_2, x_3\}$ получится в результате осуществления над строками табл. 3 следующих операций:

$$(L4)''' = \frac{4}{3}(L4)'',$$

$$(L1)''' = (L1)'' + \frac{4}{3}(L4)'',$$

$$(L2)''' = (L2)'' + \frac{1}{3}(L4)'',$$

$$(L3)''' = (L3)'' + \frac{2}{3}(L4)''.$$

Таблица 4

x_1	x_2	x_3	x_4	x_5	z		
0	0	0	1/3	4/3	−1	8	(L1)‴
0	1	0	1/3	1/3	0	3	(L2)‴
1	0	0	−1/3	2/3	0	2	(L3)‴
0	0	1	−5/3	4/3	0	2	(L4)‴

Полученное новое базисное решение имеет вид

$$x_1 = 2, \quad x_2 = 3, \quad x_3 = 2, \quad x_4 = 0, \quad x_5 = 0;$$

его значение равно −8. А поскольку приведенные значения всех переменных неотрицательны, то это решение оптимально, и алгоритм заканчивается.

§ 3. Понятие двойственности

Двойственность есть фундаментальное понятие в линейном программировании, приводящее к важному результату теоретического и практического характера: теореме двойственности.

3.1. Двойственная задача к линейной задаче в стандартной форме. Рассмотрим линейную задачу (в стандартной форме)

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ x &\geqslant 0. \end{aligned} \qquad (P)$$

Каждому i-му ($i = 1, \ldots, m$) ограничению поставим в соответствие переменное u_i, положительное, отрицательное или нуль (называемое двойственным переменным), и рассмотрим линейную задачу

$$\begin{aligned} \max w &= ub, \\ uA &\leqslant c, \end{aligned} \qquad (D)$$

где u есть m-я вектор-строка $(u_1, u_2, \ldots, u_m)$.

Линейная задача (D) тесно связана с линейной задачей (P). В самом деле, легко видеть, что:

— матрица ограничений задачи (D) есть транспонированная матрица задачи (P);

— вектор цен для задачи (P) есть вектор правых частей задачи (D) и наоборот.

Задача (D) называется *двойственной* к задаче (P).

В противоположность двойственной задача (P) называется *прямой*, или *исходной*.

3.2. Определение двойственности в общем случае. Очевидно, что можно определить двойственность для любой линейной задачи (не обязательно в стандартной форме). Приведенная ниже таблица соответствий между исходной и двойственной задачами по-

зволяет записать непосредственно двойственную задачу для любой линейной задачи.

Прямая (исходная)	Двойственная
Целевая функция (min)	Правая часть
Правая часть	Целевая функция (max)
A — матрица ограничений	A^T — матрица ограничений
i-е ограничение: $\geqslant$	Переменное $u_i \geqslant 0$
i-е ограничение: $=$	Переменное $u_i \gtrless 0$
Переменное $x_j \geqslant 0$	j-е ограничение: $\leqslant$
Переменное $x_j \gtrless 0$	j-е ограничение: $=$

Заметим, в частности, что двойственная к двойственной задаче совпадает с исходной.

Пример.

$$\begin{aligned} &\min 2x_1 - 3x_2, \\ &x_1 - x_2 \leqslant 1, \\ &2x_1 + 3x_2 \geqslant 4, \qquad \text{(исходная)} \\ &x_1 + x_2 = 3, \\ &x_1 \geqslant 0, \quad x_2 \gtrless 0; \end{aligned}$$

$$\begin{aligned} &\max u_1 + 4u_2 + 3u_3, \\ &u_1 + 2u_2 + u_3 \leqslant 2, \\ &-u_1 + 3u_2 + u_3 = -3, \qquad \text{(двойственная)} \\ &u_1 \leqslant 0, \quad u_2 \geqslant 0, \quad u_3 \gtrless 0. \end{aligned}$$

3.3. Теорема двойственности. Без ограничения общности можем предположить, что прямая задача (P) берется в стандартной форме. Тогда двойственная задача (D) берется в форме, указанной в п. 3.1.

Лемма 1. *Если $\bar{x}$, $\bar{u}$ — соответственно решения произвольной исходной и двойственной задач, то*

$$\bar{z} = c\bar{x} \geqslant \bar{w} = ub.$$

Доказательство. Имеем

$$A\bar{x} = b \Rightarrow \bar{u}A\bar{x} = \bar{u}b,$$

а поскольку

$$\bar{x} \geqslant 0, \quad \bar{u}A \leqslant c, \quad \bar{u}A\bar{x} = \bar{u}b \leqslant c\bar{x},$$

то лемма доказана.

Отсюда сразу вытекает

Следствие 5. *Если x^*, u^* — соответственно решения исходной и двойственной задач, удовлетворяющие равенству $cx^* = u^*b$, то x^* есть оптимальное решение прямой, а u^* — оптимальное решение двойственной задачи.*

Лемма 2. *Пусть* (P) *имеет конечный оптимум и пусть* π^* — *симплекс-множители, соответствующие оптимальному решению* x^* *задачи* (P). *Тогда* π^* *есть решение двойственной задачи и* $cx^* = \pi^* b$ (*и, стало быть,* π^* *есть оптимальное решение двойственной задачи*).

Доказательство. Согласно теореме 3, если π^* — симплекс-множители, соответствующие оптимальному решению x^* задачи (P), то

$$\bar{c}_j = c_j - \pi^* A_j \geqslant 0 \quad \text{для внебазисных } x_j,$$

и

$$\bar{c}_j = c_j - \pi^* A_j = 0 \quad \text{для базисных } x_j.$$

Следовательно, $\pi^* A \leqslant c$ и π^* есть решение двойственной задачи (D).

С другой стороны, пусть B — оптимальный базис задачи (P), соответствующий решению x^*. Имеем $\pi^* = c_B B^{-1}$, и, значит,

$$\pi^* = c_B B^{-1},$$

и, следовательно,

$$\pi^* b = c_B B^{-1} b = cx^*.$$

В силу следствия 5 π^* есть оптимальное решение двойственной задачи (D).

Лемма 2 соответствует случаю, когда обе задачи (P) и (D) имеют решения. Следующая теорема охватывает множество всех возможных случаев.

Теорема 5 (теорема двойственности). *Пусть заданы линейная задача* (P) *и отвечающая ей двойственная задача* (D).

a) *Если* (P) *и* (D) *имеют решения, то каждая из этих задач имеет оптимальное решение и*

$$z^* = \min(P) = \max(D) = w^*.$$

b) *Если одна из них имеет неограниченный оптимум, то другая не имеет решения.*

Доказательство. Пункт a) следует из леммы 2 (в самом деле, можно свести все к случаю, когда задача (P) представлена в стандартной форме). Для доказательства пункта b) предположим, что, например, задача (D) неограничена. Это значит, что всегда для любого достаточно большого M найдется такое двойственное решение u, что $ub > M$. Тогда если бы (P) имела решение $\bar{x}$, то в силу леммы 1 выполнялось бы неравенство $ub \leqslant c\bar{x}$ для любого двойственного решения, что приводит к противоречию.

Теорема 6. *Два решения* $(\bar{x}, \bar{u})$ *соответственно прямой и двойственной задач оптимальны в том и только том случае, если*

$$(\bar{u}A_j - c_j)x_j = 0 \quad \forall j = 1, \ldots, n,$$

где A_j — j-*й столбец матрицы* A.

Доказательство. Имеем

$$A\bar{x} = b \Rightarrow \bar{u}A\bar{x} - c\bar{x} = \bar{u}b - c\bar{x}.$$

Поэтому если $(\bar{x}, \bar{u})$ — пара оптимальных решений, то $c\bar{x} = \bar{u}b$, откуда $(\bar{u}A - c)\bar{x} = 0$.

Поскольку $\bar{x} \geqslant 0$, $\bar{u}A - c \leqslant 0$, то равенство нулю скалярного произведения влечет для каждого j равенство

$$(\bar{u}A_j - c_j)\bar{x}_j = 0.$$

Обратно, если $(\bar{u}A - c)\bar{x} = 0$, то в силу следствия 5 имеем $c\bar{x} = \bar{u}$, и, значит, пара $(\bar{x}, \bar{u})$ оптимальна.

Теорема дополнительности 6 допускает еще следующую форму.

Если линейная задача имеет ограничения типа неравенств (например, двойственная к (P) задача (D)), тогда в точке a оптимума:

— двойственное переменное, соответствующее ограничению, не выполняющемуся в виде равенства, обращается в нуль;

— строго положительному двойственному переменному соответствует ограничение, выполняющееся в виде равенства.

3.4. Двойственные переменные и теневые цены. Задача (P) из п. 1.1 может рассматриваться как представитель семейства линейных задач, параметризованных правой частью:

$$\min cx, \qquad (P(b))$$
$$Ax = b, \quad x \geqslant 0.$$

Теперь мы исследуем изменение оптимального значения $z(b)$ для задачи $(P(b))$ как функцию правой части b.

Пусть для некоторого фиксированного значения в правой части через B обозначен оптимальный базис задачи $(P(b))$ и пусть $u = c_B B^{-1}$ — оптимальные двойственные переменные. Имеем $\bar{c}_j = -uA_j \geqslant 0$ для внебазисного j.

Предположим теперь, что b меняется и что снова рассматривается линейная задача

$$\min cx, \qquad (P(b'))$$
$$Ax = b', \quad x \geqslant 0, \quad b' = b + \Delta b.$$

Заметим, что $u = c_B B^{-1}$ не зависит от b и всегда остается двойственным решением.

При этих условиях, если новый вектор b' удовлетворяет неравенству $B^{-1}b \geqslant 0$, то базис B остается одновременно прямым и двойственным реализуемым базисом, а значит, оптимальным. Следовательно, базис B остается оптимальным для всех Δb, удовлетворяющих неравенству

$$B^{-1}(b + \Delta b) \geqslant 0.$$

При отсутствии вырожденности имеем $B^{-1}b > 0$ и, стало быть, найдется такое $\varepsilon > 0$, что предыдущее условие выполняется, как только $\|\Delta b\| < \varepsilon$. Таким образом, для достаточно малых $\|\Delta b\|$ имеем

$$z(b') = ub' = ub + u\Delta b.$$

Отсюда следует, что $\frac{\partial z(b)}{\partial b_i} = u_i$.

Это приводит к интерпретации двойственного переменного u_i как значения единичной вариации $\Delta b_i = 1$ правой части i-го ограничения (теневая цена).

§ 4. Двойственные и исходно-двойственные алгоритмы

Понятие двойственности, введенное в предыдущем параграфе, позволяет определить другие методы решения линейных задач.

4.1. Двойственный алгоритм. Он состоит в применении прямого алгоритма из § 2 к двойственной задаче, причем операции замены базиса (повороты) осуществляются на прямой таблице.

Предположим, что мы отыскиваем решение линейной задачи (в стандартной форме)

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ x &\geqslant 0. \end{aligned} \qquad (P) \quad (3)$$

Соответствующая двойственная задача имеет вид

$$\max w = ub, \qquad (D)$$

$$uA \leqslant c, \ u \quad \text{произвольного знака.}$$

Пусть B — двойственно-реализуемый базис задачи (3) (регулярная квадратная подматрица матрицы A). Иными словами, пусть соответствующее двойственное базисное решение $u = c_B B^{-1}$ удовлетворяет условию $uN \leqslant c_N$.

Если B — также исходно-реализуемый базис, т. е. если $x_B = B^{-1}b \geqslant 0$, то по теореме двойственности u есть двойственное оптимальное решение и $x = [x_B, x_N] = [B^{-1}b, 0]$ есть прямое оптимальное решение. В противном случае вектор $x = [x_B, x_N]$ не будет прямым решением, т. е.

$$x_B = \bar{b} = B^{-1}b \not\geqslant 0.$$

Если допустить, что матрица B состоит из m первых столбцов матрицы A, то симплекс-таблица двойственной задачи запишется в виде табл. 5:

Таблица 5

$b_1\ b_2\ \ldots\ b_m$	$0\quad 0\quad \ldots\quad 0$		0 ← $-w$
$a_{11}\ \ldots\ a_{m1}$ B^T $a_{1m}\ \ldots\ a_{mm}$	$+1$ $\quad +1$ $\quad\quad \ddots$ $\quad\quad\quad +1$	0	c_1 $\vdots$ c_m } c_B
$a_{1m+1}\ldots a_{m,m+1}$ N^T $a_{1n}\ \ldots\ a_{mn}$	0	$+1$ $\quad +1$ $\quad\quad \ddots$ $\quad\quad\quad +1$	c_{m+1} $\vdots$ c_n } c_N
$u_1, u_2, \ldots u_m$	$v_1, v_2\ \ldots\ v_m$	$v_{m+1}\ \ldots\ v_n$	

где введены переменные скачка $v_j \geqslant 0$ $(j = 1, \ldots, n)$: $v_j = c_j - uA_j$ (A_j — j-й столбец матрицы A).

Базисными переменными будут: $u_1, \ldots, u_m, v_{m+1}, \ldots, v_n$.

Умножив эту таблицу спереди на матрицу

$$\begin{array}{c} m \\ n-m \end{array} \left\{ \begin{array}{|c|c|} \hline (B^{-1})^T & 0 \\ \hline -(B^{-1}N)^T & I_{n-m} \\ \hline \end{array} \right. \qquad \begin{array}{cc} m & n-m \end{array}$$

получим канонической форму задачи (D) относительно базиса переменных $(u_1, \ldots, u_m, v_{m+1}, \ldots, v_n)$:

Таблица 6

		$-\bar{b}$		
	$0 \ldots 0$	$-\bar{b}_1, -\bar{b}_2, \ldots, -\bar{b}_m$	$0 \ldots 0$	πb ← w
$1 \ldots m$	I_m	$(B^{-1})^T$	0	$\pi^T = (c_B B^{-1})^T$
поворот $m+1 \ldots n$	0	$-(\bar{N})^T$	I_{n-m}	$(\bar{c}_N)^T$
переменные →	$u_1 \ldots u_m$	$v_1 \ldots v_m$	$v_{m+1} \ldots v_n$	

где $\pi = c_B B^{-1}$ — симплекс-множители и $\bar{c}_N = c_N - \pi N \geqslant 0$ — приведенные значения внебазисных переменных (x_N).

Легко видеть, что целевая функция w также взята в канонической форме (все коэффициенты при базисных переменных равны нулю).

Применение к этой таблице прямого алгоритма из § 2 приводит к следующему правилу замены базиса:

а) если $\bar{b} = B^{-1}b \geqslant 0$, то конец: решение $u = c_B B^{-1}$ есть двойственное оптимальное решение, а решение $[x_B, x_N] = [\bar{b}, 0]$ есть прямое оптимальное решение;

б) если существует такое r, что $\bar{b}_r < 0$, то могут представиться два случая.

Случай 1. В r-й строке матрицы $\bar{N}$ найдется отрицательный член (*ибо в силу произвольности знака переменных и разрешающий элемент принадлежит одному из* $n - m$ последних ограничений двойственной таблицы). Тогда вычисляем

$$\frac{\bar{c}_s}{-\bar{a}_{rs}} = \min_{\substack{j \\ \text{внебазисны}}} \left\{ \frac{\bar{c}_j}{-\bar{a}_{rj}} \right\}_{j|-\bar{a}_{rj}>0}.$$

В двойственной таблице переменное v_r входит в базис, переменное v_s выходит из базиса. В прямой таблице переменное x_s возвращается в базис, а x_r выходит из базиса.

Случай 2. Все члены r-й строки из $\bar{N}$ положительны. Тогда двойственный оптимум неограничен, что указывает (теорема двойственности) на отсутствие прямого решения.

Заметим, что при этом правиле все вычисления могут осуществляться на прямой таблице.

Применение этого алгоритма предполагает знание исходного двойственно-реализуемого базиса. Это часто встречающийся случай; и вот примеры таких ситуаций:

а) когда вычислено оптимальное решение линейной задачи с заданной правой частью b и отыскивается новое оптимальное решение для правых частей b', мало отличающихся от b. Симплекс-множители оптимума первой задачи составляют двойственно-реализуемое базисное решение для второй задачи;

b) когда оптимальное решение задачи (P) определено, но мы хотим получить оптимальное решение после добавления к (P) дополнительных ограничений (это, например, случай алгоритмов сечения в целочисленном программировании, см. гл. 7, § 3).

Двойственный алгоритм сходится в смысле конечной сходимости при условии невырожденности, которое записывается тогда в виде

$$\bar{c}_j > 0 \quad \forall j \qquad \text{(внебазисного)}$$

на каждой итерации.

В вырожденном случае конечная сходимость снова может обеспечиваться за счет использования тех же методов, которые использовались для прямого алгоритма: перестановкой значений цен лексикографического метода, правила выбора поворота (см. п. 2.3).

4.2. Исходно-двойственный алгоритм. Чтобы решить линейную задачу (в стандартной форме)

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ x &\geqslant 0, \end{aligned} \qquad (P) \qquad (3)$$

предположим, что известно решение двойственной задачи, т. е. вектор $u = (u_1, \ldots, u_m)$, удовлетворяющий условию $\tilde{c} = c - uA \geqslant 0$.

Любое решение x задачи (P) удовлетворяет условию $Ax = b$, поэтому очевидно, что x будет оптимальным решением задачи (P) в том и только том случае, если x будет оптимальным решением задачи

$$\begin{aligned} \min \tilde{z} &= \tilde{c}x = (c - uA)x, \\ Ax &= b, \\ x &\geqslant 0. \end{aligned} \qquad (\tilde{P})$$

Это будет так, в частности, в случае решения задачи (P), удовлетворяющего условиям

$$x_j > 0 \Rightarrow c_j - uA_j = 0,$$
$$c_j - uA_j > 0 \Rightarrow x_j = 0,$$

т. е. условиям дополняемости с u (теорема 6).

Итак, требуя от компонент x_j вектора x быть положительными лишь при $\tilde{c}_j = 0$, приходим к условиям дополняемости. Именно на этом принципе основан исходно-двойственный алгоритм, или *прямо-двойственный алгоритм*, который мы сейчас изложим.

1. Пусть произвольно задано двойственное решение u; попытаемся определить решение x задачи (P), удовлетворяющее условию

$$\operatorname{supp}(x) \subset J(u) = \{j \mid \tilde{c}_j = 0\}.$$

Поиск такого решения сводится к решению следующей линейной задачи (прямая ограниченная задача с переменными $x_\mu \in J(u)$):

$$PR(u) \begin{cases} \min \rho = \sum\limits_{i=1}^{m} y_i, \\ Ax + y = b, \\ x \geqslant 0, \\ x_j = 0 \quad \forall j \notin J(u) \\ y \geqslant 0, \end{cases}$$

исходя из допустимого базисного решения $y_i = b_i \ \forall i$ (всегда можно предположить $b_i > 0$; в противном случае достаточно предварительно умножить i-ю строку в системе (3) на -1).

2. Тогда могут представиться два случая:

а) если в оптимуме для $PR(u)$ имеем $\rho = 0$, то $y = 0$, и полученное решение x есть реализуемое базисное решение задачи (P), удовлетворяющее условию $\operatorname{supp}(x) \subset J(u)$. А поскольку для u выполняются условия дополняемости, то (x, u) есть пара оптимальных решений соответственно для прямой и двойственной задач;

b) если в оптимуме для $PR(u)$ снова имеем $\rho > 0$, то это означает, что не существует решения задачи (P), удовлетворяющего условиям дополняемости с u. Тогда заменим двойственные переменные u, чтобы определить новую, ограниченную задачу $PR(u')$, имеющую более широкое множество решений, чем предыдущая.

Обозначив через $\bar{\omega} = (\bar{\omega}_1, \ldots, \bar{\omega}_m)$ переменное задачи, двойственной к $PR(u)$, запишем

$$\max \bar{\omega} b,$$

$$\bar{\omega} A_j \leqslant 0, \quad j \in J(u),$$

$$\bar{\omega} \leqslant 1,$$

$\bar{\omega}$ любого знака.

По условию имеем

$$\min \rho = \rho^* = \bar{\omega}^* b = \max \bar{\omega} b > 0.$$

С другой стороны, оптимальные двойственные переменные $\bar{\omega}^* = (\bar{\omega}_1^*, \bar{\omega}_2^*, \ldots, \bar{\omega}_m^*)$ для задачи $PR(u)$ удовлетворяют условиям до-

полняемости:

$$x_j > 0 \Rightarrow \bar{\omega}^* A_j = 0.$$

Рассмотрим новый вектор u':

$$u' = u + \lambda \bar{\omega}^*.$$

Имеем $\tilde{c}_j = 0$ и $\bar{\omega}^* A_j = 0$ для любого базисного переменного x_j, поэтому $\tilde{c}_j = c_j - uA_j - \lambda \bar{\omega}^* A_j = \tilde{c}_j = 0$ $\forall j$ (базисного), и, значит, условия дополняемости будут выполняться при любом значении λ.

Для $\lambda > 0$ имеем $u'b = ub + \lambda \bar{\omega}^* b > ub$; чтобы уменьшить разрыв между двойственной функцией $u'b$ и прямой функцией cx, представляется интересным способ выбора наибольшего положительного значения λ, согласующегося с ограничениями

$$\tilde{c}'_j = \tilde{c}_j - \lambda \bar{\omega}^* A_j \geqslant 0 \quad \forall j,$$

чем установлено, что u' остается двойственным решением.

Для $j \in J(u)$ имеем $\tilde{c}_j = 0$, но $\omega^* A_j \leqslant 0$, и, значит, соответствующие ограничения не лимитируют значение λ.

Если при любом $j \notin J(u)$ выполняется неравенство

$$\bar{\omega}^* A_j \leqslant 0,$$

то λ не ограничено сверху, и, значит, двойственная функция ub не ограничена сверху; отсюда выводим, что задача (P) не имеет решения (теорема двойственности), и алгоритм заканчивается.

Если существует такое $j \notin J(u)$, что $\bar{\omega}^* A_j > 0$, то максимальное значение λ равно

$$\lambda^* = \min_{j \mid \bar{\omega}^* A_j > 0} \left\{ \frac{\tilde{c}_j}{\bar{\omega}^* A_j} \right\}.$$

Заменив u на $u' = u + \lambda^* \bar{\omega}^*$, получим $\tilde{c}'_j = 0$ по крайней мере для одного $j \notin J(u)$.

Для решения новой ограниченной задачи $PR(u')$, соответствующей новому двойственному решению

$$u' = u + \lambda^* \bar{\omega}^*,$$

вернемся к 1.

При отсутствии вырожденности (т. е. в предположении $\tilde{c}_j \neq 0$ $\forall j$ (внебазисных)) число λ^*, получаемое на каждой итерации, положительно, значит, двойственная функция строго возрастает и один и тот же базис никогда не встречается дважды. Отсюда получаем конечную сходимость метода.

В вырожденном случае используем те же методы (возмущение значений цен, лексикографическую процедуру), которые использовались для прямого и двойственного алгоритмов.

Заметим, наконец, что на любом этапе алгоритма оптимальное решение ограниченной задачи $PR(u)$ является допустимым базисным решением ограниченной задачи $PR(u')$, полученной после модификации двойственных переменных: стало быть, представляет интерес его использование в качестве начального решения для задачи $PR(u')$ в прямом симплекс-алгоритме.

СПИСОК ЛИТЕРАТУРЫ

1. Aspvall B., Stone R. E. (1980), Khachiyan's Linear Programming Algorithm, *Journal of Algorithms*, 1, p. 1-13.
2. Avis D., Chvatal V. (1978), Notes on Bland's pivoting rule, *Mathematical Programming*, 8, p. 24-34.
3. Bartels R. H., Golub G. H. (1969), The Simplex Method of Linear Programming using L. U. Decomposition, *Comm. A. C. M.*, 12, 5, p. 266-268.
4. Bazaraa M. S., Jarvis J. J. (1977), *Linear Programming and Network Flows*, Wiley, New York.
5. Berge C., Ghouila-Houri A. (1962), *Programmes, jeux et réseaux de transport*, Dunod, Paris.
6. Bland R. G. (1977a), New finite pivoting rules for the simplex method, *Math. Ops. Res.*, 2, p. 103-107.
7. Bland R. G. (1977b), A Combinatorial Abstraction of Linear Programming, *Journal Combinatorial Theory*, B-23, p. 33-57.
8. Charnes A. (1952), Optimality and Degeneracy in Linear Programming *Econometrica*, 20, p. 160-170.
9. Charnes A., Cooper W. W. (1961), *Management Models and Industrial Applications of Linear Programming* (2 vol.), Wiley, New York.
10. Cooper L., Steinberg D. I. (1974), *Methods and Applications of Linear Programming*, W. B. Sanders Co., Philadelphia.
11*. Dantzig G. B. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton New Jersey.
12. Denel J., Fiorot J. C., Huard P. (1981), The Steepest-Ascent Method for the Linear Programming problem, *R. A. I. R. O. Analyse Numérique*, vol. 15, n° 3, p. 195-200.
13. Gacs P., Lovasz L. (1981), Khachiyan's Algorithm for Linear Programming, *Mathematical Programming Study*, 14, p. 61-68.
14*. Gass S. I. (1969), *Linear programming*, McGraw-Hill, New York.
15. Hadley G. (1962), *Linear Programming*, Addison Wesley, Reading, Mass.
16*. Khachian L. G. (1979), A Polynomial Algorithm in Linear Programming, *Soviet Math. Dokl.*, 20, 1, p. 191-194.
17. Klee V., Minty G. J. (1972), How good is the Simplex Algorithm, in: *Inequalities III* (O. Shish ed.), Academic Press, New York, p. 159-175.
18. Kotiah T. C. T., Steinberg D. I. (1978), On the possibility of cycling with the simplex method. *Operations Research*, 26, n° 2, p. 374-376.
19. Liebling T. M. (1972), On the number of iterations of the simplex method, *Operations Research-Verfahren*, 17, p. 248-264.
20. Mangasarian O. L. (1976), Linear Complementarity Problems Solvable by a Single Linear Program, *Mathematical Programming*, 10, p. 265-270.
21. Minoux M. (1975), *Programmation linéaire*, Cours de l'École Nationale Supérieure des Télécommunications, Paris.
22. Murty K. (1976), *Linear and Combinatorial Programming*, Wiley, New York.
23. Oettli W. (1972), An iterative method, having linear rate of convergence, for solving a pair of dual linear programs, *Mathematical Programming*, 3, p. 302-311.
24. Orchard-Hays W. (1968), *Advanced Linear Programming Computing Techniques*, McGraw-Hill, New York.
25. Orden A. (1980), A Step toward probabilistic Analysis of Simplex Method Convergence, *Mathematical programming*, 19, p. 3-13.

26*. Shor N. Z. (1970), Convergence rate of the gradient descent method with dilatation of the space, *Cybernetics*, 6, 2, p. 102-108.
27*. Shor N. Z. (1977), Cut-off methods with space extension in convex programming problems, *Cybernetics*, 13, 1, p. 94-96.
28. Simounard M. (1962). *Programmation linéaire*, Dunod, Paris.
29. Thompson G. E. (1971), *Linear Programming*, MacMillan, New York.
30. Tolla P. (1974), Méthodes de triangularisation et programmes d'ordinateur assurant l'amélioration de la précision des résultats en programmation linéaire, *Thèse 3e cycle, Université Paris VI.*
31. Tucker A. W. (1963), Combinatorial Treory Underlying Linear Programs, in: *Recent Advances in Mathematical Programming* (R. L. Graves & P. Wolfe, eds.), McGraw-Hill, New York.
32. Vajda S. (1974), *Theory of Linear and Nonlinear Programming*, Longman, London.
33. Van de Panne C. (1974), *Methods for Linear and Quadratic Programming*, North Holland, Amsterdam.
34. Zangwill W. I. (1967), The Convex Simplex Method, *Management Science*, 14, p. 221-283.
35. Zionts S. (1974), *Linear and integer programming*, Prentice Hall.

ГЛАВА 3
ОДНОМЕРНАЯ ОПТИМИЗАЦИЯ

Для нахождения оптимума функции f от n переменных $x_1, \ldots, x_n$ в гл. 4, 5 будут изложены несколько итеративных методов, требующих на каждой итерации решения оптимизационной задачи с единственным переменным,— задачи следующего типа:

найти $\alpha \geqslant 0$, минимизирующее функцию $g(\alpha) = f(x^0 + \alpha d)$, где

$x^0 = (x_1^0, \ldots, x_n^0)^T$ — последняя полученная точка,

$d = (d_1, \ldots, d_n)^T$ — направление перемещения.

Таким образом, речь идет о нахождении оптимума функции f, исходя из x^0 в направлении d. В общем случае направление d есть *направление спуска*, т. е.

$$\nabla f^T(x^0)\, d = \frac{dg}{d\alpha}(\alpha = 0) < 0.$$

Эту задачу приходится решать в огромном количестве случаев, и важно располагать алгоритмами, пригодными для ее решения.

§ 1. Методы, использующие производные

1.1. Метод Ньютона — Рафсона. Функция $g(\alpha)$ предполагается дважды непрерывно дифференцируемой. Отыскание минимума функции $g(\alpha)$ производится при помощи отыскания стационарной точки, т. е. точки α^*, удовлетворяющей (нелинейному) уравнению

$$dg/d\alpha = g'(\alpha) = 0,$$

при помощи метода Ньютона.

Если α^k — точка, полученная на k-м этапе, то функция $g'(\alpha)$ аппроксимируется своим уравнением касательной:

$$z = g'(\alpha^k) + (\alpha - \alpha^k) g''(\alpha^k),$$

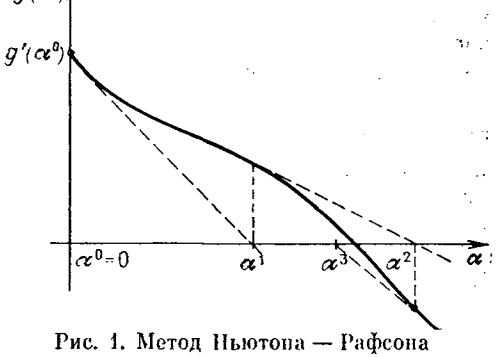


Рис. 1. Метод Ньютона — Рафсона

а точка α^{k+1} выбирается как пересечение этой прямой с осью $O\alpha$, т. е. (рис. 1)

$$\alpha^{k+1} = \alpha^k - \frac{g'(\alpha^k)}{g''(\alpha^k)}.$$

Интересно заметить, что этот процесс обладает свойством конечной сходимости, когда он применяется к квадратичной функции вида

$$g(\alpha) = u\alpha^2 + v\alpha + w \qquad (u > 0).$$

В самом деле, для любой начальной точки α^0 имеем

$$g'(\alpha^0) = 2u\alpha^0 + v,$$
$$g''(\alpha^0) = 2u,$$
$$\alpha^1 = \alpha^0 - \frac{2u\alpha^0 + v}{2u} = -\frac{v}{2u},$$

т. е. получаем минимум функции g.

Это интересная характеристика, поскольку произвольная, но достаточно регулярная функция (дважды непрерывно дифференцируемая) ведет себя как квадратичная функция в окрестности оптимума.

Неудобство этого метода состоит в необходимости вычисления в каждой точке первой и второй производных. Значит, он применим лишь тогда, когда функция g имеет достаточно простую аналитическую форму, чтобы производные могли быть вычислены в явном виде вручную. Действительно, всякий раз, когда решается новая задача, необходимо выбрать две специфические подпрограммы вычисления производных g' и g'', что не позволяет построить общие алгоритмы (т. е. применимые априори к функции любого типа).

Когда начальная точка итераций достаточно близка к искомому минимуму $\bar{\alpha}$, скорость сходимости метода Ньютона в общем случае квадратическая (см. гл. 4, п. 2.12, общее исследование скорости сходимости метода Ньютона). Однако, как уже указывалось в гл. 1, п. 4.3, глобальная сходимость метода Ньютона, вообще говоря, не гарантируется, что приводит к появлению некоторых тонкостей в его использовании.

Пример. Для иллюстрации этого пункта рассмотрим функцию $g(\alpha) = -e^{-\alpha^2}$, имеющую единственный минимум в $\alpha = 0$ и имеющую производную $g'(\alpha) = 2\alpha e^{-\alpha^2}$, представленную на рис. 2. Если взять α^0 сильно удаленным от нуля (например, $\alpha^0 = 1$), то метод порождает последовательность точек α^k, стремящихся к ∞.

Хороший способ гарантировать глобальную сходимость метода Ньютона состоит в комбинировании его с другим процессом (например, дихотомии; см. п. 1.3 и п. 2.3 выше) для быстрого получения хорошей аппроксимации искомого оптимума. Тогда несколько итераций метода Ньютона, с этой точкой в качестве исходной, достаточны для получения превосходной точности.

1.2. Метод секущих. Важным препятствием в практическом применении метода Ньютона является одновременная оценка первой и второй производных в каждой точке.

Приближая вторую производную в точке α^k выражением

$$\frac{g'(\alpha^k) - g'(\alpha^{k-1})}{\alpha^k - \alpha^{k-1}},$$

из формулы Ньютона получаем:

$$\alpha^{k+1} = \alpha^k - g'(\alpha^k)\frac{\alpha^k - \alpha^{k-1}}{g'(\alpha^k) - g'(\alpha^{k-1})}.$$

Эта формула известна под названием *метода секущих*. В самом деле, будучи примененным к отысканию корня уравнения

$$\frac{dg}{d\alpha} = g'(\alpha) = 0,$$

этот метод состоит в аппроксимации функции $g'(\alpha)$ не при помощ

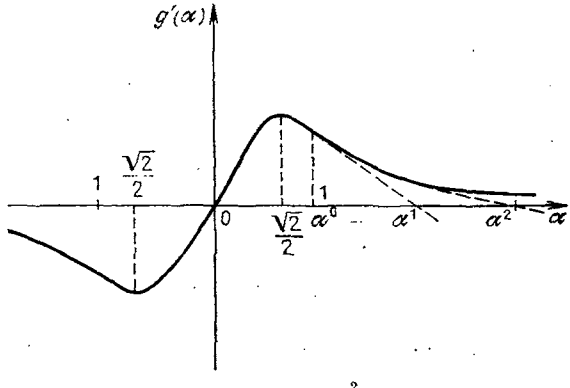


Рис. 2. Производная функции $y(\alpha) = -e^{-\alpha^2}$ и отсутствие сходимости метода Ньютона для начальной точки α^0, слишком удаленной от $\alpha^* = 0$ (здесь $\alpha^0 = 1$)

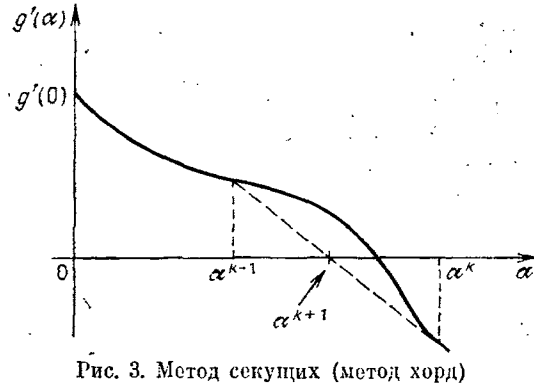


Рис. 3. Метод секущих (метод хорд)

щи своей касательной в точке α^k (как в методе Ньютона), а при помощи прямой, проходящей через точки $[\alpha^{k-1}, g'(\alpha^{k-1})]$ и $[\alpha^k, g'(\alpha^k)]$ (рис. 3). Этот метод называется также *методом хорд*.

Глобальная сходимость метода секущих, как и метода Ньютона, не гарантирована. Стало быть, начальные точки α^0 и α^1 должны быть выбраны достаточно близкими к оптимуму $\bar{\alpha}$.

Анализ асимптотической сходимости показывает, наконец, что если g трижды непрерывно дифференцируема в окрестности точки $\bar{\alpha}$ и если $g''(\bar{\alpha}) > 0$, то метод имеет сходимость порядка $\gamma = 1{,}618$ (золотое сечение). (См., например, [7].)

Заметим, наконец, что метод может рассматриваться как метод квадратичной интерполяции, в котором функция $g(\alpha)$ аппроксимируется квадратично, переходя через точку $[\alpha^k, g(\alpha^k)]$ и имея те же производные, что и g, в точках α^k и α^{k-1}.

1.3. Метод дихотомии с производными. Этот метод может рассматриваться как одномерный эквивалент алгоритмов субградиента (или градиента с фиксированным шагом), которые будут изучаться в гл. 4, § 3. Следовательно, он может применяться, в частности:

а) к выпуклым функциям, не являющимся всюду дифференцируемыми, но для которых мы умеем вычислять субградиент в каждой точке;

б) к непрерывно дифференцируемым функциям, для которых мы умеем определять значение производной в каждой точке.

Для определенности предположим, что $g(\alpha)$ непрерывно дифференцируема, что $g'(\alpha) < 0$ (направление перемещения есть направление спуска) и что существует такое $\bar{\alpha}$, что для $\alpha \geqslant \bar{\alpha}$ имеем $g'(\alpha) > 0$.

Стало быть, существует по крайней мере одна точка α^*, в которой производная функции g обращается в нуль, и речь идет о нахождении такой точки. Вообще, точка α^* будет локальным минимумом функции g на $[0, +\infty)$.

Если, кроме того, наложить условие, чтобы функция g была унимодальной (см. п. 2.2), то α^* будет (единственным) глобальным минимумом функции g на $[0, +\infty)$.

Метод состоит в определении первого интервала $[\alpha_{\min}, \alpha_{\max}]$, для которого $g'(\alpha_{\min}) < 0$, $g'(\alpha_{\max}) > 0$, а затем в прогрессивном сведении этого интервала посредством дихотомии к получению конечного интервала с достаточно малой амплитудой $\leqslant \varepsilon$.

Более точно, с одной заданной итерацией вычисляется $g'(\alpha_1)$ в точке

$$\alpha_1 = \frac{\alpha_{\min} + \alpha_{\max}}{2}.$$

Если $g'(\alpha_1) > 0$, то $\alpha_{\max}$ заменяется значением α, и производится итерация.

Если $g'(\alpha_1) < 0$, то $\alpha_{\min}$ заменяется значением α_1, и производится итерация.

Поскольку длина интервала на каждой итерации делится на 2, то отсюда выводим, что метод дихотомии линейно сходится с множителем сходимости 0,5.

Для определения начального интервала $[\alpha_{\min}, \alpha_{\max}]$ можно использовать следующую стратегию:

1) $h = h_0$ — фиксированный шаг перемещения,

$$\alpha_{\min} = 0;$$

2) вычислить $g'(h)$;
если $g'(h) < 0$, сделать $\alpha_{\min} \leftarrow h$, $h \leftarrow 2h$ и вернуться к 2);
если $g'(h) > 0$, сделать $\alpha_{\max} = h$. Конец.

§ 2. Методы, не использующие производных

2.1. Квадратичная интерполяция. По отношению к предыдущим этот метод обладает тем преимуществом, что он не требует вычисления первой или второй производных функции g. Однако ниже будет показано, что его сходимость может быть гарантирована лишь для достаточно регулярных функций (непрерывных и много раз дифференцируемых).

Идея заключается в следующем. Если $\alpha_1 \leqslant \alpha_2 \leqslant \alpha_3$ — три значения α, для которых $g(\alpha_1) \geqslant g(\alpha_2)$, $g(\alpha_3) \geqslant g(\alpha_2)$, то приближаем функцию $g(\alpha)$ на интервале $[\alpha_1, \alpha_3]$ квадратичной функцией, имеющей те же значения, что g, в точках α_1, α_2, α_3. Уравнение имеет вид

$$q(\alpha) = \sum_{i=1}^{3} g(\alpha_i) \frac{\prod_{j \neq i} (\alpha - \alpha_j)}{\prod_{j \neq i} (\alpha_i - \alpha_j)}.$$

Минимум функции $q(\alpha)$ достигается в точке

$$\alpha_4 = \frac{1}{2} \frac{r_{23} g(\alpha_1) + r_{31} g(\alpha_2) + r_{12} g(\alpha_3)}{s_{23} g(\alpha_1) + s_{31} g(\alpha_2) + s_{12} g(\alpha_3)}$$

(где $r_{ij} = \alpha_i^2 - \alpha_j^2$, $s_{ij} = \alpha_i - \alpha_j$).

Заметим, что $\alpha_4 \in [\alpha_1, \alpha_3]$. Тогда точка α_4 берется в качестве приближения точки оптимума функции $g(x)$ на $[\alpha_1, \alpha_3]$.

Далее построение повторяется с тремя новыми точками:

$$\begin{aligned}
(\alpha_1', \alpha_2', \alpha_3') &= (\alpha_2, \alpha_4, \alpha_3), \text{ если } \alpha_2 \leqslant \alpha_4 \leqslant \alpha_3, \quad g(\alpha_4) \leqslant g(\alpha_2), \\
&= (\alpha_1, \alpha_2, \alpha_4), \text{ если } \alpha_2 \leqslant \alpha_4 \leqslant \alpha_3, \quad g(\alpha_4) > g(\alpha_2), \\
&= (\alpha_1, \alpha_4, \alpha_2), \text{ если } \alpha_1 \leqslant \alpha_4 \leqslant \alpha_2, \quad g(\alpha_4) \leqslant g(\alpha_2), \\
&= (\alpha_4, \alpha_2, \alpha_3), \text{ если } \alpha_1 \leqslant \alpha_4 \leqslant \alpha_2, \quad g(\alpha_4) > g(\alpha_2).
\end{aligned}$$

Для определения начала процесса выберем три значения $a < b < c$, удовлетворяющие равенствам $b - a = c - b = \Delta$ и такие, что

$$g(b) \leqslant g(a), \quad g(b) \leqslant g(c).$$

Воспользуемся, например, следующей процедурой.

Зададим начальную точку α^0, произвольный шаг перемещения δ, затем вычислим $g(\alpha^0)$ и $g(\alpha^1) = g(\alpha^0 + \delta)$. Могут представиться два случая.

Случай 1: $g(\alpha^0+\delta)\leqslant g(\alpha^0)$.

Вычисляем тогда последовательно $g(\alpha^2)$, ..., $g(\alpha^p)$ для $\alpha^2=\alpha^1+2\delta$, $\alpha^3=\alpha^2+4\delta$, ..., $\alpha^p=\alpha^{p-1}+2^{p-1}\delta$, пока значения функции убывают.

Останавливаемся на шаге p, как только функция снова начинает возрастать, и, значит, имеем

$$g(\alpha^0)\geqslant g(\alpha^1)>g(\alpha^2)>\ldots>g(\alpha^{p-1})<g(\alpha^p).$$

После этого вычисляем $g(\alpha^{p+1})$ для $p+1$-й точки $\alpha^{p+1}=\alpha^p-2^{p-2}\delta$.

При таком построении четыре точки α^{p-2}, α^{p-1}, α^p, α^{p+1} будут равноотстоящими с $\Delta=2^{p-1}\delta$.

Находим искомые три точки a, b, c, исключая
либо α^p (если $g(\alpha^{p-1})<g(\alpha^{p+1})$),
либо α^{p-2} (если $g(\alpha^{p+1})<g(\alpha^{p-1})$).

Случай 2: $g(\alpha^0+\delta)>g(\alpha^0)$.

Теперь заменим α^0 на $\alpha^0+\delta$ (новая начальная точка), а δ на $-\delta$ (новый шаг перемещения). Вернемся к случаю 1 (заметив, однако, что новое $g(\alpha^1)$ равно старому $g(\alpha^0)$ и что бесполезно пересчитывать это значение).

(Отметим, что эта процедура сходится всегда очень быстро, даже при неудачном выборе слишком малого значения δ. Например, при $p=10$ шаг перемещения умножается на $2^{10}\simeq 1000$.)

Если предположить, что функция $g(\alpha)$ унимодальна, т. е. (см. п. 2.2) допускает единственный минимум на начальном интервале, то глобальная сходимость метода к минимуму функции $g(\alpha)$ может быть доказана при помощи теоремы о глобальной сходимости из гл. 1, п. 4.5.

Действительно, одна итерация алгоритма переводит три точки $(\alpha_1, \alpha_2, \alpha_3)$ в новые три точки $(\alpha_1', \alpha_2', \alpha_3')$.

Стало быть, алгоритм может быть представлен как отображение A: $R^3\to R^3$, определяемое равенством $(\alpha_1', \alpha_2', \alpha_3')=A(\alpha_1, \alpha_2, \alpha_3)$.

Легко убедиться в том, что это отображение непрерывно (а значит, замкнуто). Множество решений есть точка $(\bar\alpha, \bar\alpha, \bar\alpha)$, где $\bar\alpha$ — точка минимума функции $g(\alpha)$. Если функция $g(\alpha)$ непрерывна и унимодальна, то функция

$$z(\alpha_1, \alpha_2, \alpha_3)=g(\alpha_1)+g(\alpha_2)+g(\alpha_3)$$

есть функция спуска. Наконец, все последовательно полученные точки содержатся в начальном интервале (а значит, в замкнутом ограниченном множестве). А поскольку все условия теоремы выполнены, отсюда следует глобальная сходимость алгоритма.

Что касается асимптотической сходимости, то можно показать посредством некоторых условий регулярности (трижды непрерывно дифференцируемая функция), что скорость сходимости метода суперлинейна и имеет порядок $\gamma=1{,}3$ (см., например, [7]).

Очевидно, можно представить себе существование других методов, основанных на принципе полиномиальной интерполяции,

которые могут использовать полиномы более высокого порядка и возможную информацию относительно первых производных или производных более высокого порядка.

2.2. Унимодальные функции. Теперь мы перейдем к описанию других методов, более общих, чем методы, изучавшиеся до сих пор, в том смысле, что они не требуют условия непрерывности или дифференцируемости. Они просто предполагают, что по крайней мере в некотором интервале $[A, B]$ минимизируемая функция g обладает свойством унимодальности.

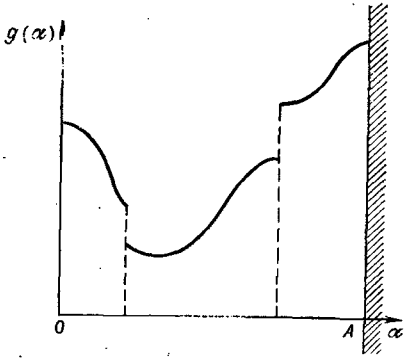


Рис. 4. Унимодальная функция на $[0, A]$

Определение 1. Говорят, что функция g *унимодальна* на действительном интервале $[A, B]$, если она имеет минимум $\bar{\alpha} \in [A, B]$ и если $\forall \alpha_1, \alpha_2 \in [A, B]$ $(\alpha_1 < \alpha_2)$ выполняются соотношения

$$\alpha_2 \leqslant \bar{\alpha} \Rightarrow g(\alpha_1) > g(\alpha_2),$$

$$\alpha_1 \geqslant \bar{\alpha} \Rightarrow g(\alpha_1) < g(\alpha_2).$$

Следовательно, унимодальная функция на $[A, B]$ обладает тем свойством, что она имеет единственный локальный минимум, но она не обязана быть дифференцируемой — она даже не обязана быть непрерывной (рис. 4).

Из определения унимодальных функций следует, что если мы вычисляем значение унимодальной функции g в четырех точках a, b, c, d интервала $[a, d]$, то всегда существует подынтервал, который не содержит оптимума и может быть, таким образом, удален (рис. 5). Мы получаем уменьшенный интервал, на котором эту операцию можно повторить.

Для получения интервала $[a, d]$, содержащего оптимум, можно применить тот же метод, который был развит выше в п. 2.1 в рамках квадратичной интерполяции.

2.3. Метод дихотомии без производных. Принцип этого метода очень близок изложенному в п. 1.3, но здесь мы предполагаем, что либо информации о производных нет, либо функция не дифференцируема.

Этот метод позволяет на каждом шаге уменьшать вдвое длину интервала, содержащего оптимум, причем значения функции g вычисляются с этой целью в двух определенных точках. Если над функцией g произведено n вычислений, то длину интервала можно уменьшить в $2^{(n-3)/2}$ раза.

Сначала разбиваем интервал $[a^0, b^0]$. Берем середину $c^0 = (a^0 + b^0)/2$, затем две точки $d^0 = (a^0 + c^0)/2$, $e^0 = (c^0 + b^0)/2$; получим пять точек, удаленных на равные расстояния $\delta^0 = (b^0 - a^0)/4$.

С учетом унимодальности легко видеть, что всегда можно исключить два из четырех подынтервалов (поскольку точка оптимума не может в них содержаться) и что остаются только два смежных подынтервала $[a^1, c^1]$ и $[c^1, b^1]$ (рис. 6).

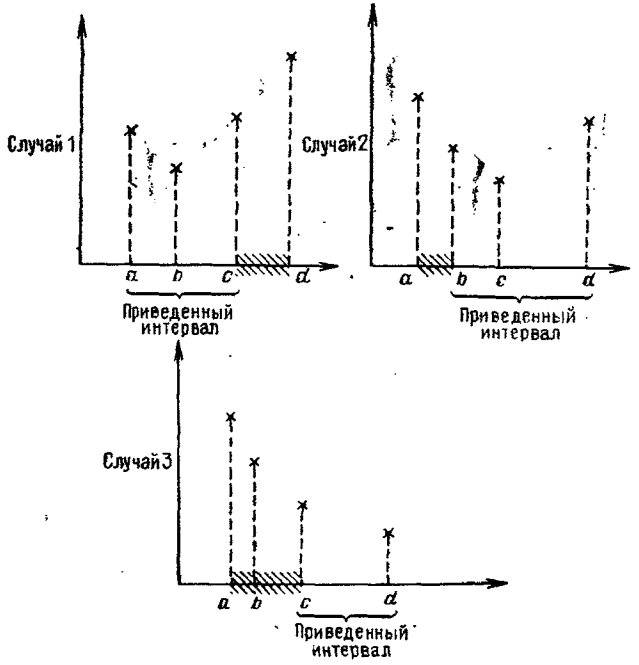


Рис. 5. Уменьшение (приведение) интервала $[a, d]$ с использованием свойства унимодальности

Таким образом, все сводится к той же задаче на отрезке $[a^1, b^1]$ половинной длины.

Например, на рис. 6 можно исключить отрезки $[a^0, d^0]$ и $[d^0, c^0]$. Тогда $a^1 = b^0$, $c^1 = e^0$.

На следующем этапе понадобится только два дополнительных вычисления функции g в точках $d^1 = (a^1 + c^1)/2$ и $e^1 = (c^1 + b^1)/2$ и т. д.

Таблица 1 содержит числовые значения для начального интервала в зависимости от n при вычислении функции g.

Таблица 1

n	$\frac{b^n - a^n}{b^0 - a^0} = \frac{1}{2^{(n-3)/2}}$	n	$\frac{b^n - a^n}{b^0 - a^0} = \frac{1}{2^{(n-3)/2}}$
17	10^{-2}	29	10^{-4}
23	10^{-3}	42	10^{-6}

Глобальная сходимость метода для унимодальных функций следует из определения. С другой стороны, легко видеть, что скорость сходимости имеет линейный характер с коэффициентом $1/\sqrt{2} \simeq 0{,}7$.

2.4. Использование последовательности Фибоначчи. Метод дихотомии не оптимален в том смысле, что для конечного фиксированного числа N вычислений значений функции g он не приводит к наименьшему возможному уменьшенному интервалу. Как будет показано ниже, оптимальный метод может быть получен при помощи последовательности Фибоначчи [5].

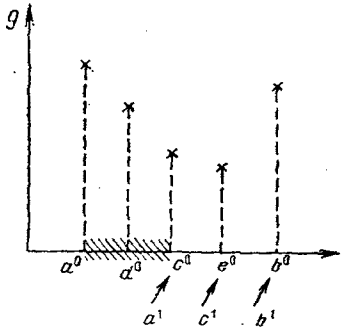


Рис. 6. Иллюстрация действия метода дихотомии

Идея состоит в вычислении значения функции g в N точках, выбранных так, чтобы результат, полученный для каждой новой точки, позволял исключать (по возможности наибольший) подынтервал начального интервала.

Предположим, что задан начальный интервал $[a^1, d^1]$ и что мы располагаем значениями функции g в точках a^1, d^1 и в двух промежуточных точках b^1 и c^1. Тогда можно, используя унимодальность, исключить один подынтервал, $[a^1, b^1]$ или $[c^1, d^1]$ (см. п. 2.2).

Пусть $\Delta^1 = d^1 - a^1$ — длина начального интервала. Если исключить $[c^1, d^1]$, то останется интервал длины $c^1 - a^1$. Если исключить $[a^1, b^1]$, то останется интервал длины $d^1 - b^1$. Если мы хотим, чтобы длина d^2 полученного интервала не зависела от результата теста (а значит, и от функции), то должны выполняться равенства $c^1 - a^1 = d^1 - b^1 = \Delta^2$; иными словами, точки b^1 и c^1 должны быть симметричны относительно середины интервала $[a^1, d^1]$.

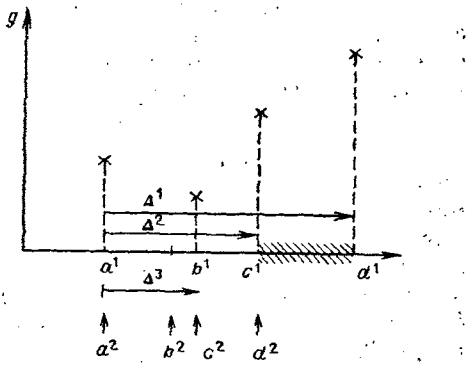


Рис. 7. В методе, называемом методом Фибоначчи, должно выполняться соотношение $\Delta^1/\Delta^2 = F_{N-1}/F_{N-2}$, где F_{N-1} и F_{N-2} — два соседних члена последовательности Фибоначчи

Предположим теперь, что интервал $[c^1, d^1]$ исключен (как на рис. 7). Чтобы возобновить операцию с приведенным интервалом

$[a^1, c^1]$, необходимо вычислить значение функции в дополнительной точке b^2, и эта точка, как мы видели, может быть лишь симметричной точке b^1 относительно середины интервала $[a^1, c^1]$.

Тогда из соотношений

$$d^1 - a^1 = (c^1 - a^1) + (d^1 - c^1) = (c^1 - a^1) + (b^1 - a^1)$$

получаем

$$\Delta^1 = \Delta^2 + \Delta^3.$$

Легко обобщить предыдущие рассуждения, чтобы показать, что

$$\Delta^k = \Delta^{k+1} + \Delta^{k+2} \quad \forall k.$$

Пусть Δ^{N-1} — длина интервала, полученного в конце N вычислений функции (N фиксировано); введем числа F_0, F_1, ..., F_{N-1}:

$$\Delta^k = F_{N-k}\Delta^{N-1}.$$

(Заметим, что при $k = N - 1$ имеем $\Delta^{N-1} = F_1\Delta^{N-1} \Rightarrow F_1 = 1$.)

Запишем

$$\frac{\Delta^k}{\Delta^{N-1}} = \frac{\Delta^{k+1}}{\Delta^{N-1}} + \frac{\Delta^{k+2}}{\Delta^{N-1}}.$$

Легко видеть, что числа F_n удовлетворяют рекуррентному соотношению

$$F_n = F_{n-1} + F_{n-2}, \qquad n = 3, 4, \ldots, N - 1,$$
$$F_1 = 1.$$

Заметим, однако, что существует некоторая неопределенность в выборе F_2, и, как только F_2 выбрано, вся последовательность F_i определена, равно как и длина Δ^k всех последовательных интервалов. Тогда соотношение

$$\Delta^{N-1} = \Delta^1/F_{N-1}$$

показывает, что для того, чтобы конечный интервал имел минимальную длину Δ^{N-1}, следует выбрать F_2 так, чтобы F_{N-1} было максимальным, а значит, требуется, чтобы само F_2 было максимальным. А поскольку должно быть $F_1 \geqslant F_2/2$, то отсюда следует, что максимум для F_2 есть $F_2 = 2$. Тогда последовательность чисел F_n полностью определяется двумя первыми членами: $F_n = 1$ и $F_2 = 2$.

Эта последовательность называется *последовательностью Фибоначчи* (псевдоним Леонардо Пизанского, первым изучавшего эти последовательности в 1202 г.). В табл. 2 приведены первые 20 чисел этой последовательности.

Следует заметить, что в этом методе расположение первых точек, т. е. соотношение

$$\frac{\Delta^1}{\Delta^2} = \frac{F_{N-1}}{F_{N-2}}$$

зависит от числа N вычислений функции g, которые мы собираемся осуществить. Это не очень стеснительное ограничение, в той мере, в которой вообще фиксируется заранее точность вычисления. Например, если начальный интервал имеет длину $\Delta^2 = 1$ и

Таблица 2

n	F_n	n	F_n	n	F_n	n	F_n
1	1	6	13	11	144	16	1597
2	2	7	21	12	233	17	2584
3	3	8	34	13	377	18	4181
4	5	9	55	14	610	19	6765
5	8	10	89	15	987	20	10946

принятая точность есть 10^{-3}, то таблица позволяет определить N:

$$\frac{\Lambda^{N-1}}{\Delta^1} = \frac{1}{F_{N-1}} = 10^{-3},$$

что дает $N = 16$.

Наконец, укажем, что метод Фибоначчи может быть обобщен на случай функции n переменных (см. [9]).

2.5. Метод золотого сечения. Если мы не всегда знаем заранее число N вычислений, которые мы хотим осуществить, то можно воспользоваться методом золотого сечения.

Следуем тому же принципу, что и выше (исключения подынтервалов для каждой новой вычисляемой точки); *метод золотого сечения* состоит в том, что длины последовательных интервалов берутся в фиксированном отношении:

$$\frac{\Lambda^1}{\Delta^2} = \frac{\Lambda^2}{\Delta^3} = \ldots = \gamma,$$

так, чтобы на $k+1$-м этапе относительное расположение точек было (с точностью до преобразования подобия) то же, что и на k-м этапе.

Поскольку

$$\Delta^k = \Delta^{k+1} + \Delta^{k+2},$$

то при условии требования

$$\frac{\Lambda^k}{\Delta^{k+1}} = \frac{\Delta^{k+1}}{\Delta^{k+2}} = \gamma$$

получаем

$$\frac{\Lambda^k}{\Delta^{k+1}} = 1 + \frac{\Delta^{k+2}}{\Delta^{k+1}}.$$

Пусть $\gamma = 1 + (1/\gamma)$ (или, еще, $\gamma^2 - \gamma - 1 = 0$) — уравнение, корнем которого является золотое сечение:

$$\gamma = \frac{\sqrt{5}+1}{2} \simeq 1{,}618.$$

Из изложенного выше следует, что скорость сходимости нашего метода имеет линейный порядок с коэффициентом $1/\gamma = 0{,}618$.

Разумеется, метод золотого сечения не оптимален, но мы без труда можем убедиться в том, что для достаточно высокого числа N вычислений функции g он асимптотически приводит к тому же расположению точек, что и в методе Фибоначчи.

В самом деле,

$$\lim_{N\to\infty} \frac{F_N}{F_{N-1}} = \gamma$$

(например, $F_{11}/F_{10} = 144/89 = 1{,}617$).

В табл. 3 приведено сравнение коэффициентов уменьшения интервалов, полученных различными методами (дихотомии, Фибоначчи, золотого сечения) для различных значений числа N вычислений.

Таблица 3

Число вычислений значения функции: сравнение различных методов

Отношение уменьшения Δ^n/Δ^1	Дихотомия без производных $2^{-(n-3)/2}$	Золотое сечение $(\sqrt{5}-1/2)^{n-3}$	Фибоначчи F_n^{-1}
10^{-2}	17	13	11
10^{-3}	23	18	15
10^{-4}	29	22	20
10^{-5}	35	27	25
10^{-6}	42	31	30

2.6. Одномерный «экономичный» («наилучший») поиск. Описанные выше методы преследовали цель получить более или менее точное приближение оптимума (или локального оптимума) функции $g(\alpha) = f(x + \alpha d)$ для $\alpha \geqslant 0$.

На самом деле сходимость некоторых нелинейных оптимизационных алгоритмов (наискорейшего спуска, алгоритма БФГШ; см. гл. 4, пп. 2.2, 2.11) может быть получена без того, чтобы требовалось находить близкую к оптимальной точку $x + \alpha d$ (или близкую к локальному оптимуму) для функции 1 в направлении d.

Предлагались разные правила выбора, основанные на гораздо менее ограничительных условиях. Они позволяют получить точку с требуемыми характеристиками с весьма ограниченным числом оценок функции.

Здесь будут представлены два типа процессов: правило Голдстейна [2], которое применяется, когда нельзя оценить градиент функции (или когда получение оценки обходится слишком дорого), и процедуру, предложенную Вольфе [10] и Пауэллом [8], которая требует оценки градиента всякий раз, когда функция вычислена.

Общий принцип, лежащий в основе всех этих методов, заключается в следующем:

a) α не должно выбираться слишком большим (в противном случае алгоритм может начать осциллировать);

b) α не должно выбираться слишком малым (в противном случае алгоритм может сходиться слишком медленно).

В правиле Голдстейна (G) условие a) выполняется в силу соотношения

$$g(\alpha) \leqslant g(0) + m_1 \alpha g'(0), \tag{1}$$

где m_1 — коэффициент, выбранный в (0, 1); условие b) обеспечивается соотношением

$$g(\alpha) \geqslant g(0) + m_2 \alpha g'(0), \tag{2}$$

где m_2 — коэффициент, выбранный в интервале $(m_1, 1)$.

Рис. 8 дает пример множества точек, удовлетворяющих соотношениям (1), (2).

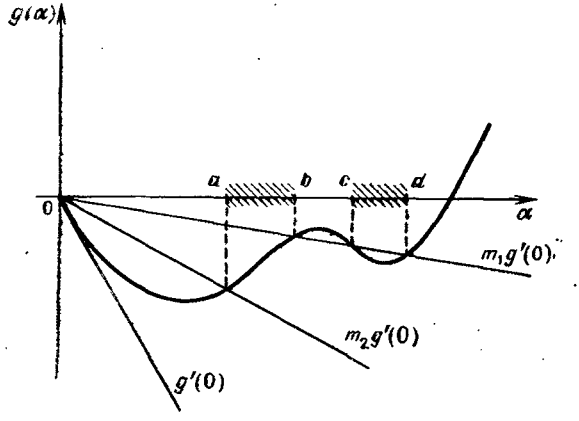


Рис. 8. Множество точек, удовлетворяющих правилу Голдстейна (отрезки $[a, b]$ и $[c, d]$)

Отметим, что условие (1) обеспечивает тот факт, что новая полученная точка $x' = x + \alpha d$ удовлетворяет неравенству $f(x') < f(x)$ (условие спуска).

Замечание. Вариант правила Голдстейна, предложенный Армийо [1], состоит в выборе такого α, что:

1) α удовлетворяет соотношению (1);

2) $M\alpha$ не удовлетворяет соотношению (1).

При этом $M > 1$ (обычно выбирается в пределе от 5 до 10).

В правиле Вольфе — Пауэлла (WP) снова именно соотношение (1) приводит к тому, что α выбирается не слишком большим. Но поскольку предполагается, что вычисление градиента функции f требует не намного больших вычислений, чем оценка самой функции, то соотношение (2) может быть заменено условием

$$g'(\alpha) \geqslant m_3 g'(0), \tag{3}$$

где m_3 — постоянный коэффициент, выбранный в $(m_1, 1)$. В самом деле, $g'(\alpha)$ в каждой точке α может вычисляться по формуле

$$g'(\alpha) = d^T \nabla f(x + \alpha d).$$

На рис. 9 показано множество точек, удовлетворяющих соотношениям (1), (3) правила Вольфе — Пауэлла.

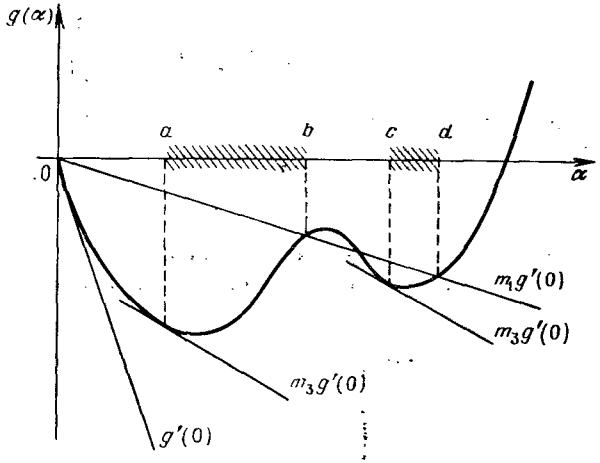


Рис. 9. Множество точек, удовлетворяющих правилу Вольфе и Пауэлла (отрезки $[a, b]$ и $[c, d]$)

Выбор коэффициентов m_1 и m_3 не является слишком критическим. Обычно выбираются значения $m_1 = 0{,}1$, $m_3 = 0{,}7$ (см., например, [6]).

Замечание. Чтобы подчеркнуть необходимость и полезность каждого из двух условий (1) и (3), интересно показать, как можно установить, например, сходимость алгоритма наискорейшего спуска (см. также гл. 4, п. 2.2), используя правило (WP).

Для этого предположим, что функция f ограничена снизу и что ∇f удовлетворяет условию Липшица типа:

$$\forall x, \quad \forall y, \quad \|\nabla f(x) - \nabla f(y)\| \leqslant K\|x - y\|,$$

где K — постоянная.

Обозначим через $\{x^k\}$ последовательность точек, полученных при помощи правила (WP), если отыскивать последовательно x^1 из x^0 в направлении $d_0 = -\nabla f(x^0)$, ..., x^k из x^{k-1} в направлении $d_{k-1} = -\nabla f(x^{k-1})$. Соотношение (1) позволяет записать

$$f(x^{k+1}) \leqslant f(x^k) + m_1 \alpha \nabla f^T(x^k) d_k.$$

В силу равенства

$$x^{k+1} - x^k = \alpha d_k = -\alpha \nabla f(x^k)$$

получаем

$$f(x^{k+1}) \leqslant f(x^k) - m_1 \|x^{k+1} - x^k\| \cdot \|\nabla f(x^k)\|.$$

Если функция f ограничена снизу, то должно быть

$$\sum_{k=0}^{\infty} (f(x^k) - f(x^{k+1})) < +\infty,$$

значит, ряд

$$\sum_{k=0}^{\infty} \| x^{k+1} - x^k \| \cdot \| \nabla f(x^k) \|$$

должен быть сходящимся.

Теперь, используя соотношение (3), получаем

$$\nabla f^T(x^{k+1}) d_k \geqslant m_3 \nabla f^T(x^k) d_k,$$

откуда

$$(\nabla f(x^{k+1}) - \nabla f(x^k))^T d_k \geqslant (m_3 - 1) \nabla f^T(x^k) d_k;$$

отсюда, принимая во внимание тот факт, что правая часть положительна и что $-\nabla f(x^k) = d_k$, выводим неравенство

$$\| \nabla f(x^{k+1}) - \nabla f(x^k) \| \cdot \| d_k \| \geqslant (1 - m_3) \| \nabla f(x^k) \| \cdot \| d_k \|.$$

Условие Липшица для ∇f позволяет записать

$$K \| x^{k+1} - x^k \| \geqslant (1 - m_3) \| \nabla f(x^k) \|.$$

Отсюда следует, что ряд

$$\sum_{k=0}^{\infty} \| \nabla f(x^k) \|^2$$

должен быть сходящимся, и, значит, $\| \nabla f(x^m) \| \to 0$.

Итак, последовательность $\{x^k\}$, порожденная алгоритмом, имеет стационарную предельную точку x^* (т. е. удовлетворяющую $\nabla f(x^*) = 0$; в самом деле, функция ∇f непрерывна, поскольку является липшицевой).

Ясно, как два условия (1) и (3) правила WP участвуют здесь при доказательстве сходимости.

Использование правил Голдстейна (G) или Вольфе — Пауэлла (WP) особенно просто. Алгоритм имеет следующий вид:

a) Определить $\alpha_{\min} = 0$, $\alpha_{\max} = +\infty$. Найти $g'(0) = \nabla f^T(x) d$ и присвоить переменному α начальное значение (первую тестированную точку).

b) Вычислить $g(\alpha) = f(x + \alpha d)$; если $g(\alpha) \leqslant g(0) + m_1 \alpha g'(0)$, то перейти к c); иначе положить $\alpha_{\max} = \alpha$ и перейти к e).

c) Вычислить $g'(\alpha) = \nabla f^T(x + \alpha d) d$, затем сравнить $g'(\alpha)$ и $m_3 g'(0)$; если $g'(\alpha) \geqslant m_3 g'(0)$, то конец; если $g'(\alpha) < m_3 g'(0)$, то перейти к d); } (WP)
сравнить $g(\alpha)$ и $g(0) + m_2 \alpha g'(0)$; если $g(\alpha) \geqslant g(0) + m_2 \alpha g'(0)$, то конец; иначе перейти к d). } (G)

d) Положить $\alpha_{\min} = \alpha$.

e) Найти новое значение для α в интервале $(\alpha_{\min}, \alpha_{\max})$ и вернуться к b).

Заметим также, что для выбора начального значения необходимо каждый раз, как это возможно, использовать информацию, доставляемую алгоритмом оптимизации. Таким образом, в квазиньютоновских методах (гл. 4, пп. 2.9, 2.11) на каждой итерации получают оценку шага. Очень интересно использовать это значение для выбора начального значения α, ибо если оценка хорошая (таков случай с квазиньютоновскими методами, как только мы попадаем в окрестность оптимума), то для получения методом (G) или (WP) удовлетворительной точки достаточно единственной оценки функции (возможно, ее градиента). Тем самым эти методы оправдывают свое наименование «экономичные».

§ 3. Алгоритмы одномерной оптимизации и замкнутые многозначные отображения

3.1. Точная одномерная оптимизация. Итерационный тип многих алгоритмов математического программирования строится по следующей схеме:

a) выбрать в текущей точке x направление перемещения d;

b) исходя из x, найти минимум функции f в направлении d, т. е. найти такое α, что

$$f(x + \bar{\alpha} d) = \min_{\alpha \geqslant 0} \{f(x + \alpha d)\}.$$

Тогда точка $y = x + \bar{\alpha} d$ выбирается в качестве начальной точки следующей итерации.

Ясно, что эта схема ведет к рассмотрению алгоритма как композиции двух многозначных отображений D и U:

— отображение D ставит в соответствие точке x направление перемещения $d \in D(x)$;

— отображение U ставит в соответствие паре (x, d) точку $y \in U(x, d)$, где $U(x, d)$ определяется как множество точек вида $x + \alpha d$, где f достигает своего минимума:

$$U(x, d) = \{y | y = x + \alpha d,\ \alpha \geqslant 0,\ f(y) = f(x + \bar{\alpha} d)\}.$$

В гл. 1 (§ 4, теорема 10) показано, что существенным свойством для установления сходимости такого алгоритма является замкнутость произведения $U \circ D$.

Чтобы иметь возможность применить одно из предложений 2 или 3 гл. 1, необходимо убедиться в том, что отображение U, соответствующее одномерной оптимизации, замкнуто.

Итак, приводимый ниже результат очень полезен для доказательства сходимости многих алгоритмов математического программирования.

Теорема 1. *Если функция f непрерывна на $\mathbf{R}^n$, то отображение U, определенное формулой*

$$U(x, d) = \left\{ y | y = x + \alpha d\ (\alpha \geqslant 0);\ f(y) = \min_{\alpha \geqslant 0} f(x + \alpha d) \right\},$$

замкнуто в точке (x, d) при любом $d \neq 0$.

Доказательство. Рассмотрим две последовательности $\{x^k\}$, $\{d^k\}$, удовлетворяющие условиям $x^k \to x$, $d^k \to d \neq 0$. Предположим, что $y^k \in U(x^k, d^k)$ для любого k и что $y^k \to y$. Покажем, что при этих условиях $y \in U(x, d)$.

По определению отображения U при любом k существует такое $\alpha^k \geqslant 0$, что $y^k = x^k + \alpha^k d^k$. Следовательно,

$$\alpha^k = \frac{\|y^k - x^k\|}{\|d^k\|},$$

а значит,

$$\alpha^k \to \bar{\alpha} = \frac{\|y - x\|}{\|d\|}, \qquad k \to \infty;$$

и, стало быть, $y = x + \bar{\alpha} d$.

Теперь достаточно показать, что минимум функции f на луче, исходящем из x в направлении d, достигается в точке y.

Имеем

$$f(y^k) \leqslant f(x^k + \alpha d^k) \quad \forall k, \quad \forall \alpha \geqslant 0;$$

в силу непрерывности функции f, переходя к пределу, получаем

$$\forall \alpha \geqslant 0, \quad f(y) \leqslant f(x + \alpha d),$$

чем доказываем, что

$$f(y) = \min_{\alpha \geqslant 0} \{f(x + \alpha d)\}$$

и, значит, $y \in U(x, d)$.

Условие $d \neq 0$ совершенно необходимо. В самом деле, если $d = 0$, то понятие оптимизации функции f в направлении d теряет смысл, и отображение $U(x, d)$ становится неопределенным. Это условие не является ограничительным, ибо, как будет показано, для большинства алгоритмов, когда вычисление направления перемещения дает $d = 0$, это означает, что текущая точка x удовлетворяет необходимому условию оптимальности (стационарность для оптимизации без ограничений; условие Куна — Таккера для оптимизации с ограничениями).

3.2. Приближенная одномерная оптимизация. Поскольку методы, изложенные в § 1.2, имеют итеративную природу, то точный минимум функции $g(\alpha) = f(x + \alpha d)$ не может быть получен за конечное число итераций. Стало быть, практически всегда придется ограничиться приближенным минимумом. Тогда главное — показать, что условие замкнутости, установленное выше для идеализированного случая точной одномерной оптимизации, сохраняется.

Мы будем более точно исследовать два типа аппроксимации.

Тип 1. Он соответствует случаю, когда поиск прерывается, как только относительная погрешность для $\bar{\alpha}$ минимума функции $g(\alpha)$ становится меньше фиксированного процента δ, т. е.

$$\frac{|\alpha - \bar{\alpha}|}{\bar{\alpha}} \leqslant \delta.$$

Именно эта ситуация встречается, когда используются методы § 2 (дихотомия, метод Фибоначчи, метод золотого сечения).

Тип 2. Он соответствует случаю, когда поиск прерывается, как только получена аппроксимация с точностью ε для искомого оптимума, т. е. когда $|g(\alpha^k) - g(\bar{\alpha})| \leqslant \varepsilon$.

Эти два типа аппроксимации, вообще говоря, очень тесно связаны. Например, предположим, что функция g дважды непрерывно дифференцируема с положительной второй производной, ограниченной константой $K > 0$ на отрезке (α_1, α_2) длины $\alpha_2 - \alpha_1 = \delta\alpha_1$, содержащем оптимум $\bar{\alpha}$.

Предположив для определенности, что $g(\alpha_1) = \min\{g(\alpha_1); g(\alpha_2)\}$, мы можем показать, что

$$|g(\alpha_1) - g(\bar{\alpha})| \leqslant \varepsilon \quad \text{для} \quad \varepsilon = \frac{K}{2}\left[\frac{g(\alpha_2) - g(\alpha_1)}{K\delta\alpha_1} - \frac{\delta\alpha_1}{2}\right]^2.$$

Аппроксимация типа 1 приводит к изучению многозначного отображения

$$U_\delta(x, d) = \{y \mid y = x + \alpha d (\alpha \geqslant 0);\ |\alpha - \bar{\alpha}| \leqslant \delta\bar{\alpha}\}$$

с $\bar{\alpha}$, удовлетворяющим равенству $g(\bar{\alpha}) = \min\limits_{\alpha \geqslant 0} g(\alpha)$.

Теорема 2. *Если функция f непрерывна на $\mathbb{R}^n$ и если $f(x) \to +\infty$ при $\|x\| \to +\infty$, то отображение U_δ замкнуто в любой точке (x, d), для которой $d \neq 0$.*

Доказательство. Можно рассматривать U_d как композицию двух отображений: a) отображения U из § 3.1, которое точке (x, d) ставит в соответствие множество точек $y = x + \alpha d$, где α удовлетворяет равенству $g(\alpha) = g(\bar{\alpha})$ (это, вообще говоря, отрезок вида $[\alpha_{\min}, \alpha_{\max}]$); b) отображение V_δ, которое отрезку $[\alpha_{\min}, \alpha_{\max}]$ с $\alpha_{\min} > 0$ ставит соответствие «расширенный» отрезок

$$V_\delta[\alpha_{\min}, \alpha_{\max}] = [\alpha_{\min}(1 - \delta), \alpha_{\max}(1 + \delta)].$$

Отображение U замкнуто в силу теоремы 1 из п. 3.1, и легко показать, что V_δ замкнуто.

Если множество $U(x, d)$ ограничено (это, например, случай, когда $f(x) \to +\infty$ для $\|x\| \to +\infty$); то тогда можно применить предложение 2 гл. 1, § 4, и получить отсюда, что U_δ замкнуто.

Аппроксимация типа 2 ведет к изучению многозначного отображения

$$U_\varepsilon(x, d) = \{y \mid y = x + \alpha d (\alpha \geqslant 0);\ g(\alpha) \leqslant g(\bar{\alpha}) + \varepsilon\},$$

где

$$g(\bar{\alpha}) = \min_{\alpha \geqslant 0} g(\alpha).$$

Легко показать, что если f непрерывно, то многозначное отображение U замкнуто. Доказательство пошагово аналогично доказательству теоремы 1 п. 3.1.

В заключение отметим интерес и удобство понятия многозначного отображения для анализа погрешностей из-за прерывания итерационного алгоритма и, в более общем виде, для анализа недетерминистских алгоритмических процессов.

3.3. Одномерная оптимизация с ограничениями. В гл. 5 будет показано, что некоторые методы оптимизации с ограничениями (например, методы возможных направлений) приводят к одномерной оптимизации, в которой требуется, чтобы полученная точка y была включена в некоторое множество $\mathscr{D} \subset \mathbf{R}^n$, называемое областью допустимых решений задачи. Предполагается, очевидно, что начальная точка x принадлежит $\mathscr{D}$.

Тогда мы приходим к изучению отображения

$$U_{\mathscr{D}}(x, d) = \left\{ y \mid y = x + \alpha d\,(\alpha \geqslant 0);\ y \in \mathscr{D};\ f(y) = \min_{\substack{\alpha \geqslant 0 \\ x+\alpha d \in \mathscr{D}}} f(x + \alpha d) \right\}.$$

Исследуем замкнутость отображения $U_{\mathscr{D}}$.

Прежде всего, если $y^k \to y$ при $y^k \in U_{\mathscr{D}}(x^k, d^k)$ (а значит, $y^k \in \mathscr{D}$), то необходимо, чтобы $y \in \mathscr{D}$. Для этого требуется, чтобы множество $\mathscr{D}$ было компактным (т. е. ограниченным замкнутым множеством в $\mathbf{R}^n$).

Заметим, что можно довольствоваться условием ограниченности множества $\mathscr{D}$, если функция f удовлетворяет условию типа $f(x) \to +\infty$ при $\|x\| \to +\infty$.

Но это условие не является достаточным.

Пример. Рассмотрим в $\mathbf{R}^2$ пример, представленный на рис. 10. Область $\mathscr{D}$ ограничений есть треугольник, определенный неравенствами

$$0 \leqslant x_1 \leqslant 1;\quad x_2 \geqslant 0;\quad x_1 - x_2 \geqslant 0.$$

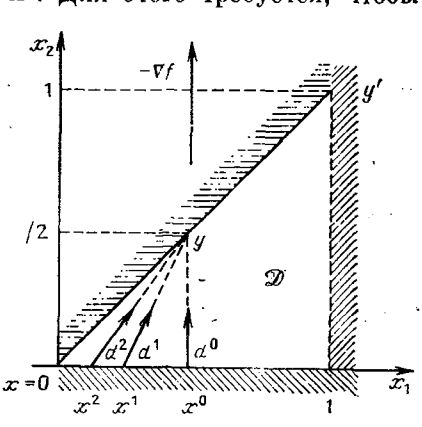


Рис. 10. Пример, в котором отображение $U_{\mathscr{D}}$ не замкнуто

Минимизируемая функция f имеет следующий вид: $f(x_1, x_2) = -x_2$.

Рассмотрим последовательность точек $\{x^k\}$ вида

$$x^0 = \left(\frac{1}{2}, 0\right), \quad x^1 = \left(\frac{1}{4}, 0\right), \quad x^2 = \left(\frac{1}{8}, 0\right), \ldots$$

и соответствующие направления d^k, для которых полупрямая, определяемая парой (x^k, d^k), проходит через точку (1/2; 1/2).

Стало быть, $d^k \to d = (1, 1)$.

Для любого k множество $U_{\mathscr{D}}(x^k, d^k)$ содержит единственную точку $y^k = (1/2, 1/2)$.

Стало быть, $y^k \to y = (1/2, 1/2)$.

Но при этом заметим, что $y \notin U_{\mathscr{D}}(x, d)$, ибо одномерная минимизация с начальной точкой $x=(0, 0)$ в направлении $d=(1, 1)$ приводит к точке $y'=(1,1)$.

Таким образом, в этом случае отображение $U_{\mathscr{D}}$ не замкнуто.

В действительности детальное изучение этого примера (или подобных примеров) показывает, что замкнутость отображения существенным образом зависит от выбора направлений перемещения. Вообще, даже для областей $\mathscr{D}$, обладающих всеми принятыми свойствами регулярности, замкнутость отображения $U_{\mathscr{D}}$ получается лишь при условии выбора конкретных направлений перемещения. Иными словами, условие замкнутости, вообще говоря, будет выполняться не на множестве всех возможных пар (x, d), а лишь на некотором подмножестве произведения $\mathbf{R}^n \times \mathbf{R}^n$.

Следуя [7], назовем множество $\mathscr{R}$ пар (x, d), где $x \in \mathscr{D}$, множеством равномерно возможных направлений, если существует такая постоянная $\eta > 0$, что

$$0 \leqslant \alpha \leqslant \eta \Rightarrow x + \alpha d \in \mathscr{D};$$

постоянная η называется постоянной возможности множества $\mathscr{D}$, соответствующего множеству $\mathscr{R}$ (η зависит от $\mathscr{R}$).

Хорошо понятно, что существование множества равномерно возможных направлений есть условие, накладываемое на область $\mathscr{D}$: произвольная область $\mathscr{D}$ не обязательно удовлетворяет этому условию.

Определим теперь на $\mathscr{R}$ отображение U_η:

$$U_\eta(x, d) = \left\{ y \mid y = x + \alpha d\,(\alpha \geqslant 0);\ y \in \mathscr{D};\ f(y) \leqslant \min_{0 \leqslant \tau \leqslant \eta} f(x + \tau d) \right\}.$$

Теорема 3 [7]. *Если функция f непрерывна, множество $\mathscr{D}$ компактно и существует множество $\mathscr{R}$ равномерно возможных направлений, то отображение U_η замкнуто в каждой точке: $(x, d) \in \mathscr{R}$ $(d \neq 0)$.*

Доказательство. Оно почти совпадает с доказательством теоремы 1 п. 3.1.

Предположим, что $(x^k, d^k) \to (x, d)$ $(k \to \infty)$, $d \neq 0$, и пусть $y^k \to y$ $(y^k \in U_\eta(x^k, d^k))$. В силу компактности множества $\mathscr{D}$ имеем $y^k \in \mathscr{D} \Rightarrow y \in \mathscr{D}$. Покажем, что $y \in U_\eta(x, d)$. Для любого k имеем $y^k = x^k + \alpha^k d^k$; значит,

$$\alpha^k \to \bar{\alpha} = \frac{\|y - x\|}{\|d\|}$$

и, стало быть, y имеет вид $y = x + \bar{\alpha} d$. Для любого τ $(0 \leqslant \tau \leqslant \eta)$ имеем

$$\forall k, \quad f(y^k) \leqslant f(x^k + \tau d^k).$$

В силу непрерывности f, переходя к пределу, получаем

$$f(y) \leqslant f(x + \tau d), \qquad \forall \tau, \qquad 0 \leqslant \tau \leqslant \eta.$$

Отсюда следует, что $y \in U_\eta(x, d)$.

В предыдущем примере мы показали, что

$$y = \left(\frac{1}{2}, \frac{1}{2}\right) \notin U_{\mathscr{D}}(x, d) = \{(1, 1)\}.$$

Заметим, что, напротив, множество $\mathscr{R}$ пар (x^k, d^k) есть множество равномерно возможных направлений с постоянной возможности $\eta = 1/2$ (предполагается, что d^k — векторы с нормой 1).

При этих условиях множество $U_\eta(x, d)$ представляет собой отрезок $[y'', y']$, где $y'' = (\sqrt{2}/4, \sqrt{2}/4)$, $y' = (1, 1)$, и, как легко видеть, $y = (1/2, 1/2) \in U_\eta(x, d)$.

Для построения алгоритмов всегда будет использоваться отображение $U_{\mathscr{D}}$, и при этом мы будем получать схему типа $U_{\mathscr{D}} \circ D$, где D — отображение, которое точке $x \in \mathscr{D}$ ставит в соответствие направление d.

Поскольку $U_{\mathscr{D}}(x, d) \subset U_\eta(x, d)$, то для получения сходимости алгоритма достаточно будет обеспечить, чтобы отображение D в дополнение к обычным условиям (непрерывность, замкнутость) порождало равномерно возможные направления.

СПИСОК ЛИТЕРАТУРЫ

1. Armijo L. (1966), Minimization of functions having continuous partial derivatives, *Pacific J. Math.*, vol. 16, p. 1-3.
2. Goldstein A. A. (1967), *Constructive real analysis*, Harper, New York.
3. Goldstein A. A. (1977), Optimization of Lipschitz Continuous Functions, *Mathematical Programming*, 13, 1, p. 14-22.
4. Huang H. Y., Chamblis J. P. (1973), Quadratically Convergent Algorithms and One-Dimensional Search Schemes, *J. Optimization Theory and Appl.*, 11, p. 175-188.
5. Kiefer J. (1953), Sequential minimax search for a maximum, *Proc. Am. Math. Soc.*, 4, p. 502-506.
6. Lemarechal C. (1980), A view of line-searches, *First Franco-German Conference on optimization and optimal control, Oberwolfach, March* 1980.
7. Luenberger D. G. (1973), *Introduction to linear and nonlinear programming*, Addison Wesley.
8. Powell M. J. D. (1976), Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in: *Nonlinear Programming*, SIAM-AMS Proceedings 9, R. W. Cottle et C. E. Lemke, Eds., Providence R. I.
9. Sugie N. (1964), An Extension of Fibonaccian searching to multidimensional cases, *I. E. E. E. Trans. Autom. Control* AC-9, p. 105.
10. Wolfe P. (1969), Convergence conditions for ascent methods, *S.I.A.M. Review* 11, p. 226-235.

ГЛАВА 4
НЕЛИНЕЙНАЯ ОПТИМИЗАЦИЯ БЕЗ ОГРАНИЧЕНИЙ

§ 1. Введение. Условия оптимальности

Задача, которую мы здесь изучаем, состоит в отыскании минимума (максимума) действительной функции f переменных $x_1, \ldots, x_n$, каждая из которых может принимать любое значение от $-\infty$ до $+\infty$.

Многие задачи могут быть приведены к этому виду. С другой стороны, в задачах, где переменные $x_1, \ldots, x_n$ подчинены дополнительным условиям (типа $g_i(x) \leqslant 0$, $i = 1, \ldots, m$), как мы увидим (см. гл. 6, § 1), можно при некоторых условиях свести все к оптимизационным задачам без ограничений.

Пусть отображение f: $\mathbf{R}^n \to \mathbf{R}$ каждому $x \in \mathbf{R}^n$ $(x = (x_1, \ldots, x_n)^T)$ ставит в соответствие действительное значение

$$f(x) = f(x_1, x_2, \ldots, x_n).$$

Требуется решить задачу

$$\min f(x),$$
$$x \in \mathbf{R}^n.$$

Стало быть, речь идет о нахождении такой точки $x^* \in \mathbf{R}^n$, что

$$\forall x \in \mathbf{R}^n: f(x^*) \leqslant f(x), \qquad (1)$$

т. е. об отыскании точки глобального минимума функции f на $\mathbf{R}^n$.

Если строгое неравенство $f(x^*) < f(x)$ выполняется для любого $x \in \mathbf{R}^n$ $(x \neq x^*)$, то x^* — единственная точка глобального минимума (см. рис. 1).

Однако существуют ситуации, в которых имеет место неединственность глобального минимума (см. рис. 2)

Для многих оптимизационных задач без ограничения основные известные методы не дают возможности определить глобальный минимум: тогда приходится ограничиваться локальными минимумами, т. е. (см. гл. 1) точками, удовлетворяющими (1) только в некоторой окрестности точки x^*. Покажем теперь, как такие точки могут быть охарактеризованы.

1.1. Необходимые условия локальной оптимальности. Мы предполагаем здесь, что $f(x)$ непрерывна и имеет непрерывные первые и вторые частные производные $\partial f/\partial x$ и $\partial^2 f/\partial x_i \partial x_j$ в любой точке $x \in \mathbf{R}^n$. Тогда справедлива

Теорема 1. *Для того чтобы точка x^* была минимумом (локальным или глобальным) функции f, необходимо, чтобы:*

а) $\nabla f(x^*) = 0$ (*условие стационарности*);

b) *гессиан*

$$\nabla^2 f(x^*) = [\partial^2 f/\partial x_i \partial x_j (x^*)]$$

был положительно полуопределенной матрицей *).

Доказательство. Пусть x^* — точка локального минимума функции f. Поскольку f дважды непрерывно дифференцируема,

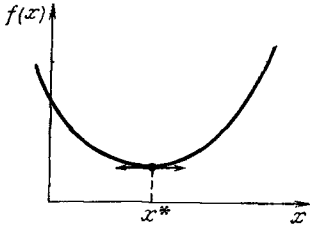


Рис. 1. Эта функция имеет единственный глобальный оптимум x^*

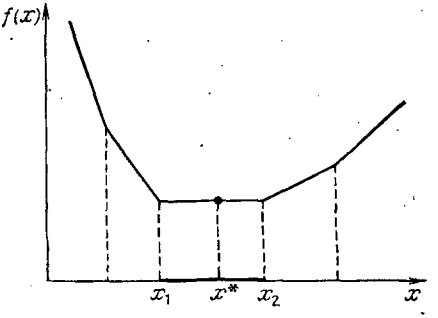


Рис. 2. Эта функция имеет неединственный глобальный оптимум: все точки отрезка $[x_1, x_2]$ являются точками глобального оптимума

то разложение Тейлора в окрестности точки x^* имеет вид

$$f(x) = f(x^*) + \nabla f^T(x^*)(x - x^*) + \\ + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 O(x - x^*),$$

причем $O(x - x^*) \to 0$ при $x \to x^*$.

Если $\nabla f(x^*) \neq 0$, то, выбрав $x = x^* - \theta \nabla f(x^*)$, для достаточно малого $\theta > 0$ имеем $f(x) < f(x^*)$, но это противоречит предположению, что x^* есть локальный минимум. Значит, условие а) необходимо, и мы можем записать:

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 O(x - x^*).$$

Если матрица $\nabla^2 f(x^*)$ не является положительно полуопределенной, то существует такой вектор $d \in \mathbf{R}^n$ $(d \neq 0)$, что

$$d^T \nabla^2 f(x^*) d < 0.$$

Выбрав тогда $x = x^* + \theta d$ для достаточно малого $\theta > 0$, имеем $f(x) < f(x^*)$, но это снова противоречит предположению о локальной оптимальности точки x^*.

Итак, условие b) тоже необходимо.

Точка x^*, удовлетворяющая условию а), т. е. условию $\partial f/\partial x_i (x^*) = 0$ $(i = 1, \ldots, n)$, называется стационарной точкой.

Рис. 3 иллюстрирует тот факт, что стационарность не является достаточным условием локальной оптимальности.

*) Напомним условие положительной полуопределенности: $y^T \nabla^2 f(x^*) y \geqslant 0$ $\forall y \in \mathbf{R}^n$.

1.2. Достаточные условия локальной оптимальности. Примем предположения п. 1.1.

Теорема 2. *Для того чтобы точка x^* была локальным оптимумом функции f на $\mathbf{R}^n$, достаточно, чтобы:*

a) $\nabla f(x^*) = 0$ (*стационарная точка*);

b) *гессиан $\nabla^2 f(x^*)$ был положительно определенной матрицей* *).

Доказательство. Рассмотрим точку x^*, удовлетворяющую условиям a) и b). Тогда разложение Тейлора функции f в окрестности точки x^* будет иметь вид

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 O(x - x^*),$$

причем $O(x - x^*) \to 0$ при $x \to x^*$.

Для любого направления перемещения $d \in \mathbf{R}^n$ ($\|d\| = 1$) имеем тогда

$$f(x^* + \theta d) = f(x^*) + \frac{\theta^2}{2} d^T \nabla^2 f(x^*)\, d + \theta^2 O(\theta).$$

В силу условия b) $d^T \nabla^2 f(x^*) d > 0$; значит, при достаточно малом θ имеет место неравенство $f(x^* + \theta d) > f(x^*)$.

Тем самым показано, что x^* есть локальный минимум функции f.

Заметим, что условие b) теоремы 2 сводится к предположению, что функция f строго выпукла в окрестности точки x^*.

Рис. 3 иллюстрирует случай, когда достаточные условия оптимальности не выполняются: из-за того что f имеет в x^* точку перегиба, гессиан является не положительно определенным, а лишь положительно полуопределенным ($d^2f/dx^2(x^*) = 0$).

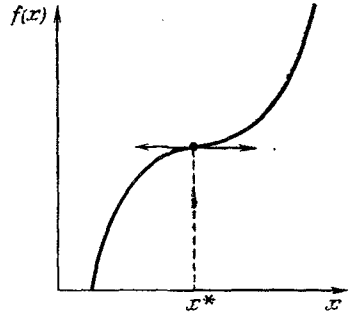


Рис. 3. Точка x^* стационарна, но не является точкой локального оптимума

1.3. Случай выпуклых функций. Необходимые и достаточные условия существования глобального оптимума. В случае выпуклой функции f, определенной на $\mathbf{R}^n$, было показано (гл. 1, теорема 8), что для того, чтобы точка x^* была глобальным минимумом функции f, необходимо и достаточно, чтобы 0 был субградиентом функции f в x^*.

Таким образом, для непрерывно дифференцируемой функции справедлива

Теорема 3. *Пусть f — выпуклая непрерывно дифференцируемая функция. Для того чтобы точка x^* была глобальным оптимумом функции f на $\mathbf{R}^n$, необходимо и достаточно, чтобы $\nabla f(x^*) = 0$.*

*) Напомним условие положительной определенности: $\forall y \in \mathbf{R}^n$, $y \neq 0$: $y^T \nabla^2 f(x^*) y > 0$.

Иными словами, в выпуклом случае стационарность, и только она, составляет необходимое и достаточное условие глобальной оптимальности.

1.4. Случай произвольных функций. Трудности общей задачи. Условия пп. 1.1, 1.2 практически являются самыми общими условиями оптимальности, известными для произвольных непрерывно дифференцируемых функций в $\mathbf{R}^n$.

Если минимизируемая функция выпукла, удается пользоваться аналогом дифференцируемости в каждой точке при помощи понятия субградиента, введенного в гл. 1, п. 3.5. Однако этот частный случай очень важен, ибо он возникает при решении многих задач в математическом программировании; например, в комбинаторной оптимизации двойственная задача для целочисленной задачи состоит в оптимизации выпуклой или вогнутой функции, не являющейся всюду дифференцируемой (см. п. 3.1 этой главы и гл. 7); в области декомпозиции больших систем решение главной задачи (см. гл. 8) во многих случаях сводится к оптимизации выпуклых или вогнутых функций, не являющихся всюду дифференцируемыми. Для изучения алгоритмов, применимых к задачам указанного типа, отсылаем к § 3 этой главы.

Однако существует много функций, которые, хотя и непрерывны, *не являются ни выпуклыми, ни субдифференцируемыми*; имеется много функций, которые даже не удовлетворяют условию непрерывности. Самая общая задача оптимизации без ограничений для таких функций в $\mathbf{R}^n$ снова очень далека от того, чтобы быть решенной удовлетворительным образом.

§ 2. Численные методы для оптимизации дифференцируемых функций

Здесь мы предполагаем, что функция f непрерывна и имеет непрерывные первые производные. Поскольку во всех случаях необходимым условием оптимальности является условие стационарности функции f, то практически все методы оптимизации без ограничений в $\mathbf{R}^n$ состоят в отыскании стационарной точки x^* $(\nabla f(x^*) = 0)$.

Эта задача эквивалентна решению нелинейной системы уравнений

$$\frac{\partial f}{\partial x_i}(x) = 0, \qquad i = 1, \ldots, n. \tag{2}$$

Можно отыскивать непосредственно решение этой системы, что приводит к методу Ньютона (п. 2.8). Однако метод может не сходиться, если начальная точка для итераций слишком далека от x^*. С другой стороны, он предполагает, что функция дважды непрерывно дифференцируема, и требует вычисления в каждой точке вторых производных.

Поэтому наиболее часто используемые методы проводятся различно: речь идет об итерационных процедурах, порождающих по-

следовательность точек x^0, x^1, ..., x^k, сходящуюся к локальному оптимуму функции f.

На каждом k-м этапе x^{k+1} определяется как $x^{k+1} = x^k + \lambda_k d_k$, где d_k — направление перемещения, которое может быть:

— либо градиентом функции f в x^k: $d_k = -\nabla f(x^k)$;

— либо вычисленным исходя из градиента $\nabla f(x^k)$;

— либо выбранным более или менее произвольно, при условии, что это будет направление спуска, т. е. $\nabla f^T(x^k) d_k < 0$.

2.1. Методы градиента. Градиент с заранее заданным шагом. Речь идет о семействе методов, которые действуют следующим образом.

Исходя из точки x^0, вычисляем градиент $\nabla f(x^0)$ в точке x^0. Поскольку $\nabla f(x^0)$ указывает направление наибольшего возрастания функции f, снабжаем направление, противоположное градиенту, величиной λ_0 и находим точку

$$x^1 = x^0 - \lambda_0 \frac{\nabla f(x^0)}{\| \nabla f(x^0) \|}.$$

Повторение процедуры дает точки, удовлетворяющие соотношениям

$$x^{k+1} = x^k - \lambda_k \frac{\nabla f(x^k)}{\| \nabla f(x^k) \|}, \quad \text{где} \quad \forall k, \quad \lambda_k > 0.$$

В этом семействе методов надлежит выделить методы градиента с заданным шагом, в которых заранее задаются значения перемещений λ_k. В [63] изучалась сходимость этой итерационной схемы и доказано (см. теорему 9 п. 3.4), что $x^k \to x^*$, если выполнены только два условия:

$$\lambda_k \to 0, \quad k \to \infty,$$

$$\sum_{k=0}^{\infty} \lambda_k = +\infty.$$

(например, можно выбрать $\lambda_k = 1/k$).

Неудобство этой процедуры (называемой еще «метод расходящегося ряда») состоит в том, что сходимость может оказаться медленной. Основная особенность методов градиента с заданным шагом заключается в возможности их обобщения на случай функций, не являющихся всюду дифференцируемыми. За подробностями отсылаем к § 3, в котором исследуются многие стратегии априорного выбора перемещений λ_k, позволяющих получить более быструю сходимость, чем метод расходящегося ряда.

2.2. Метод наискорейшего спуска [7, 10]. В этом широко используемом методе λ_k выбирается так, чтобы минимизировать функцию от λ:

$$g(\lambda) = f[x^k - \lambda \nabla f(x^k)]$$

на множестве значений $\lambda \geqslant 0$ (одномерная минимизация, см. гл. 3).

Тогда приходим к процедуре следующего типа.

Алгоритм наискорейшего спуска.

a) Выбрать начальную точку x^0;

b) на k-й итерации $d_k = -\nabla f(x^k)$; найти такое λ_k, что

$$f(x^k + \lambda_k d_k) = \min_{\lambda \geqslant 0} \{f(x^k + \lambda d_k)\}.$$

Положить $x^{k+1} = x^k + \lambda_k d_k$;

c) тест на остановку: если выполнен, то конец; иначе выполнить $k \leftarrow k + 1$ и возвратиться к b).

Поскольку сходимость, вообще говоря, не будет конечной, то необходимо определить окончание процесса. Перечислим несколько наиболее употребительных критериев.

1. $\max\limits_{i=1,\ldots,n} \left|\dfrac{\partial f}{\partial x_i}\right| < \varepsilon$ ($\varepsilon > 0$ задано).

2. $\|\nabla f\|^2 = \sum\limits_{i=1}^{n} \left(\dfrac{\partial f}{\partial x_i}\right)^2 < \varepsilon$ ($\varepsilon > 0$ задано).

3. $|f(x^{k+1}) - f(x^k)| < \eta$ ($\eta > 0$ задано).

Для каждого из этих критериев из предосторожности, быть может, следует потребовать, чтобы тест был проверен на p последовательных итерациях (где p — заданное заранее фиксированное число).

Очевидно, возможны и другие, более тонкие критерии. Например, можно попытаться установить либо априори, либо при помощи статистических методов максимальную точность, которая может быть достигнута с учетом арифметики используемого калькулятора. Тогда итерации будут прерываться, как только отклонения в значениях переменных или в значениях функций между двумя (или несколькими) последовательными итерациями становятся величинами того же порядка, что и максимальная точность. (В [79, 80] можно найти систематизированные методы оценки точности числовых расчетов на компьютере.)

Относительно глобальной сходимости метода наискорейшего спуска может быть сформулирована

Теорема 4. *Если непрерывно дифференцируемая функция f удовлетворяет условию $f(x) \to +\infty$ при $\|x\| \to +\infty$, то для любой начальной точки x^0 метод наискорейшего спуска (с точной или приближенной одномерной оптимизацией) сходится к стационарной точке функции f.*

Доказательство. Алгоритм наискорейшего спуска может быть представлен в виде произведения двух отображений D и U:

— отображение D точке x^k ставит в соответствие направление перемещения $d^k = -\nabla f(x^k)$;

— многозначное отображение U представляет собой процесс одномерной оптимизации: паре (x^k, d^k) ставится в соответствие $x^{k+1} \in U(x^k, d^k)$.

Если функция f непрерывно дифференцируема, то отображение D непрерывно (речь идет об «обычном» отображении); с другой стороны, в силу непрерывности функции f отображение U замкнуто (теорема 1 гл. 3 о точной одномерной оптимизации; теорема 2 гл. 3 о приближенной одномерной оптимизации).

Стало быть, согласно предложению 3 гл. 1, отображение $U \circ D$ замкнуто. При этом из свойства возрастания на бесконечности следует, что все точки x^k содержатся в ограниченном замкнутом множестве.

Тогда, принимая функцию f в качестве функции спуска, а множество Ω — в качестве множества стационарных точек функции f, можно применить теорему о сходимости из гл. 1. Отсюда следует глобальная сходимость метода к стационарной точке.

Замечание 1. Напомним, что глобальная сходимость не означает, что обязательно должен быть получен глобальный оптимум функции f. Если f непрерывно дифференцируема, то все, что можно сказать,— что мы получаем стационарную точку $\bar{x}$ функции f. Если f дважды дифференцируема в $\bar{x}$ и гессиан $\nabla^2 f(x)$ положительно определен, то $\bar{x}$ есть локальный минимум функции f.

И лишь в весьма частных случаях (например, f — выпуклая дифференцируемая функция) можно быть уверенным в получении глобального минимума функции f.

Замечание 2. Основной недостаток метода наискорейшего спуска заключается в том, что для некоторых типов функций схо-

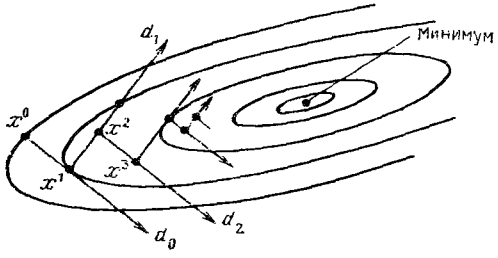


Рис. 4. Два последовательных направления перемещения в методе наискорейшего спуска ортогональны

димость может оказаться медленной. В самом деле, если число λ_k минимизирует функцию

$$g(\lambda) = f(x^k + \lambda d_k),$$

то должно быть

$$\frac{dg}{d\lambda}(\lambda_k) = d_k^T \nabla f(x^k + \lambda_k d_k) = d_k^T \nabla f(x^{k+1}) = 0,$$

откуда вытекает равенство

$$d_k^T d_{k+1} = 0,$$

которое доказывает, что последовательные направления перемещения ортогональны (рис. 4).

Таким образом, число итераций, необходимых для минимизации плохо обусловленных функций «овражного» типа (рис. 5), может оказаться значительным. Мы вернемся к этому явлению в п. 2.12 в связи с изучением скорости сходимости.

2.3. Ускоренные методы наискорейшего спуска. Чтобы избежать искажения свойств методов наискорейшего спуска при их применении к плохо обусловленным функциям («овражного» типа), предложен (см., например, [19, 50]) следующий процесс ускорения.

На каждой k-й итерации (с началом в x^k) осуществляется p этапов метода наискорейшего спуска, что дает точку y^k. Тогда по-

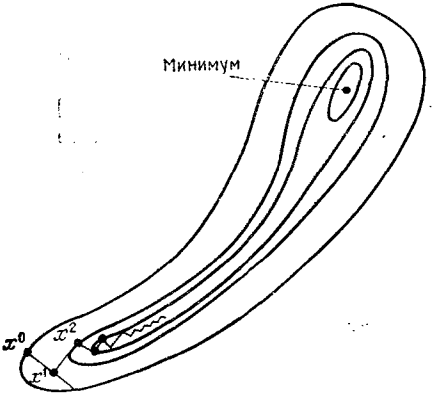


Рис. 5. Пример, иллюстрирующий медленную сходимость методов наискорейшего спуска

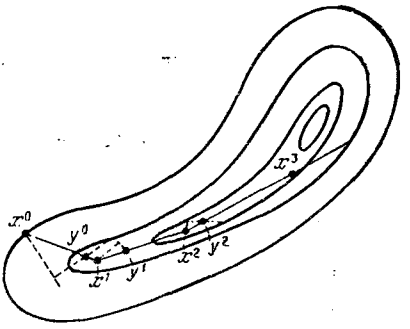


Рис. 6. Ускоренный метод наискорейшего спуска порядка $p = 2$

лучаем точку x^{k+1}, определенную одномерной минимизацией в направлении $d_k = y^k - x^k$ с началом в x^k (ускоренный метод p-го порядка).

Легко убедиться на примере размерности два в преимуществах метода порядка $p = 2$ (рис. 6).

Вообще же выгоды «ускоренных» методов наискорейшего спуска по отношению к «обычному» могут быть выявлены при помощи анализа асимптотической сходимости (см. п. 2.12).

В частности, для $p = n$ показано, что каждая итерация процесса p-го порядка требует $p + 1$ одномерных минимизаций. Значит, речь идет о методах, вообще говоря, довольно дорогостоящих (по числу оценок функции или ее градиента), поэтому они сравнительно мало используются.

При тех же условиях глобальная сходимость «ускоренных» методов наискорейшего спуска вытекает из свойства глобальной сходимости «обычного» метода наискорейшего спуска.

2.4. Методы второго порядка. Для улучшения сходимости градиентных методов созданы новые, более тонкие методы. В их основе лежит следующий факт.

Поскольку в точке (локального) минимума x^* имеем $\nabla f(x^*) = 0$, то первый член разложения функции f в ряд Тейлора в окрестности точки x^* имеет вид

$$f(x) \simeq f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*),$$

где гессиан функции f в точке x^* (если она удовлетворяет достаточным условиям теоремы 2 п. 1.2) положительно определен.

Таким образом, функция f ведет себя в окрестности точки x^* как строго выпуклая квадратичная функция. Отсюда следует, что общий метод минимизации, чтобы быть эффективным, должен по крайней мере быстро сходиться на квадратичных функциях. В самом деле, в противном случае он станет медленно сходящимся и неэффективным, как только мы попадем в окрестность точки оптимума.

2.5. Методы сопряженных направлений. Общий принцип. Речь идет об итерационных методах, применяемых к квадратичной функции n переменных и приводящих к оптимуму не более чем в n этапов.

Рассмотрим произвольную квадратичную функцию

$$q(x) = \frac{1}{2} x^T A x + b^T x + c,$$

где A — положительно определенная симметричная матрица порядка n; b есть n-вектор $(b_1, \ldots, b_n)^T$; c — некоторая константа.

Суть методов сопряженных направлений состоит в том, чтобы, начиная от точки x^0, минимизировать последовательно $q(x)$ по n линейно независимым направлениям $d_0, d_1, \ldots, d_{n-1}$, обладающим свойством быть попарно сопряженными относительно квадратичной формы $q(x)$. Напомним, что это записывается в виде

$$\left.\begin{array}{l} \forall i \quad (0 \leqslant i \leqslant n-1) \\ \forall j \quad (0 \leqslant j \leqslant n-1) \\ \qquad i \neq j \end{array}\right\} \Rightarrow d_i^T A d_j = 0. \tag{3}$$

Итак, предположим, что точки x^{k+1} $(k = 0, 1, \ldots, n-2)$ определены, начиная с x^k, формулой

$$x^{k+1} = x^k + \lambda_k d_k,$$

где λ_k — значение величины λ, минимизирующее $q(x^k + \lambda d_k)$.

Мы покажем, что при этих условиях точка, полученная на n-м этапе, т. е. точка

$$x^n = x^0 + \sum_{j=0}^{n-1} \lambda_j d_j,$$

непременно будет оптимумом задачи, т. е. удовлетворяет соотношениям

$$Ax^n + b = \nabla q(x^n) = 0. \tag{4}$$

А поскольку λ_k минимизирует q в направлении d_k, то

$$d_k^T \nabla q(x^{k+1}) = d_k^T (Ax^{k+1} + b) = 0 \quad \forall k.$$

Пусть

$$d_k^T A(x^k + \lambda_k d_k) + d_k^T b = 0,$$

откуда получаем

$$\lambda_k = -\frac{d_k^T (Ax^k + b)}{d_k^T A d_k} \tag{5}$$

(заметим, что в силу положительной определенности матрицы знаменатель не может обращаться в нуль).

Учитывая, что

$$x^k = x^0 + \sum_{j=0}^{k-1} \lambda_j d_j,$$

можем снова написать

$$d_k^T A x^k = d_k^T A x^0 + \sum_{j=0}^{k-1} \lambda_j d_k^T A d_j = d_k^T A x^0,$$

и, значит, λ_k определяется выражением

$$\lambda_k = -\frac{d_k^T (Ax^0 + b)}{d_k^T A d_k}. \tag{6}$$

Теперь мы можем доказать свойство, характеризующее все методы сопряженных направлений.

Свойство 1. *Для любого* k $(1 \leqslant k \leqslant n)$ *точка*

$$x^k = x^0 + \sum_{j=0}^{k-1} \lambda_j d_j$$

есть оптимум функции $q(x)$, *определенной на аффинном многообразии* V^k, *порожденном направлениями* $(d_0, d_1, \ldots, d_{k-1})$ *и проходящем через* x^0.

В частности, $x^n = x^0 + \sum_{j=0}^{n-1} \lambda_j d_j$ *есть оптимум функции* $q(x)$ *на* $\mathbf{R}^n$.

Доказательство. Заметим, что

$$d_i^T A x^k = d_i^T A x^0 + \sum_{j=0}^{k-1} \lambda_j d_i^T A d_j = d_i^T A x^0 + \lambda_i d_i^T A d_i, \qquad 0 \leqslant i \leqslant k-1,$$

отсюда, используя (6), получаем

$$d_i^T A x^k = d_i^T A x^0 - d_i^T (Ax^0 + b),$$

и, значит,

$$\forall i = 0, 1, \ldots, k-1: \; d_i^T (Ax^k + b) = 0.$$

Это показывает, что $\nabla q(x^k) = Ax^k + b$ ортогонально многообразию V^k, и, стало быть, x^k есть оптимум функции q на V^k.

В частности, при $k = n$ имеем $V^k = \mathbf{R}^n$ (ибо $d_0, d_1, \ldots, d_{n-1}$ линейно независимы), и x^n есть оптимум для q на всем пространстве $\mathbf{R}^n$.

Следовательно, при поиске минимума квадратичной функции в $\mathbf{R}^n$ метод сопряженных направлений сходится конечным образом не более чем за n этапов.

Для выбора направлений $(d_0, d_1, \ldots, d_{n-1})$ существует несколько степеней свободы, поэтому можно представить себе многочисленные алгоритмы, основанные на методе сопряженных направлений для квадратичных функций и на методе Флетчера — Ривза для произвольных функций.

2.6. Метод сопряженного градиента для квадратичных функций. Мы предполагаем здесь, что минимизируемая функция квадратична и имеет вид

$$q(x) = \frac{1}{2} x^T A x + b^T x + c.$$

Идея метода состоит в последовательном построении направлений $d_0, d_1, \ldots, d_{n-1}$, взаимно сопряженных относительно матрицы A квадратичной формы. На каждом k-м шаге направление d_k получается как линейная комбинация градиента $-\nabla q(x^k)$ в x^k и предшествующих направлений $(d_0, d_1, \ldots, d_{k-1})$, причем коэффициенты линейной комбинации выбираются так, чтобы d_k было сопряженным ко всем предшествующим направлениям.

Обозначим через $g_k = \nabla q(x^k)$ градиент функции q в x^k. Тогда метод можно записать в следующем виде.

Алгоритм сопряженного градиента для квадратичных функций.

a. Пусть x^0 — начальная точка, $g_0 = \nabla q(x^0) = Ax^0 + b$; положить $d_0 = -g_0$, $k = 0$.

b. Определить $x^{k+1} = x^k + \lambda_k d_k$, где

$$\lambda_k = -\frac{g_k^T d_k}{d_k^T A d_k}, \tag{7}$$

затем

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{8}$$

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}. \tag{9}$$

Положить $k \leftarrow k + 1$ и вернуться в b.

Заметим, что выражение (7) идентично (5) из п. 2.5. Значит, для доказательства справедливости алгоритма (его сходимости за n шагов) достаточно убедиться в том, что направления, порожденные соотношениями (8), (9), последовательно сопряжены.

Теорема 5. *На любой k-й итерации алгоритма, где оптимум еще не достигается* (т. е. $g_i \neq 0$; $i = 0, 1, \ldots, k$), *имеем*:

а)
$$\lambda_k = \frac{g_k^T g_k}{d_k^T A d_k} \neq 0; \tag{10}$$

b)
$$\beta_k = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k} = \tag{11}$$

$$= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}; \tag{12}$$

c) *направления* $d_0, d_1, \ldots, d_{k+1}$, *порожденные алгоритмом, попарно сопряжены.*

Доказательство. Проведем индукцию по k, предположив, что $d_0, d_1, \ldots, d_{k+1}$ попарно сопряжены.

a) Прежде всего докажем эквивалентность (7) и (10). Имеем $d_k = -g_k + \beta_{k-1} d_{k-1}$. Значит, (7) записывается в виде

$$\lambda_k = \frac{g_k^T g_k}{d_k^T A d_k} - \beta_{k-1} \frac{g_k^T d_{k-1}}{d_k^T A d_k}.$$

Поскольку $(d_0, d_1, \ldots, d_{k-1})$ попарно сопряжены, то x^k есть оптимум для $q(x)$ на многообразии V^k (с начальной точкой x^0), порожденном $(d_0, d_1, \ldots, d_{k-1})$ (см. п. 2.5).

b) Для доказательства (11) заметим, что

$$g_{k+1} - g_k = A(x^{k+1} - x^k) = \lambda_k A d_k.$$

Тогда

$$g_{k+1}^T A d_k = \frac{1}{\lambda_k} g_{k+1}^T [g_{k+1} - g_k]$$

и при помощи (10) получаем

$$\beta_k = \frac{g_{k+1}^T [g_{k+1} - g_k]}{g_k^T g_k},$$

что и доказывает (11).

Теперь (12) следует из того, что $g_{k+1}^T g_k = 0$, ибо $g_k = d_k - \beta_{k-1} d_{k-1}$ принадлежит подпространству, порожденному $(d_0, d_1, \ldots, d_k)$, и из того, что g_{k+1} ортогонально этому подпространству (см. п. 2.5).

c) Покажем, наконец, что направление d_{k+1} сопряжено с $d_0, d_1, \ldots, d_k$. Имеем $d_{k+1}^T A d_k = 0$, ибо при помощи (8) получаем (с учетом определения β_k)

$$(-g_{k+1} + \beta_k d_k)^T A d_k = -g_{k+1}^T A d_k + \beta_k d_k^T A d_k = 0.$$

Теперь убедимся в том, что

$$d_{k+1}^T A d_i = 0, \qquad i = 0, 1, \ldots, k-1.$$

Имеем

$$d_{k+1}^T A d_i = -g_{k+1}^T A d_i + \beta_k d_k^T A d_i.$$

По предположению индукции второе слагаемое обращается в нуль.

Покажем, что первое — тоже нуль. Поскольку $x^{i+1} = x^i + \lambda_i d_i$ и $\lambda_i \neq 0$, то

$$Ad_i = \frac{1}{\lambda_i}(Ax^{i+1} - Ax^i) = \frac{1}{\lambda_i}(g_{i+1} - g_i).$$

Записывая

$$g_{i+1} = -d_{i+1} + \beta_i d_i,$$
$$g_i = -d_i + \beta_{i-1} d_{i-1},$$

видим, что Ad_i есть линейная комбинация направлений d_{i+1}, d_i и d_{i-1} (при $i = 0$ имеем $g_0 = -d_0$, и, значит, Ad_0 есть линейная комбинация лишь d_1 и d_0).

Но в силу попарной сопряженности направлений $(d_0, d_1, \ldots, d_k)$, как известно (см. п. 2.5), точка x^{k+1} есть оптимум функции $q(x)$ на многообразии V^{k+1}, порожденном $(d_0, d_1, \ldots, d_k)$. Следовательно, g_{k+1} ортогонально подпространству, порожденному $(d_0, d_1, \ldots, d_k)$, а поскольку Ad_i принадлежит этому подпространству при $i = 0, 1, \ldots, k-1$, то отсюда получаем $g_{k+1}^{\prime} Ad_i = 0$, и доказательство завершено.

2.7. Случай произвольных функций. Метод Флетчера — Ривза (и вариант Полака — Рибьера). Метод Флетчера — Ривза [18] есть прямое расширение предшествующего метода на случай произвольных функций. В применении к квадратичной функции он становится равносильным методу сопряженного градиента. Этот метод очень интересен, с одной стороны, тем, что он требует хранения в памяти малого количества информации (по существу, нужны три n-мерных вектора); с другой стороны, его скорость сходимости значительно превышает таковую для классических градиентных алгоритмов (см. п. 2.12). При этом квазиньютоновские методы (см. пп. 2.9—2.11), которые со строгой точки зрения асимптотической сходимости, вообще говоря, лучше (см. п. 2.12), оказываются неудобными тем, что они требуют гораздо больше вычислений на каждой итерации и больше места в памяти.

Метод Флетчера — Ривза.

a. Этап 0: x^0 есть выбранная начальная точка; положить $d_0 = -\nabla f(x^0)$.

b. Этап k: выбрать λ_k минимизирующим функцию

$$g(\lambda) = f(x^k + \lambda d_k),$$

положить

$$x^{k+1} = x^k + \lambda_k d_k,$$
$$d_{k+1} = -\nabla f(x^{k+1}) + \beta_k d_k,$$
$$\beta_k = \frac{\|\nabla f(x^{k+1})^2\|}{\|\nabla f(x^k)\|^2}.$$

c. Тест на остановку; если выполнено, то конец.

Иначе: положить $k \leftarrow k+1$ и вернуться к b).

Ясно, что на каждом этапе мы сохраняем в памяти лишь градиенты в точках x^k и x^{k+1} и текущее направление перемещения d_k.

Другой вариант (Полака — Рибьера) состоит в определении β_k не по формуле (12), а по формуле (11) (см., например, [61]). Оба метода равносильны в случае квадратичных функций, а в применении к произвольным функциям они, вообще говоря, приводят к различным результатам.

Важно заметить, что глобальная сходимость метода Флетчера — Ривза (или Полака — Рибьера) обеспечена лишь в случае процесса с периодической сменой начала. Например, все n итераций будут начинаться из последней полученной точки с градиентом в этой точке в качестве направления перемещения. Тогда глобальная сходимость этой процедуры следует из глобальной сходимости методов наискорейшего спуска.

2.8. **Метод Ньютона.** Предположим, что функция f дважды непрерывно дифференцируема и что мы умеем вычислять все ее вторые производные.

Идея заключается в замене функции f в окрестности текущей точки x^k ее квадратичной аппроксимацией:

$$q(x) = f(x^k) + \nabla f^T(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k).$$

Возьмем в качестве точки x^{k+1} точку минимума функции $q(x)$, если таковой существует. Это может быть лишь в случае, когда $\nabla^2 f(x^k)$ будет положительно определенной матрицей. Тогда функция $q(x)$ строго выпукла и имеет единственную точку минимума x^{k+1}, определяемую равенством $\nabla q(x^{k+1}) = 0$.

Это приводит к линейной системе

$$\nabla f(x^k) = -\nabla^2 f(x^k)(x^{k+1} - x^k),$$

из которой получаем итерационную формулу

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

Эта формула есть не что иное, как *метод Ньютона* в применении к решению системы нелинейных уравнений

$$\frac{\partial f}{\partial x_i}(x) = 0, \qquad i = 1, \ldots, n.$$

Заметим, что здесь как направление, так и шаг перемещения фиксированы.

Метод интересен тем, что в применении к строго выпуклой квадратичной функции он сходится за одну-единственную итерацию. Однако когда мы хотим использовать его для произвольной функции, возникают трудности, существенным образом из-за того, что метод не обладает свойством глобальной сходимости: если начальная точка x^0 слишком далека от x^*, то метод не сходится (см., например, гл. 3, п. 1.1).

Стало быть, чтобы избавиться от этих трудностей, надо ввести некоторые модификации. Прежде всего, поскольку аппроксимация $f(x)$ функцией $q(x)$ имеет место только в окрестности точки x^k,

можно воздействовать на шаг перемещения, используя итерационную формулу

$$x^{k+1} = x^k - \lambda_k [\nabla^2 f(x^k)]^{-1} \nabla f(x^k),$$

где λ_k — скаляр, выбранный, например, так, чтобы норма $\|x^{k+1} - x^k\|$ была не слишком велика. Можно также выбрать его так, чтобы f^{k+1} была точкой минимума функции $g(\lambda) = f(x^k + \lambda d_k)$ в направлении

$$d_k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

Другой способ определения шага перемещения состоит в пробе значения $\lambda_k = 1$. Если $f(x^k + d_k) \geqslant f(x^k)$, то λ^*, минимизирующее функцию $g(\lambda) = f(x^k + \lambda d_k)$, разумеется, содержится между 0 и 1 (так как матрица $\nabla^2 f(x^k)$ положительно определена, d_k есть направление спуска, т. е. $dg/d\lambda(0) < 0$).

Тогда можно применять метод дихотомии, до тех пор пока не найдем такое λ, чтобы $f(x^k + \lambda d_k) < f(x^k)$.

Это замечание применимо равным образом к квазиньютоновским методам, изучаемым ниже (пп. 2.9—2.11).

Другая трудность может проявиться, когда гессиан $\nabla^2 f(x^k)$ не является положительно определенным. В самом деле, в этом случае направление перемещения $-[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$ может не быть направлением спуска и глобальная сходимость метода не будет обеспечена. Когда это происходит, некоторые авторы пытаются подвергнуть гессиан $\nabla^2 f(x^k)$ небольшому возмущению, чтобы получить положительно определенную матрицу M_k. Таким образом приходят к итерационной формуле

$$x^{k+1} = x^k - \lambda_k [M_k]^{-1} \nabla f(x^k),$$

где скаляр λ_k получен техникой, описанной выше.

Заметим, что положительная определенность матрицы M_k обеспечивает свойство направления перемещения $d_k = -[M_k]^{-1} \nabla f(x^k)$ быть направлением спуска. В самом деле:

$$\nabla f^T(x^k) d_k = -\nabla f^T(x^k) [M_k]^{-1} \nabla f(x^k) < 0.$$

Чтобы построить M_k, исходя из $\nabla^2 f(x^2)$, можно, например, воспользоваться возмущением типа

$$M_k = \mu_k I + \nabla^2 f(x^k),$$

где $\mu_k > 0$ — некоторый выбранный минимальный скаляр с ограничением, состоящим в том, что все собственные значения матрицы M_k должны быть больше или равны некоторой заданной постоянной $\delta > 0$. Глобальная сходимость этой процедуры легко доказывается (при очень большом μ_k вновь приходим к методу наискорейшего спуска) [50].

2.9. **Квазиньютоновские методы (или методы с переменной метрикой). Общий принцип.** Основу метода существенным образом составляет обобщение итерационной формулы Ньютона (см. п. 2.8)

$$x^{k+1} = x^k - \lambda_k [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

Серьезным ограничением для метода Ньютона, как мы видели выше, является требование, чтобы гессиан ∇^2 был положительно определенным.

Естественное расширение состоит в замене матрицы $[\nabla^2 f(x)]^{-1}$ положительно определенной матрицей H_k, дающей направление перемещения, исходя из градиента $\nabla f(x^k)$. Отсюда получаем итерационную формулу

$$x^{k+1} = x^k - \lambda_k H_k \nabla f(x^k),$$

где λ_k выбирается так, чтобы минимизировать функцию $g(\lambda) = f(x^k + \lambda d_k)$ в направлении $d_k = -H_k \nabla f(x^k)$ (или по крайней мере так, чтобы $g(\lambda_k) < g(0)$).

Очевидно, что матрица H_k на каждой итерации модифицируется так, чтобы для каждой квадратичной функции вида

$$q(x) = \frac{1}{2} x^T A x + b^T x + c$$

(с положительно определенной матрицей A) матрицы H_k сходились к обращению A^{-1} гессиана функции q.

Следовательно, на конечном этапе сходимости мы вновь придем к методу Ньютона.

Если метод применяется к произвольной функции, то H_k может рассматриваться в каждый момент как аппроксимация (положительно определенная) обращения гессиана функции f.

Существует, очевидно, много различных вариантов выбора формулы представления матрицы H_k. Вообще говоря, требуется, чтобы выполнялось соотношение

$$H_k[\nabla f(x^k) - \nabla f(x^{k-1})] = x^k - x^{k-1}. \qquad (13)$$

Формула коррекции, позволяющая получить матрицу H_{k+1}, исходя из матрицы H_k, использует новую информацию, полученную на k-м шаге алгоритма, т. е. существенно использует градиент $\nabla f(x^{k+1})$ в точке x^{k+1} (получаемый, вообще говоря, методом одномерного поиска в направлении $d_k = -H_k \nabla f(x^k)$ с начальной точкой x^k).

Предлагались различные формулы коррекции типа

$$H_{k+1} = H_k + \Delta_k.$$

В зависимости от того, имеет матрица Δ_k ранг 1 или ранг 2, мы будем говорить о коррекции ранга 1 или ранга 2. Одна из первых корректирующих формул, используемых для построения аппроксимации обращения гессиана, состоит в выборе матрицы (ранга 1) вида $\Delta_k = \alpha_k u_k u_k^T$, где u_k, α_k — соответственно вектор и скаляр, выбранные так, чтобы выполнялось соотношение

$$H_{k+1}[\nabla f(x^{k+1}) - \nabla f(x^k)] = x^{k+1} - x^k.$$

Заметим, что если матрица H_0 симметрична, то приведенная выше коррекция сохраняет симметрию матриц H_k.

Положив

$$\delta_k = x^{k+1} - x^k,$$
$$\gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k),$$

покажем, как можно определить α_k и u_k, чтобы $H_{k+1}\gamma_k = \delta_k$.

Пусть снова

$$\left[H_k + \alpha_k \left(u_k u_k^T\right)\right] \gamma_k = \delta_k.$$

Умножив скалярно обе части на γ_k, получим

$$\gamma_k^T H_k \gamma_k + \alpha_k \left(\gamma_k^T u_k\right)\left(u_k^T \gamma_k\right) = \gamma_k^T \delta_k.$$

Пусть

$$\alpha_k \left(u_k^T \gamma_k\right)^2 = \gamma_k^T \left(\delta_k - H_k \gamma_k\right).$$

Используя тождество

$$\alpha_k \left(u_k u_k^T\right) = \frac{\left(\alpha_k u_k u_k^T \gamma_k\right)\left(\alpha_k u_k u_k^T \gamma_k\right)^T}{\alpha_k \left(u_k^T \gamma_k\right)^2}$$

и осуществив замену

$$\alpha_k u_k u_k^T \gamma_k \quad \text{на} \quad \delta_k - H_k \gamma_k,$$
$$\alpha_k \left(u_k u_k^T\right)^2 \quad \text{на} \quad \gamma_k^T \left(\delta_k - H_k \gamma_k\right),$$

получим формулу коррекции (ранга 1):

$$H_{k+1} - H_k = \alpha_k \left(u_k u_k^T\right) = \frac{\left(\delta_k - H_k \gamma_k\right)\left(\delta_k - H_k \gamma_k\right)^T}{\gamma_k^T \left(\delta_k - H_k \gamma_k\right)}.$$

Справедливость этой формулы проистекает из следующей теоремы.

Теорема 6. *Пусть f — квадратичная функция, A — ее гессиан (предполагаемый положительно определенным).*

Пусть итерационный процесс с началом в точке x^0 при перемещении вдоль n последовательно независимых направлений $\delta_1, \delta_2, \ldots, \delta_n$ *порождает последовательно точки* $x^1 = x^0 + \delta_1, \ldots, x^n = x^{n-1} + \delta_n$.

Тогда последовательность матриц H_k:

H_0 — произвольная симметричная матрица,

$$H_{k+1} = H_k + \frac{\left(\delta_k - H_k \gamma_k\right)\left(\delta_k - H_k \gamma_k\right)^T}{\gamma_k^T \left(\delta_k - H_k \gamma_k\right)}, \tag{14}$$

сходится не более чем за n этапов к обращению A^{-1} гессиана функции f, причем

$$\gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k) = A\left(x^{k+1} - x^k\right) = A\delta_k.$$

Доказательство. Если гессиан функции есть постоянная, равная A, то

$$A\delta_i = \gamma_i \quad \forall i.$$

Матрица H_{k+1} по условию строится так, чтобы

$$H_{k+1}\gamma_k = \delta_k.$$

Докажем, что это так и для

$$H_{k+1}\gamma_i = \delta_i, \qquad i = 1, 2, \ldots, k-1.$$

Рассуждаем по индукции, допустив, что это условие выполнено для H_k или, что то же самое,

$$H_k\gamma_i = \delta_i, \quad i = 1, 2, \ldots, k-1.$$

Пусть $i \leqslant k-1$ произвольно. Имеем

$$H_{k+1}\gamma_i = H_k\gamma_i + \frac{(\delta_k - H_k\gamma_k)\left(\delta_k^T\gamma_i - \gamma_k^T H_k\gamma_i\right)}{\gamma_k^T(\delta_k - H_k\gamma_k)}.$$

По предположению индукции $H_k\gamma_i = \delta_i$. Значит,

$$\delta_k^T\gamma_i - \gamma_k^T H_k\gamma_i = \delta_k^T\gamma_i - \gamma_k^T\delta_i.$$

Но $\gamma_k^T = \delta_k^T A$, поэтому

$$\gamma_k^T\delta_i = \delta_k^T A\delta_i = \delta_k^T\gamma_i$$

и, стало быть,

$$H_{k+1}\gamma_i = H_k\gamma_i = \delta_i \qquad \forall i = 1, 2, \ldots, k.$$

В частности, после n шагов имеем

$$H_{n+1}\gamma_i = \delta_i \qquad \forall i = 1, 2, \ldots, n;$$

с учетом равенства $\gamma_i = A\delta_i$ это записывается в виде

$$H_{n+1}A\delta_i = \delta_i \qquad \forall i = 1, 2, \ldots, n.$$

А поскольку δ_i образуют n линейно независимых направлений, то

$$H_{n+1}A = I \quad \text{или} \quad H_{n+1} = A^{-1}.$$

Отсюда получаем требуемое.

Заметим, что корректирующая формула (14) предпочтительна тем, что точка x^{k+1} не должна выбираться как точка минимума функции f в направлении δ_k с началом в точке x^k. Поэтому часто эту формулу рассматривают с точки зрения ее применения для построения алгоритмов, не обязательно требующих одномерной минимизации (*обзор различных методов см., например, в* [66]).

В то же время неудобство формулы (14) состоит в том, что даже если функция f квадратична, а H_0 и гессиан функции f положительно определены, тем не менее матрицы H_k могут не быть положительно определенными. С другой стороны, формула (14) может не иметь смысла, например, если член $\gamma_k^T(\delta_k - H_k\gamma_k)$ обращается в нуль или просто принимает очень малое значение. Если это происходит, необходимо предвидеть специальные процедуры для представления матриц H_k (можно, например, положить $H_{k+1} =$

$= H_k$), но тогда сходимость H_k к обращению гессиана уже не будет обеспечена.

Квазиньютоновские методы, которые будут рассмотрены ниже и которые основаны на формулах коррекции ранга 2, лишены этих недостатков. Напротив, они требуют использования процедуры одномерной оптимизации — точной (алгоритм ДФП) или приближенной (алгоритм БФГШ).

2.10. Алгоритм Давидона — Флетчера — Пауэлла (ДФП). Этот алгоритм использует следующую формулу коррекции (ранга 2):

$$H_{k+1} = H_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k}, \tag{15}$$

где точка x^{k+1} получается из x^k перемещением в направлении $d_k = -H_k \nabla f(x^k)$ и где

$$\delta_k = x^{k+1} - x^k, \quad \gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

Следующий результат показывает, что при некоторых условиях формула (15) сохраняет положительную определенность матриц H_k.

Теорема 7. *Пусть матрицы H_k положительно определены. Тогда при условии $\delta_k^T \gamma_k > 0$ матрица H_{k+1}, полученная по формуле* (15), *положительно определена. Эти условия выполняются, если точка x^{k+1} получается из x^k одномерной минимизацией в направлении $d_k = -H_k \nabla (f(x^k))$.*

Доказательство. Пусть x — произвольный вектор, $x \neq 0$. Требуется показать, что $x^T H_{k+1} x > 0$. Имеем

$$x^T H_{k+1} x = x^T H_k x + \frac{(x^T \delta_k)^2}{\delta_k^T \gamma_k} - \frac{(x^T H_k \gamma_k)^2}{\gamma_k^T H_k \gamma_k};$$

с заменой $u = (H_k)^{1/2} x$, $v = (H_k)^{1/2} \gamma_k$ это записывается в виде

$$x^T H_{k+1} x = \frac{(u^T u)(v^T v) - (u^T v)^2}{(v^T v)} + \frac{(\delta_k^T x)^2}{\delta_k^T \gamma_k}.$$

Первый член справа положителен или равен нулю (согласно неравенству Коши — Буняковского), а второй член положителен или равен нулю при $\delta_k^T \gamma_k > 0$. С другой стороны, оба члена не могут одновременно обращаться в нуль. В самом деле, если первый член равен нулю, то

$$u = \lambda v, \qquad (\lambda \neq 0) \Rightarrow x = \lambda \gamma_k.$$

Но тогда $\delta_k^T x = \lambda \delta_k^T \gamma_k \neq 0$ и второй член отличен от нуля.

Покажем, в частности, что $\delta_k^T \gamma_k > 0$, если точка x^{k+1} получена из x^k одномерной минимизацией в направлении $d_k = -H_k \nabla f(x^k)$. Имеем

$$\delta_k = x^{k+1} - x^k = \theta d_k = -\theta H_k \nabla f(x^k), \quad \theta > 0,$$

$$d_k^T \nabla f(x^{k+1}) = 0 \Rightarrow \delta_k^T \nabla f(x^{k+1}) = 0.$$

При этих условиях

$$\delta_k^T \gamma_k = \delta_k^T (\nabla f(x^{k+1}) - \nabla f(x^k)) = -\delta_k^T \nabla f(x^k) = \theta \nabla f^T(x^k) H_k \nabla f(x^k) > 0.$$

Это свойство сохранения положительной определенности существенно, ибо оно гарантирует, в частности, что направления d_k, последовательно порождаемые алгоритмом, являются направлениями спуска.

Замечание. Условие $\delta_k^T \gamma_k > 0$ выполняется также при использовании методов одномерного поиска, не требующих точного оптимума в данном направлении. Например, если применяется «экономичный» одномерный поиск с правилом Вольфе — Пауэлла (см. гл. 3, п. 2.6), то получаем точку x^{k+1}, удовлетворяющую неравенству

$$d_k^T \nabla f(x^{k+1}) \geqslant m_3 d_k^T \nabla f(x^k), \qquad m_3 < 1$$

откуда

$$-d_k^T \nabla f(x^{k+1}) < -d_k^T \nabla f(x^k),$$

и, следовательно,

$$\delta_k^T \gamma_k > 0.$$

Метод, использующий рекуррентную формулу (15), имеет следующий вид.

Алгоритм Давидона — Флетчера — Пауэлла.

a) Взять начальную точку x^0; выбрать в качестве H_0 произвольную положительно определенную матрицу (например, единичную); $k = 0$.

b) На k-й итерации определить направление

$$d_k = -H_k \nabla f(x^k).$$

Получить x^{k+1} как минимум функции

$$f(x^k + \theta d_k), \qquad \theta \geqslant 0;$$

положить $\delta_k = x^{k+1} - x^k$; вычислить $\gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k)$, затем

$$H_{k+1} = H_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k}.$$

c) $k \leftarrow k + 1$.

Окончить или вернуться в b).

Справедливость этого алгоритма устанавливает следующая

Теорема 8. *Алгоритм Давидона — Флетчера — Пауэлла в применении к квадратичной функции (гессиан A положительно определен) порождает направления $\delta_0, \delta_1, \ldots, \delta_k$, которые при любом k удовлетворяют соотношениям*

$$\delta_i^T A \delta_j = 0, \qquad 0 \leqslant i < j \leqslant k, \tag{16}$$

$$H_{k+1} A \delta_i = \delta_i, \quad 0 \leqslant i \leqslant k. \tag{17}$$

Доказательство. В силу (15) для любого k имеем

$$H_{k+1}A\delta_k = H_{k+1}\gamma_k = H_k\gamma_k + \frac{\delta_k\delta_k^T\gamma_k}{\delta_k^T\gamma_k} - \frac{H_k\gamma_k\gamma_k^TH_k\gamma_k}{\gamma_k^TH_k\gamma_k} = \delta_k.$$

Значит, соотношения (16), (17) верны при $k = 0$.

Допустив, что они верны при $k-1$, докажем их справедливость для k. При $0 \leqslant i < k$ имеем

$$\nabla f(x^k) - \nabla f(x^i) = \gamma_i + \gamma_{i+1} + \ldots + \gamma_{k-1}.$$

Гессиан функции f постоянен и равен A; поэтому $A\delta_i = \gamma_i$ $\forall i$ и, значит,

$$\nabla f(x^k) - \nabla f(x^i) = A(\delta_i + \delta_{i+1} + \ldots + \delta_{k-1}).$$

А поскольку x^i есть оптимум функции f в направлении δ_{i-1}, то

$$\delta_{i-1}^T\nabla f(x^i) = 0.$$

Следовательно, по предположению индукции,

$$\delta_{i-1}^T\nabla f(x^k) = \delta_{i-1}^TA(\delta_i + \delta_{i+1} + \ldots + \delta_{k-1}) = 0, \quad i = 1, 2, \ldots, k-1.$$

В силу (17) и предположения индукции

$$\delta_{i-1} = H_kA\delta_{i-1}.$$

Отсюда

$$\delta_{i-1}^TAH_k\nabla f(x^k) = 0 \quad \forall i, \qquad 1 \leqslant i \leqslant k.$$

А поскольку

$$\delta_k = x^{k+1} - x^k = -\theta H_k\nabla f(x^k), \qquad \theta > 0,$$

то из этого выводим, что

$$\delta_{i-1}A\delta_k = 0 \qquad \forall i = 1, 2, \ldots, k$$

и, стало быть, соотношение (16) выполняется для k.

Докажем теперь, что

$$H_{k+1}A\delta_i = \delta_i \qquad \forall i = 0, 1, \ldots, k-1$$

(как показано выше, для $i = k$ это верно).

Имеем

$$H_{k+1}A\delta_i = H_kA\delta_i + \frac{\delta_k\delta_k^TA\delta_i}{\delta_k^T\gamma_k} - \frac{H_k\gamma_k\gamma_k^TH_kA\delta_i}{\gamma_k^TH_k\gamma_k}.$$

Второй член справа равен нулю, ибо

$$\delta_k^TA\delta_i = 0, \qquad i \leqslant k-1.$$

Третий член — тоже нуль, так как по предположению индукции $H_kA\delta_i = \delta_i$ и $\gamma_k^T = \delta_k^TA$, и, значит,

$$\gamma_k^TH_kA\delta_i = \gamma_k^T\delta_i = \delta_k^TA\delta_i = 0, \qquad i \leqslant k-1.$$

Следовательно,

$$H_{k+1}A\delta_i = H_kA\delta_i = \delta_i, \quad 0 \leqslant i \leqslant k.$$

Теорема показывает, что в квадратичном случае направления $\delta_0, \delta_1, \ldots, \delta_k$, порожденные алгоритмом, последовательно сопряжены относительно матрицы A квадратичной формы. В этом случае алгоритм, стало быть, сходится не более чем за n итераций (при этом заметим, что если за H_0 принимается единичная матрица, то мы вновь приходим к методу сопряженного градиента (см. п. 2.6)).

Наконец, при $k = n - 1$ соотношение (17) дает

$$H_n A \delta_i = \delta_i, \qquad i = 0, 1, \ldots, n-1,$$

а поскольку δ_i линейно независимы (ибо они последовательно сопряжены относительно A), то отсюда получаем

$$H_n A = I; \text{ следовательно, } H_n = A^{-1}.$$

Следовательно, рекуррентная формула (15) позволяет построить в квадратичном случае не более чем за n шагов обращение гессиана функции f.

Как и в методе Флетчера — Ривза (п. 2.7), глобальная сходимость метода гарантируется лишь в случае, если алгоритм периодически обновляется: например, на всех n итерациях в качестве начальной точки выбирается последняя полученная точка и матрица H обновляется (например, берется $H_0 = I$ — единичной матрице).

Первые итерации приводят к перемещению в направлении градиента, поэтому глобальная сходимость алгоритма вытекает из глобальной сходимости метода наискорейшего спуска. Опыт показывает, что когда алгоритм периодически не обновляется, метод может не сходиться к локальному оптимуму.

Асимптотическая сходимость является, вообще говоря, более быстрой, чем в методе Флетчера — Ривза (см. п. 2.12). Напротив, наблюдается значительное превышение объема памяти ($n \times n$-матрица) и количества операций на каждой итерации.

Другая характерная черта алгоритма ДФП состоит в том, что его условия сходимости довольно чувствительны к неточностям в подзадачах одномерной минимизации. Стало быть, для каждой итерации требуется число оценок функции, достаточно высокое для получения требуемой точности, и это может частично снизить преимущества, происходящие из-за квадратичной асимптотической сходимости (см. 2.12).

Алгоритм, который будет описан ниже и который снова связан с семейством квазиньютоновских методов, лишен этого недостатка, при сохранении существенных преимуществ алгоритма ДФП.

2.11. Алгоритм Бройдена — Флетчера — Гольдфарба — Шанно (БФГШ). Этот алгоритм, разработанный независимо в [4, 17, 24, 71], используется для построения аппроксимации обращения гессиана, причем соответствующая формула коррекции ранга 2 непосредственно происходит из формулы (15), которую мы

напоминаем:

$$H_{k+1} = H_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k}.$$

Матрица H_{k+1}, полученная по формуле (15), как известно, удовлетворяет соотношению

$$H_{k+1}\gamma_k = \delta_k. \tag{18}$$

Если (как замечено в [17]) в (15) переставить местами δ_k и γ_k и рассмотреть последовательность матриц, определенных формулой (где матрица G_0 — произвольная симметричная положительно определенная)

$$G_{k+1} = G_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{G_k \delta_k \delta_k^T G_k}{\delta_k^T G_k \delta_k}, \tag{19}$$

то полученные матрицы будут удовлетворять соотношению, «обратному» к (18):

$$G_{k+1}\delta_k = \gamma_k. \tag{20}$$

Теперь мы видим, что формула (19) позволяет построить аппроксимацию самого гессиана (а не его обращения).

Следовательно, если мы хотим, исходя из формулы (19), получить формулу коррекции для аппроксимации обращения гессиана, то достаточно взять обращение для обеих частей формулы (19). После проведения выкладок получаем

$$[G_{k+1}]^{-1} = [G_k]^{-1} + \left[1 + \frac{\gamma_k^T [G_k]^{-1} \gamma_k}{\delta_k^T \gamma_k}\right] \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\delta_k \gamma_k^T [G_k]^{-1} \gamma_k \delta_k^T}{\delta_k^T \gamma_k}. \tag{21}$$

Ясно, что (21) позволяет выразить непосредственно $[G_{k+1}]^{-1}$ как функцию от $[G_k]^{-1}$, и мы приходим к рекуррентной формуле

$$H_{k+1} = H_k + \left[1 + \frac{\gamma_k^T H_k \gamma_k}{\delta_k^T \gamma_k}\right] \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\delta_k \gamma_k^T H_k + H_k \gamma_k \delta_k^T}{\delta_k^T \gamma_k}. \tag{22}$$

После этого алгоритм БФГШ получается непосредственно из алгоритма ДФП заменой формулы (15) на формулу (22).

Формула (22) обладает свойствами, аналогичными свойствам формулы (15). В частности:

— если $\delta_k^T \gamma_k > 0$, то положительная определенность матриц H_k сохраняется;

— эта формула, примененная к квадратичной функции (гессиан A положительно определен), позволяет не более чем за n итераций получить обращение A^{-1} гессиана функции f. Кроме того, направления δ_i, последовательно порождаемые алгоритмом БФГШ, последовательно сопряжены относительно A^{-1}.

Когда алгоритм применяется к произвольной (не квадратичной) нелинейной функции, то необходимо, как и для алгоритма

ДФП, обратиться к периодическому обновлению, чтобы обеспечить глобальную сходимость.

Преимущество алгоритма БФГШ перед алгоритмом ДФП общепризнано. Существенной причиной для этого служит то, что БФГШ гораздо менее чувствителен к неточностям в процедуре одномерного поиска.

Это позволяет использовать «экономичные» одномерные методы оптимизации (типа Голдстейна, Вольфе — Науэлла; см. гл. 3, п. 2.6), которые требуют очень мало оценок функции на каждой итерации, но их применение не ухудшает скорость сходимости алгоритма (сходимость высшего порядка; см. [14]).

2.12. Сравнение различных методов. Результаты, относящиеся к скорости сходимости. В дальнейшем будем всегда предполагать выполненным условие (*II*):

$$\begin{aligned} &- f(x) \text{ дважды непрерывно дифференцируема;} \\ &- \text{гессиан } \nabla^2 f(x^*) \text{ положительно определен.} \end{aligned} \qquad (II)$$

За доказательствами (очень техническими) представленных ниже утверждений отсылаем к библиографии.

1. *Метод наискорейшего спуска.* Можно доказать (см., например, [50]), что последовательность $f(x^k)$ удовлетворяет неравенству

$$\limsup_{(k\to\infty)} \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)} = \alpha \leqslant \left(\frac{A-a}{A+a}\right)^2,$$

где A, a — *соответственно наибольшее и наименьшее собственные значения* гессиана $\nabla^2 f(x^*)$ в точке x^*.

Стало быть, в худшем случае имеет место линейная сходимость, для которой коэффициент асимптотической сходимости $[(A-a)/(A+a)]^2$ (называемый *отношением Канторовича*) непосредственно связан с *обусловленностью* A/a матрицы $\nabla^2 f(x^*)$.

Этот результат с очевидностью показывает, что сходимость может быть очень медленной для плохо обусловленных функций (овражного типа).

2. *Ускоренные методы наискорейшего спуска.* Для ускоренного метода второго порядка доказано (например, [50]), что

$$\limsup_{k\to\infty} \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)} = \alpha \leqslant \left(\frac{B-a}{B+a}\right)^2,$$

где B — второе наибольшее собственное значение гессиана $\nabla^2 f(x^*)$ (a — все еще наименьшее собственное значение).

В двумерном случае имеем $B = a$, и, следовательно, ускоренный метод второго порядка имеет сходимость высшего порядка. Значит, он очень эффективен (но при этом надо заметить, что для того, чтобы перейти от x^k к x^{k+1}, понадобятся три одномерные минимизации). Однако в размерностях выше второй сходимость с $p = 2$ может практически быть такой же плохой, как в обычном методе, ибо второе наибольшее собственное значение для $\nabla^2 f(x^*)$ может мало отличаться от A.

Таким образом, может оказаться необходимым перейти к процессам более высокого порядка p, за счет существенного возрастания времени вычисления. Предыдущие результаты без труда распространяются на случай $p > 2$, при этом нужно учитывать p-е наибольшее собственное значение для $\nabla^2 f(x^*)$.

В частности, для $p = n$ (размерность пространства) находим наличие суперлинейной сходимости (но на каждой итерации требуется $n+1$ одномерных минимизаций). Тогда метод становится эквивалентным методу сопряженного градиента [19].

3. *Метод Ньютона.* Легко доказать квадратичную сходимость, если предположить, что $\nabla^2 f$ удовлетворяет в окрестности точки x^* условию Липшица типа

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leqslant c\|x - y\|.$$

В самом деле, согласно формуле Тейлора имеем

$$\nabla f(x^*) - \nabla f(x^k) = \nabla^2 f(\xi)(x^* - x^k),$$

где ξ принадлежит сегменту $[x^*, x^k]$. А поскольку

$$\nabla f(x^*) - \nabla f(x^k) - \nabla^2 f(x^k)(x^* - x^k) = [\nabla^2 f(\xi) - \nabla^2 f(x^k)](x^* - x^k),$$

то для x^k, достаточно близких к x^*, имеем

$$\|\nabla f(x^*) - \nabla f(x^k) - \nabla^2 f(x^k)(x^* - x^k)\| \leqslant c\|x^k - x^*\|^2,$$

откуда следует

$$\|\nabla^2 f(x^k)(x^{k+1} - x^*)\| \leqslant c\|x^k - x^*\|^2.$$

Таким образом, можем написать

$$\|\nabla^2 f(x^*)(x^{k+1} - x^*)\| \leqslant \|[\nabla^2 f(x^k)](x^{k+1} - x^*)\| +$$
$$+ \|\nabla^2 f(x^k)(x^{k+1} - x^*)\| \leqslant c\|x^k - x^*\|\,\|x^{k+1} - x^*\| + c\|x^k - x^*\|^2$$

и, обозначив через $\lambda > 0$ наибольшее собственное значение для $\nabla^2 f(x^*)$, получаем

$$\lambda\|x^{k+1} - x^*\| \leqslant \|\nabla^2 f(x^*)(x^{k+1} - x^*)\| \leqslant$$
$$\leqslant c\|x^k - x^*\|\,\|x^{k+1} - x^*\| + c\|x^k - x^*\|^2,$$

откуда вытекает квадратичная сходимость. Заметим снова, что здесь речь идет о локальной сходимости, но на практике глобальная сходимость метода не гарантируется.

4. *Методы сопряженных направлений.* Излагаемые ниже результаты применимы, в частности, к методу Флетчера — Ривза.

Все еще при условии (H) доказано [51], что

$$\frac{\|x^{k+n} - x^*\|}{\|x^k - x^*\|} \to 0, \qquad k \to \infty,$$

или, что то же самое, что последовательность $x^0, x^n, x^{2n}, \ldots, x^{qn}$ сходится к x^* суперлинейно.

Итак, речь идет о суперлинейной сходимости по n-шагам (причем каждый шаг требует одной одномерной оптимизации).

Если при этом кроме условия (H) гессиан $\nabla^2 f(x)$ удовлетворяет условию Липшица

$$\|\nabla^2 f(x)y\| \leqslant K\|y\| \quad \forall y$$

в окрестности точки x^*, то доказана квадратичная сходимость по n-шагам:

$$\limsup_{k\to\infty} \frac{\|x^{k+n} - x^*\|}{\|x^k - x^*\|^2} < +\infty.$$

Интересно отметить, что предыдущие результаты применимы лишь в случае, когда предполагается, что точный оптимум получен в каждой одномерной оптимизации. Они остаются справедливыми, даже если последовательные одномерные оптимизации перестают быть точными, лишь бы выполнялось

$$\frac{\|\nabla f^T(x^{k+1})(x^{k+1} - x^k)\|}{\|\nabla f(x^{k+1})\|\,\|x^{k+1} - x^k\|} \to 0, \qquad k \to \infty.$$

5. *Квазиньютоновские методы.* Излагаемые результаты применимы, в частности, к алгоритму (ДФП) (но применяются и к большинству других квазиньютоновских методов).

При условии (H) доказана суперлинейная сходимость к x^*,

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \to 0, \qquad k \to \infty.$$

Если, кроме того, $\nabla^2 f(x)$ удовлетворяет условию Липшица

$$\|\nabla^2 f(x)y\| \leqslant K\|y\| \quad \forall y$$

в окрестности точки x^*, то доказана квадратичная сходимость к x^*:

$$\limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < +\infty \quad [51].$$

Доказательство [13] немного более общее, чем доказательство [51]; при этом весьма интересно отметить, что оно предполагает лишь сходимость последовательности матриц $\{H_k\}$ (см. п. 2.10) к $[\nabla^2 f(x)]^{-1}$ (так называемое условие «состоятельности»).

Предыдущие результаты выявляют преимущество квазиньютоновских методов перед методами сопряженного градиента — последние требуют приблизительно в n раз больше этапов (шагов) (в n раз больше одномерных минимизаций) для одного и того же асимптотического поведения (квадратичная или суперлинейная сходимость).

Следует отметить, однако, что это преимущество в пользу квазиньютоновских методов сильно снижается:

— загрузкой памяти пропорционально n^2;

— объемом промежуточных вычислений (матричное исчисление), пропорциональным n^2.

Наконец, метод сопряженных направлений всегда может быть улучшен при помощи процесса ускорения и преобразован в квазиньютоновский метод [2].

§ 3. Оптимизация выпуклых функций, не являющихся всюду дифференцируемыми

До сих пор во всех изучавшихся методах предполагалось, что минимизируемая функция f непрерывно дифференцируема — иными словами, что в каждой точке существует градиент функции f.

Теперь мы откажемся от этого условия *дифференцируемости*. Однако мы не будем всюду в дальнейшем считать, что имеет место самый общий случай; например, в этом параграфе мы допускаем условие выпуклости изучаемой функции (в случае минимизации) или вогнутости (в случае максимизации).

Вопреки ожиданию этот тип задач часто встречается в математическом программировании. Приведем один пример.

3.1. Пример. Двойственность в дискретном программировании. Допустим, например, что мы хотим решить задачу дискретного программирования

$$\begin{aligned} &\varphi(x) \to \min, \\ &\psi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \ x \in S, \end{aligned} \tag{P}$$

где $\varphi(x)$, $\psi_i(x)$ — произвольные функции, S — конечное дискретное множество:

$$S = \{y^1, y^2, \ldots, y^k\}.$$

Некоторые из наиболее эффективных методов решения такого рода задач (гл. 7, § 2) используют двойственную к (P) задачу, которая может быть представлена в виде (гл. 6, § 2)

$$\begin{aligned} &w(\pi) \to \max, \\ &\pi \in \mathbf{R}^{m^+}, \end{aligned} \tag{D}$$

где (двойственная) функция $w(\pi)$ для

$$\pi = (\pi_1, \pi_2, \ldots, \pi_m) \geqslant 0$$

имеет вид

$$w(\pi) = \min_{x \in S} \{\varphi(x) + \pi\psi(x)\},$$

причем

$$\psi(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_m(x)]^T.$$

Легко показать, что функция $w(\pi)$ вогнута, не является всюду дифференцируемой функцией от π и является нижней оболочкой в пространстве $\mathbf{R}^{m+1}$ координат (z, π) семейства K гиперплоскостей, заданных уравнениями

$$\begin{aligned} z &= \varphi(y^1) + \pi\psi(y^1), \\ z &= \varphi(y^2) + \pi\psi(y^2), \\ &\cdots\cdots\cdots \\ z &= \varphi(y^K) + \pi\psi(y^K). \end{aligned}$$

Таким образом, решение двойственной задачи сводится к отысканию максимума вогнутой функции $w(\pi)$, не являющейся всюду дифференцируемой, или, что эквивалентно, к отысканию минимума выпуклой функции $-w(\pi)$.

Доказано также, что (гл. 6, § 2) величина $w(\pi)$ будет при любом π минорантой всякого решения задачи (P), и, следовательно, оптимальное решение π^* задачи (D) есть наибольшая миноранта такого типа. Значения $w(\pi^*)$ могут быть использованы в перечисленных методах типа метода ветвей и границ (гл. 7, § 2), чтобы избежать перебора большого числа решений и тем самым ускорить сходимость.

3.2. Нахождение субградиента. Пример. В гл. 1 мы видели, каким образом для выпуклых функций понятие субградиента составляет обобщение понятия градиента. С точки зрения алгоритмической это понятие, очевидно, представляет интерес только в том случае, если мы умеем легко находить хотя бы один субградиент в каждой точке.

Покажем, как это может быть осуществлено в примере п. 3.1. Функция $w(\pi)$ имеет вид

$$w(\pi) = \min_{x \in S} \{\varphi(x) + \pi\psi(x)\} \qquad \forall \pi \in \mathbb{R}^{m^+}, \tag{23}$$

где $S = \{y^1, y^2, \ldots, y^\kappa\}$ — конечное дискретное множество.

Пусть $y^i \in S$ — элемент, для которого минимум достигается. Тогда доказывается (гл. 6, п. 2.7), что вектор

$$-\psi(y^i) = -[\psi_1(y^i), \psi_2(y^i), \ldots, \psi_m(y^i)]^T$$

есть субградиент (выпуклой) функции $-w(\pi)$ в точке π.

Итак, на этом примере показано, что вычисление субградиента вытекает непосредственно из вычисления самой функции, и замечателен тот факт, что это простота используется во многих примерах, представляющих практический интерес.

Однако речь идет лишь о получении какого-то субградиента функции в одной точке. Полное нахождение субдифференциала $\partial w(\pi)$ представляет собой операцию, гораздо более сложную, гораздо более дорогостоящую по времени вычислений.

3.3. Использование обобщенного линейного программирования. Первый, очень распространенный, метод минимизации выпуклой функции f, для которой можно вычислить субградиент в любой точке, состоит в аппроксимации функции f при помощи тангенциальной линеаризации, а затем итеративного измельчения полученного приближения.

Тем самым приходим к решению линейной задачи, ограничения которой добавляются по мере итерирования.

Этот метод подробно изложен в гл. 8, п. 1.6.

Однако для задач, содержащих большое число переменных, может оказаться предпочтительнее использовать описываемые ниже методы.

3.4. Один класс субградиентных алгоритмов. Сначала обратимся к случаю без ограничений и предположим, что (точно так же, как в примере из п. 3.2) минимизируемая функция f такова, что возможно практическое вычисление субградиента γ в каждой точке x.

Тогда градиентные алгоритмы (градиент с фиксированным шагом) обобщается непосредственно на недифференцируемый случай: для этого достаточно в качестве направления перемещения взять на каждом k-м этапе (шаге) субградиент в x^k (вместо градиента).

Субградиентный алгоритм.

a. На 0-м шаге выбирается начальная точка x^0.

b. На k-м шаге выбирается начальная точка x^k. Находим субградиент $\gamma^k \in \partial f(x^k)$; если $\gamma^k = 0$, то конец: x^k есть оптимум; иначе взять x^{k+1}:

$$x^{k+1} = x^k - \lambda_k \frac{\gamma^k}{\|\gamma^k\|}.$$

c. Проверка на окончание; если верно, то конец; иначе выполнить $k \leftarrow k+1$ и вернуться в b).

Этот тип алгоритма без труда обобщается на задачи оптимизации с ограничениями — задачи вида

$$f(x) \to \min,$$
$$x \in X \subset \mathbf{R}^n,$$

где f — выпуклая функция, не являющаяся всюду дифференцируемой, и где $X \subset \mathbf{R}^n$ замкнуто и выпукло.

Тогда предположим, что выполняется хотя бы одно из следующих условий:

1) $f(x) \to +\infty$ при $\|x\| \to +\infty$;

2) X ограничено.

Это обеспечивает существование оптимального решения $x^* \in X$ (см. теорему Вейерштрасса, гл. 1, п. 2.3).

Обозначив через $\mathrm{Pr}|_X$ оператор проектирования на выпуклое множество ($\mathrm{Pr}|_X = x \Leftrightarrow x \in X$), получаем алгоритм для случая с ограничением:

Субградиентный алгоритм с ограничениями.

a. На 0-м шаге выбираем начальную точку x^0.

b. На k-м шаге мы находимся в точке $x^k \in X$; если $\gamma^k = 0$, то конец: x^k оптимально. Иначе определить x^{k+1}:

$$x^{k+1} = \mathrm{Pr}|_X \left[x^k - \lambda_k \frac{\gamma^k}{\|\gamma^k\|} \right].$$

c. Тест на остановку. Если удовлетворяется, то конец; иначе взять $k \leftarrow k+1$ и вернуться к b).

Заметим, что в (часто встречающемся) случае ограничений типа $x \geqslant 0$ (см. п. 3.1) операция проектирования на X (положительный ортант) реализуется особенно просто, поскольку для любого $x \in \mathbf{R}^n$ точка $x' = \mathrm{Pr}|_X(x)$ определяется равенствами

$$x'_i = x_i, \qquad x_i \geqslant 0,$$
$$x'_i = 0, \qquad x_i < 0.$$

Сходимость сформулированных выше алгоритмов существенным образом зависит от способа выбора на каждой k-й итерации шагов перемещения λ_k.

Следующий результат, доказываемый здесь для самого общего случая задачи с ограничениями, показывает, что из последовательности $\{x^k\}$ можно выбрать подпоследовательность, сходящуюся к оптимальному решению задачи, если выполнены только следующие два условия:

$$\lambda_k \to 0, \quad k \to \infty,$$

$$\sum_{k=1}^{\infty} \lambda_k = +\infty$$

(так называемое *правило расходящегося ряда*).

Теорема 9 [64, 16]. *Пусть f — непрерывная выпуклая функция на R^n, X — замкнутое выпуклое множество (предполагаем, что либо $f(x) \to +\infty$ при $\|x\| \to +\infty$, либо множество X ограничено).*

Тогда если шаги перемещения λ_k выбраны в соответствии с правилом расходящегося ряда, то последовательность, порожденная формулой

$$x^{k+1} = \Pr|_X \left[x^k - \lambda_k \frac{\gamma^k}{\|\gamma^k\|} \right]$$

удовлетворяет соотношениям

$$\liminf \{f(x^k)\} = f^* = \min_{x \in X} \{f(x)\}.$$

Доказательство. Достаточно показать, что для любого $\varepsilon > 0$ найдется такое i, что

$$x^i \in V(\varepsilon) = \{x | f(x) \leqslant f^* + \varepsilon;\ x \in X\}.$$

Рассуждаем от противного, предположив, что $x^k \in V(0)$ $\forall k$, или, что то же самое,

$$\forall k:\ f(x^k) > f^* + \varepsilon.$$

В соответствии с условиями существует такая точка $x^* \in V(0)$, что $f(x^*) = f^*$. А поскольку f непрерывно на R^n, то найдется такое $\delta > 0$, что

$$\|x - x^*\| \leqslant \delta \Rightarrow f(x) \leqslant f^* + \varepsilon.$$

Всегда можно предположить, что $\gamma^k \neq 0$ $\forall k$ (в самом деле, из равенства $\gamma^k = 0$ следует, что x^k оптимальна, и алгоритм заканчивается); рассмотрим точку $\bar{x}^k$, определяемую формулой

$$\bar{x}^k = x^* + \delta \frac{\gamma^k}{\|\gamma^k\|}.$$

Имеем $\|\bar{x}^k - x^*\| = \delta$, значит, $f(\bar{x}^k) \leqslant f^* + \varepsilon$, и, стало быть,

$$(\gamma^k)^T (\bar{x}^k - x^k) \leqslant f(\bar{x}^k) - f(x^k) < 0 \quad \forall k.$$

Кроме того, оператор проектирования является оператором сжатия (гл. 10, п. 3.8), поэтому можно написать

$$\| x^{k+1} - x^* \| = \left\| \Pr|_X \left[x^k - \lambda_k \frac{\gamma^k}{\| \gamma^k \|} \right] - \Pr|_X [x^*] \right\| \leqslant$$

$$\leqslant \left\| x^k - \lambda_k \frac{\gamma^k}{\| \gamma^k \|} - x^* \right\|.$$

Отсюда получаем

$$\| x^{k+1} - x^* \|^2 \leqslant \| x^k - x^* \|^2 + \lambda_k^2 - 2\lambda_k \frac{(\gamma^k)^T (x^k - x^*)}{\| \gamma^k \|}.$$

Заметив, что

$$(\gamma^k)^T (x^k - x^*) = (\gamma^k)^T \left[x^k - \bar{x}^k + \frac{\delta \gamma^k}{\| \gamma^k \|} \right] = (\gamma^k)^T (x^k - \bar{x}^k) + \delta \| y^k \|,$$

приходим к неравенству

$$\| x^{k+1} - x^* \|^2 \leqslant \| x^k - x^* \|^2 + \lambda_k^2 + 2\lambda_k \frac{(y^k)^T (\bar{x}^k - x^k)}{\| \gamma^k \|} - 2\delta\lambda_k \leqslant$$

$$\leqslant \| x^k - x^* \|^2 + \lambda_k (\lambda_k - 2\delta).$$

А поскольку $\lambda_k \to 0$, то найдется такое K, что

$$k \geqslant K \Rightarrow \lambda_k \leqslant \delta.$$

Тогда

$$\| x^{k+1} - x^* \|^2 \leqslant \| x^k - x^* \|^2 - \delta\lambda_k \quad \forall k \geqslant K,$$

отсюда, суммируя по k, получаем

$$\delta \sum_{k=K}^{K+p} \lambda_k \leqslant \| x^K - x^* \|^2 - \| x^{K+p+1} - x^* \|^2 \leqslant \| x^K - x^* \|^2.$$

После этого, выбирая достаточно большое p, приходим к противоречию с тем, что $\sum_{k=1}^{\infty} \lambda_k = +\infty$, и теорема доказана.

Очевидно, что субградиентный алгоритм, основанный на правиле расходящегося ряда, не представляет практического интереса, ибо сходимость к оптимальной точке, вообще говоря, является исключительно медленной.

Поэтому мы будем изучать другие процессы выбора параметров перемещения λ_k, позволяющие при некоторых условиях получать более высокие скорости сходимости,— более конкретно, три следующих метода.

Метод 1 (метод с постоянным шагом):

$$\lambda_k = \lambda = \text{const}.$$

Метод 2 (метод сходящегося ряда):

$$\lambda_k = \lambda_0 (\alpha)^k, \qquad 0 < \alpha < 1 \qquad [23, 72].$$

Метод 3 (метод релаксации):

$$\lambda_k = \rho \frac{f(x^k) - \bar{f}}{\| \gamma^k \|} \quad [29],$$

где $\bar{f}$ — оценка оптимального значения f^*, коэффициент ρ удовлетворяет условию $0 < \rho \leqslant 2$ (коэффициент релаксации).

Для исследования сходимости субградиентных алгоритмов мы вначале сформулируем понятие обусловленности выпуклой функции f — понятие, которое будет очень часто использоваться в дальнейшем.

Оптимум x^* функции f может быть не единственным; обозначим через $\Omega \subset \mathbf{R}^n$ множество точек x, доставляющих функции f оптимальное значение f^*:

$$\Omega = \{x | x \in \mathbf{R}^n;\ f(x) = f^*\}$$

(отметим, что для выпуклой функции f множество Ω выпукло).

Для любого $x \in \mathbf{R}^n$ $(x \notin \Omega)$ обозначим через $s(x)$ проекцию точки x на Ω, а через $d(x) = \| s(x) - x \|$ — расстояние точки x до множества Ω оптимальных точек.

По определению, *обусловленность* χ функции f есть косинус наибольшего угла θ между каким-либо субградиентом $-\gamma$ в произвольной точке x $(x \notin \Omega)$ и направлением $s(x) - x$, ведущем в ближайшую точку множества Ω оптимальных точек (итак, χ есть число, заключенное между 0 и 1). Более точно,

$$\chi = \cos\theta = \inf_{\substack{x \in \mathbf{R}^n \\ x \notin \Omega}} \inf_{y \in \partial f(x)} \left\{ \frac{-\gamma^T (s(x) - x)}{\| \gamma \| \| s(x) - x \|} \right\}.$$

Замечание. Для квадратичной функции

$$f(x) = \frac{1}{2} x^T A x + bx + c,$$

где A — положительно определенная матрица с экстремальными собственными значениями $\alpha_{\min}$ и $\alpha_{\max}$, можно показать, что

$$\chi = \frac{2\sqrt{\alpha_{\min}\alpha_{\max}}}{\alpha_{\min} + \alpha_{\max}}.$$

Чтобы упростить изложение, мы в дальнейшем рассмотрим только случай без ограничений, хотя основные формулируемые результаты могут быть обобщены на случай с ограничениями вида $x \in X$, где X — замкнутое выпуклое множество.

3.5. Сходимость метода 1 (метод с постоянным шагом). Имеем

$$x^1 = x^0 - \lambda \frac{\gamma^0}{\| \gamma^0 \|}.$$

Если выбранное λ меньше $d(x^0)\cos\theta = \chi d(x^0)$, то в наиболее неблагоприятном случае (т. е. когда $\cos\theta = \chi$) имеем (см. рис. 7)

$$d(x^1) \leqslant \| x^1 - s(x^0) \| \leqslant \| y - s(x^0) \|,$$

где по построению $[x^0, s(x^0)]$, причем

$$\|y - y^1\| = \|x^1 - y^1\| = \frac{\lambda}{\chi}\sqrt{1-\chi^2}.$$

Отсюда

$$d(x^1) \leqslant \|x^1 - s(x^0)\| \leqslant \|y - s(x^0)\|$$

и, значит,

$$d(x^1) \leqslant d(x^0) - \frac{\lambda}{\chi}(1 - \sqrt{1-\chi^2}).$$

Стало быть, точка x^1 ближе точки x^0 к множеству Ω оптимальных решений, и убывание Δd расстояния по крайней мере равно

$$\Delta d = \frac{\lambda}{\chi}(1 - \sqrt{1-\chi^2}),$$

где χ — обусловленность функции f.

Если снова $d(x^1) \geqslant \lambda/\chi$, то, повторяя предыдущую выкладку, получаем точку x^2:

$$x^2 = x^1 - \lambda \frac{\gamma^1}{\|\gamma^1\|}, \qquad \gamma^1 \in \partial f(x^1),$$

причем

$$d(x^2) \leqslant d(x^1) - \frac{\lambda}{\chi}(1 - \sqrt{1-\chi^2}).$$

Тем самым показано, что алгоритм позволяет получить за конечное число итераций (меньшее или равное $(d(x^0) - \lambda/\chi)/\Delta d$) точку,

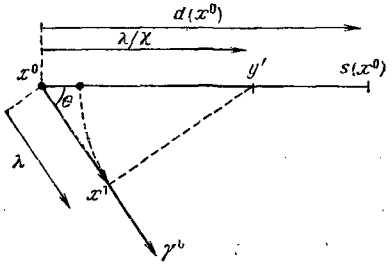


Рис. 7. Изучение сходимости метода субградиента с постоянным шагом (случай, когда $d(x^0) \geqslant \lambda/\chi$)

расстояние которой до множества оптимальных решений меньше или равно λ/χ.

Иными словами, если для любого $\varepsilon > 0$ выбираем $\lambda = \chi\varepsilon$, то за конечное число итераций получаем решение, отстоящее не более чем на расстояние ε от множества оптимальных решений.

Очевидно, что это количество операций может быть очень велико. Например, для $\chi = 1/2$ (достаточно хорошо обусловленная функция) имеем $\sqrt{1-\chi^2} = 0{,}87$. Если мы хотим разделить расстояние $d(x^0)$ от начальной точки на 1000, то берем $\lambda = d(x^0)/2000$, и число итераций будет иметь порядок

$$\frac{d(x^0)}{\Delta d} = \frac{1000}{0{,}13} = 7500.$$

Предыдущий анализ показывает, что если λ выбрано слишком малым, то сходимость может оказаться очень медленной. Стало

быть, представляет интерес начальный выбор достаточно большого λ,— конечно, при соблюдении условия $\lambda \leqslant \chi d(x^0)$.

При известных χ и $d(x^0)$ хороший выбор может состоять в том, чтобы брать $\lambda = (1/2)\chi d(x^0)$.

В этом случае число n итераций, необходимых для получения точки x на расстоянии $d(x) \leqslant \lambda/\chi = (1/2)d(x^0)$, удовлетворяет условию

$$n \leqslant \frac{d(x^0) - (1/2)\,d(x^0)}{(1/2)\,d(x^0)\left(1 - \sqrt{1-\chi^2}\right)} = \frac{1}{1-\sqrt{1-\chi^2}}.$$

Если функция не является плохо обусловленной, то это число итераций может быть очень малым. Например, для $\chi = 1/2$

$$n \leqslant \frac{1}{0{,}13} = 7{,}5.$$

Для этого случая можно предложить следующий алгоритм.

a) Выбрать начальную точку x^0; оценить χ и $d(x^0)$; вычислить

$$n = \frac{1}{1-\sqrt{1-\chi^2}},$$

положить $\lambda = (1/2)d(x^0)$.

b) Отправляясь из точки x^0, осуществить n итераций вида

$$x^{k+1} = x^k - \lambda \frac{\gamma^k}{\|\gamma^k\|};$$

пусть x^n — полученная точка.

c) Взять $\lambda \leftarrow \lambda/2$,

$$x^0 \leftarrow x^n$$

и вернуться в b).

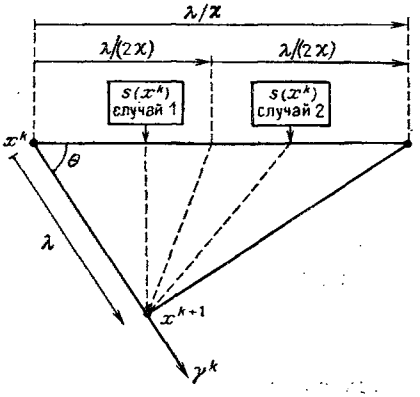


Рис. 8. Изучение сходимости метода субградиента с постоянным шагом (случай, когда $d(x^k) \leqslant \lambda/\chi$)

Задача состоит в оценке используемых величин χ и $d(x^0)$. Вообще, когда исследуется рассматриваемый сейчас класс задач, разумно предположить, что можно получить достаточно точную оценку значения обусловленности χ (полученного в результате эксперимента).

Тогда существенная проблема состоит в получении хорошего приближения для $d(x^0)$. Для этого исследуем поведение последовательности $\{x^k\}$, начиная с момента, когда была получена точка x^k, удовлетворяющая условию $d(x^k) \leqslant \lambda/\chi$.

Рассмотрим два случая (рис. 8).

Случай 1. Если

$$\frac{\lambda}{2\chi} \leqslant d(x^k) \leqslant \frac{\lambda}{\chi},$$

то

$$d(x^{k+1}) \leqslant \|x^{k+1} - s(x^k)\| < d(x^k) \leqslant \lambda/\chi$$

и расстояние продолжает строго убывать (но убывание может быть очень медленным).

Случай 2. Если $d(x^k) \leqslant \lambda/(2\chi)$, то

$$d(x^{k+1}) \leqslant \| x^{k+1} - s(x^k) \| \leqslant \max\left\{\lambda; \frac{\lambda}{2\chi}\right\}.$$

А поскольку $\chi \leqslant 1$, то $\lambda \leqslant \lambda/\chi$; отсюда выводим, что все полученные в дальнейшем точки $x^{k+1}, x^{k+2}, \ldots$ останутся от множества оптимальных точек на расстоянии, меньшем λ/χ.

Изложенное выше позволяет получить эффективный метод проверки того факта, что получена точка x^k, удовлетворяющая $d(x^k) \leqslant \lambda/\chi$, в предположении, что множество Ω оптимальных точек сводится к множеству из одной точки x^* (это условие часто выполняется в практических задачах). Очевидно, можно предположить, что χ известно (или что мы располагаем хорошей оценкой для χ).

Если точка x^k не содержится в сфере Σ с центром x^* и радиусом λ/χ, то для любой точки x^i $(0 \leqslant i < k)$ имеем

$$\|x^i - x^*\| - \|x^{i+1} - x^*\| \geqslant \Delta d,$$
$$\|x^{i+1} - x^*\| - \|x^{i+2} - x^*\| \geqslant \Delta d,$$
$$\cdots\cdots\cdots\cdots$$
$$\|x^{k-1} - x^*\| - \|x^k - x^*\| \geqslant \Delta d.$$

Отсюда следует, что

$$\|x^i - x^*\| - \|x^k - x^*\| \geqslant (k-i)\Delta d, \qquad \Delta d = (\lambda/\chi)\left(1 - \sqrt{1-\chi^2}\right),$$

и, значит, должно выполняться неравенство

$$\| x^i - x^k \| \geqslant (k-i)\frac{\lambda}{\chi}\left(1 - \sqrt{1-\chi^2}\right).$$

Следовательно, если для некоторой точки x^k из последовательности существует такая точка x^i $(0 \leqslant i < k)$, что

$$\| x^i - x^k \| \leqslant (k-i)\frac{\lambda}{\chi}\left(1 - \sqrt{1-\chi^2}\right),$$

отсюда непосредственно вытекает, что точка x^k лежит внутри сферы Σ и что бесполезно пытаться проводить итерации с этим значением λ. (И напротив, можно проводить итерации с меньшим значением λ.)

Поскольку максимальное расстояние между двумя произвольными точками сферы Σ равно $2\lambda/\chi$, можно, кроме того, легко получить верхнюю оценку для числа итераций, необходимых для того, чтобы убедиться, что последовательность $\{x^k\}$ содержится в Σ.

Это число n определяется соотношениями

$$n\Delta d = \frac{2\lambda}{\chi}, \qquad n = \frac{2}{1 - \sqrt{1 - x^2}}.$$

Например, для $\chi = 1/2$ (достаточно хорошо обусловленные функции) получаем $\sqrt{1-\chi^2} = 0{,}87$ и $n = 2/0{,}13 = 15$ итераций.

Заметим, что здесь речь идет о верхней оценке; практически число итераций может быть гораздо меньше.

3.6. Сходимость метода 2 (метод сходящегося ряда). Выберем теперь на k-й итерации шаг λ_k по формуле

$$\lambda_k = \lambda_0 (\alpha)^k,$$

где $\alpha < 1$.

Чтобы убедиться в приемлемости этой формулы, предположим, что x^{k+1} определяется из x^k формулой

$$x^{k+1} = x^k - t\gamma^k.$$

Можно записать

$$d(x^{k+1}) = \|x^{k+1} - s(x^{k+1})\| \leqslant \|x^{k+1} - s(x^k)\|$$

и

$$\|x^{k+1} - s(x^k)\|^2 = \|x^k - s(x^k) - t\gamma^k\|^2 =$$
$$= \|x^k - s(x^k)\|^2 - 2t(\gamma^k)^T [x^k - s(x^k)] + t^2 \|\gamma^k\|^2.$$

Стало быть,

$$d^2(x^{k+1}) \leqslant d^2(x^k) + t[t\|\gamma^k\|^2 - 2(\gamma^k)^T [x^k - s(x^k)]]. \qquad (24)$$

Если χ — обусловленность функции f (см. п. 3.4), то

$$\gamma^T [x - s(x)] \geqslant \chi d(x) \|\gamma\| \qquad \forall x \in \Omega, \quad \forall \gamma \in \partial f(x).$$

Тогда из (24) получаем

$$d^2(x^{k+1}) \leqslant d^2(x^k) + t[t\|\gamma^k\|^2 - 2\chi d(x^k) \|\gamma^k\|].$$

Значение t, минимизирующее правую часть, равно

$$t = \chi \frac{d(x^k)}{\|\gamma^k\|},$$

и, значит, для этого значения

$$d^2(x^{k+1}) \leqslant d^2(x^k)[1 - \chi^2].$$

Следовательно, если χ известно и если на каждой итерации известно $d(x^k)$, можно получить линейную сходимость к оптимальной точке с множителем сходимости $\sqrt{1-\chi^2}$.

В этом случае сама последовательность коэффициентов t ведет себя как геометрическая прогрессия с множителем $\sqrt{1-\chi^2}$, что сводится к тому, что мы полагаем

$$\lambda_k = \lambda_0 (\alpha)^k$$

с

$$\lambda_0 = \chi d(x^0), \qquad \alpha = \sqrt{1-\chi^2}.$$

Значит, $\sqrt{1-\chi^2}$ есть предельный множитель сходимости (самой быстрой), получаемой в идеальном случае, где наилучший выбор t осуществляется на каждой итерации.

Практически, однако, мы не знаем ни χ, ни $d(x^0)$ и должны довольствоваться, вообще говоря, множителем сходимости $\alpha > \sqrt{1-\chi^2}$.

Более точно, рассмотрим следующие величины (которые зависят лишь от χ и от α):

$$C = \max\left\{\frac{1}{\alpha};\ \frac{\chi - \sqrt{\chi^2 - (1-\alpha^2)}}{1-\alpha^2}\right\},$$

$$D = \frac{\chi + \sqrt{\chi^2 - (1-\alpha^2)}}{1-\alpha^2},$$

$$z = \begin{cases} \sqrt{1-\chi^2}, & \text{если } \chi \leqslant \sqrt{2}/2, \\ \chi/2, & \text{если } \chi > \sqrt{2}/2. \end{cases}$$

Теорема 10 [22]. *Если* $z \leqslant \alpha < 1$ *и* $d(x^0) \in [\lambda_0 C,\ \lambda_0 D]$, *то*

$$d(x^k) \leqslant d(x^0)\,(\alpha)^k \quad \forall k$$

(*линейная сходимость к оптимальной точке*).

Если $z \leqslant \alpha < 1$ *и* $d(x^0) < \lambda_0 C$, *то*

$$d(x^k) \leqslant \lambda_0 C\,(\alpha)^k \quad \forall k$$

(*линейная сходимость к оптимальной точке*).

Если $\alpha < z$ *или если* $d(x^0) > \lambda_0 D$, ***то алгоритм может сходиться к точке, не являющейся оптимальной.***

Доказательство см. в [22].

Таким образом, во всех случаях наилучшим коэффициентом сходимости, который мы можем получить (в предположении, что $\chi \leqslant \sqrt{2}/2$), будет $\alpha = z = \sqrt{1-\chi^2}$. С другой стороны, чем более близко значение α к 1 (и отлично от z), тем большей будет свобода выбора параметра λ_0 (начальное перемещение).

Замечание. Для квадратичной функции

$$f(x) = \frac{1}{2}\,x^T A x + bx + c$$

имеем

$$\chi = \frac{2\sqrt{\alpha_{\min}\alpha_{\max}}}{\alpha_{\min} + \alpha_{\max}},$$

где $\alpha_{\min}$, $\alpha_{\max}$ — экстремальные собственные значения матрицы A (предполагаемой положительно определенной). Тогда наилучшим возможным коэффициентом сходимости будет

$$\sqrt{1-\chi^2} = \frac{\alpha_{\max} - \alpha_{\min}}{\alpha_{\max} + \alpha_{\min}}.$$

Это не что иное, как квадратный корень из *отношения Канторовича*.

Таким образом, мы снова приходим к результату, полученному методом наискорейшего спуска (п. 2.12).

Метод сходящегося ряда допускает следующую геометрическую интерпретацию.

Предположим, что нам известна мажоранта R для расстояния $\|x^0 - x^*\|$ между начальной точкой x^0 и оптимальной точкой x^* (предполагается, что оптимальная точка единственна), иначе гово-

ря, оптимальная точка заранее локализована в сфере S^0 с центром x^0 и радиусом R (рис. 9).

Пусть γ^0 — субградиент функции f в точке x^0. Если $\chi = \cos\theta$ есть обусловленность функции f, то x^* обязательно будет принадлежать множеству $\bar{S}^0$ — пересечению сферы S^0 и конуса с вершиной

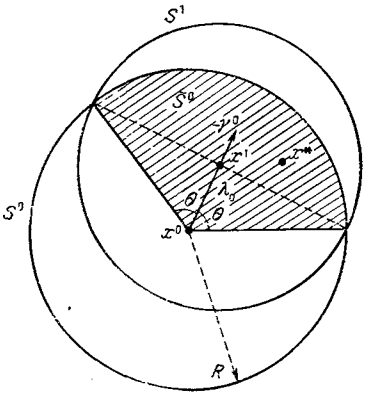


Рис. 9. Геометрическая интерпретация метода сходящегося ряда. Если $\chi = \cos\theta$ есть обусловленность функции f, то точка x^1 определяется формулой

$$x^1 = x^0 - \lambda_0\gamma^0/\|\gamma^0\| \text{ при } \lambda_0 = R\cos\theta = R\chi.$$

Если оптимум x^* первоначально был расположен в шаре S^0 с центром x^0 и радиусом R, то после первой итерации его можно искать в шаре S^1 с центром x^1 и радиусом $R\sqrt{1-\chi^2}$

x^0 и углом 2θ при вершине. Предположим, что $\theta \geqslant 2\pi/3$, т. е. $\chi = \cos\theta \leqslant 1/2$. Заметим, что множество $\bar{S}^0$ целиком содержится в сфере с центром $x^1 = x^0 - \lambda_0\gamma^2/\|\gamma^0\|$ $(\lambda_0 = R\cos\theta = R\chi)$ и радиусом $R\sin\theta = R\sqrt{1-\chi^2}$.

Таким образом, оптимум локализован в сфере S^1 радиуса, строго меньшего R.

Взяв за исходный аргумент x^1, приходим к тому, что если γ^1 — субградиент функции f в точке x^1, то можно локализовать оптимум в сфере S^2 с центром

$$x^2 = x^1 - \lambda_1\frac{\gamma^1}{\|\gamma^1\|}, \text{ где } \lambda_1 = R\chi\sqrt{1-\chi^2},$$

и радиусом $R(\sqrt{1-\chi^2})^2$.

Вообще, ясно, что процесс сводится к выбору шага перемещения

$$\lambda_k = \lambda_0(\alpha)^k,$$

где

$$\lambda_0 = R\chi, \quad \alpha = \sqrt{1-\chi^2},$$

и точка x^k, полученная на k-й итерации, удовлетворяет неравенству

$$\|x^k - x^*\| \leqslant R(\sqrt{1-\chi^2})^k.$$

Тем самым мы пришли к тому геометрическому факту, что метод сходится линейно с коэффициентом x^* сходимости $z = \sqrt{1-\chi^2}$.

Очевидно, что приведенное рассуждение остается справедливым и в том случае, когда значение угла θ переоценивается, т. е. когда мы берем

$$1 > \alpha > \sqrt{1-\chi^2}, \quad \lambda_0 \geqslant \|x^0 - x^*\|\sqrt{1-\alpha^2}.$$

Аналогичная геометрическая интерпретация будет дана в п. 3.12 для алгоритма Шора — Хачияна.

3.7. Сходимость метода 3 (метод релаксации). Этот метод тесно связан с техникой релаксации, используемой при решении систем линейных неравенств (см. [1, 55]), и с техникой последовательных проекций для нахождения точки, принадлежащей пересечению конечного числа выпуклых множеств (см. [28]).

На k-й итерации шаг λ_k выбирается по формуле

$$\lambda_k = \rho \frac{f(x^k) - \bar{f}}{\|\gamma^k\|},$$

где $\bar{f}$ — оценка оптимального значения f^*, ρ — коэффициент (коэффициент релаксации), удовлетворяющий условию $0 < \rho \leqslant 2$.

Этот выбор может быть обоснован следующим образом. Если положить $x^{k+1} = x^k - t\gamma^k$, то, как и в п. 3.6, получим

$$d^2(x^{k+1}) \leqslant d^2(x^k) + t\,[t\|\gamma^k\|^2 - 2(\gamma^k)^T \cdot (x^k - s(x^k))]. \qquad (24)$$

А поскольку функция f выпукла, то

$$(\gamma^k)^T [x^k - s(x^k)] \geqslant f(x^k) - f^*,$$

откуда

$$d^2(x^{k+1}) \leqslant d^2(x^k) + t\,[t\|\gamma^k\|^2 - 2(f(x^k) - f^*)]. \qquad (25)$$

Значит, если выбрать t строго между 0 и $2(f(x^k) - f^*)/\|\gamma^k\|^2$, то получим $d(x^{k+1}) < d(x^k)$, т. е. расстояние между текущей точкой и множеством D оптимальных решений строго убывает. Однако нельзя ничего сказать (без дополнительных условий) о скорости сходимости. Для оценки этой сходимости мы сделаем предположение, что функция f удовлетворяет условию

$$\forall x \notin \Omega \qquad \forall \gamma \in \partial f(x),$$
$$\gamma^T [x - s(x)] \geqslant f(x) - f^* \geqslant \omega \gamma^T [x - s(x)],$$

где ω — действительное число, $0 < \omega < 1$. В некотором смысле параметр ω характеризует кривизну функции f.

Тогда по определению χ (обусловленности функции f) можно записать

$$f(x^k) - f(x^*) \geqslant \omega (\gamma^k)^T [x^k - s(x^k)] \geqslant \omega \chi d(x^k) \|\gamma^k\|,$$

и (25) запишется в виде

$$d^2(x^{k+1}) \leqslant d^2(x^k) + t\,[t\|\gamma^k\|^2 - 2\omega\chi d(x^k)\|\gamma^k\|].$$

Предположив, что χ, $d(x^k)$ известны, и выбрав

$$t = \frac{\omega\gamma d(x^k)}{\|\gamma^k\|},$$

получим

$$d^2(x^{k+1}) \leqslant d^2(x^k) - \omega^2\chi^2 d^2(x^k)$$

и, значит,

$$d(x^{k+1}) \leqslant d(x^k)\sqrt{1 - \omega^2\chi^2},$$

откуда следует линейная сходимость с коэффициентом (множителем) $\sqrt{1-\omega^2\chi^2}$.

Стало быть, в рамках введенных выше условий $\sqrt{1-\omega^2\chi^2}$ есть наилучший множитель сходимости, который можно получить методом 3.

Но, поскольку $\omega < 1$, этот множитель хуже, чем $\sqrt{1-\chi^2}$ — идеальный множитель сходимости, полученный методом 2.

Практически же мы не знаем f^* (а это в точности искомое значение), поэтому следует ограничиться для f^* оценкой $\bar{f}$ и шагом λ_k, выбираемым по формуле

$$\lambda_k = \rho \frac{f(x^k) - \bar{f}}{\|\gamma^k\|}, \qquad 0 < \rho \leqslant 2.$$

Могут представиться несколько случаев.

1. Если $\bar{f} > f^*$ и $0 < \rho < 2$, то можно применить изложенное выше, и при аналогичных условиях можно заключить, что последовательность полученных x^k сходится линейно к точке $\bar{x}$, для которой $f(\bar{x}) \leqslant \bar{f}$.

2. Если $\bar{f} > f^*$ и $\rho = 2$, то можно доказать, что, вообще говоря, точка $\bar{x}$, для которой $f(\bar{x}) \leqslant \bar{f}$, получается за конечное число итераций (однако это число может быть значительным). Это непосредственный результат работ [1, 55] по системам линейных неравенств.

3. Наконец, если $\bar{f} < f^*$, то последовательность полученных значений $f(x^k)$, вообще говоря, не сходится к f^*, если коэффициент ρ поддерживается постоянным. Тогда [29] предлагается устремить $\rho \to 0$, причем сформулировано некоторое число эвристических правил выбора последовательности ρ^k. Несмотря на удовлетворительные результаты, полученные на практике, эти правила пока не имеют полного теоретического обоснования.

Замечание. В случае 3, т. е. когда $\bar{f} < f^*$, можно показать, что поведение последовательности $\{f(x^k)\}$ не будет, в действительности, совсем произвольным. Заменив в (25) t на $\rho(f(x^k) - \bar{f})/\|\gamma^k\|^2$ $(0 < \rho < 2)$, получим

$$d^2(x^{k+1}) \leqslant d^2(x^k) + \rho^2 \frac{(f(x^k) - \bar{f})^2}{\|f^k\|^2} - 2\rho \frac{(f(x^k) - \bar{f})(f(x^k) - f^*)}{\|f^k\|^2},$$

откуда

$$d^2(x^{k+1}) \leqslant d^2(x^k) + \rho \frac{f(x^k) - \bar{f}}{\|\gamma^k\|^2} [2f^* - \rho\bar{f} - (2-\rho) f(x^k)].$$

Из этой формулы следует, что для любого $\varepsilon > 0$ найдется такое i, что

$$f(x^i) < \frac{2f^* - \rho\bar{f}}{2-\rho} + \varepsilon.$$

В самом деле, допустим противное, т. е. что для любого k

$$f(x^k) \geqslant \frac{2f^* - \rho\bar{f}}{2-\rho} + \varepsilon.$$

Допустив, что $\|\gamma^k\| \leqslant c$ (c — фиксированная константа), имеем

$$d^2(x^{k+1}) \leqslant d^2(x^k) - \delta \quad \forall k,$$

где

$$\delta = \varepsilon\rho(2-\rho)\frac{f^* - \bar{f}}{c^2},$$

и мы приходим к противоречию.

Стало быть, всегда можно выбрать из последовательности $\{f(x^k)\}$ подпоследовательность, сходящуюся к значению $(2f^* - \rho\bar{f})/(2-\rho)$.

Это свойство может быть использовано для построения следующего алгоритма.

Возьмем для упрощения $\rho = 1$ и предположим, что для $\bar{f} < f^*$ и $\alpha < 1$ найдется такое фиксированное число итераций, что

$$k \geqslant K \Rightarrow f_{\min}(k) - (2f^* - \bar{f}) \leqslant \alpha[f(x^0) - (2f^* - \bar{f})],$$

где $f_{\min}(k)$ при любом k есть наилучшее из значений, полученных до k-й итерации, считая начальной точку x^0 (число K может оцениваться исходя из обусловленности функции f).

На практике приближения, даже грубого (для определенности с $\alpha = 0{,}2$), истинного значения $2f^* - \bar{f}$ будет достаточно, и тогда число K будет очень небольшим (для определенности K от 5 до 10).

Тогда возможна следующая процедура.

a) Выбрать $\bar{f} < f^*$ (во многих задачах легко получить миноранту оптимального значения). Достаточно выбрать приближение, даже грубое.

Выбрать начальную точку x^0.

b) Осуществить K итераций методом релаксации, начиная с точки x^0.

Пусть $f_{\min}$ — наилучшее полученное значение. Имеем

$$f_{\min} - (2f^* - \bar{f}) \leqslant \alpha[f(x^0) - (2f^* - \bar{f})],$$

откуда получаем

$$f^* \geqslant \frac{f_{\min} + (1+\alpha)\bar{f} - \alpha f(x^0)}{2 - 2\alpha}.$$

Значит, правая часть составляет новую миноранту для f^* (тем более близкую к f^*, чем лучше приближает $f_{\min}$ значение $2f^* - \bar{f}$).

Положить

$$\bar{f} \leftarrow \frac{f_{\min} + (1+\alpha)\bar{f} - \alpha f(x^0)}{2 - 2\alpha}.$$

Принять в качестве новой начальной точки x^0 наилучшую из полученных ранее точек и вернуться к b).

Приведенный выше алгоритм интересен также тем, что в каждый момент можно контролировать отклонение от оптимума наилучшего полученного значения, сравнивая значения $f_{\min}$ и $\bar{f}$. Итерации могут быть прерваны, как только разность $f_{\min} - \bar{f}$ станет меньше фиксированной погрешности.

3.8. Методы Шора растяжения пространства. Эффективность субградиентных алгоритмов, описанных в пп. 3.4—3.7, существенным образом зависит от обусловленности оптимизируемой функции. Чтобы увеличить скорость сходимости, можно пытаться осуществить преобразования координат (замену метрики) с целью улучшения обусловленности. Именно эта идея лежит в основе семейства методов, называемых растяжением (дилатацией) пространства [73—76], комбинирующих принципы субградиентных алгоритмов и методов с переменной метрикой (пп. 2.9—2.11). Предположим, что на k-й итерации субградиентного алгоритма производится замена переменного $x = B_k y$ (где B_k — неособенная $n \times n$-матрица) и что субградиентный алгоритм применяется к функции $\varphi(y) = f(B_k y)$.

Легко видеть, что это приводит к перемещению в пространстве переменных x по направлению

$$d = -B_k B_k^T \gamma,$$

где γ — субградиент функции f в текущей точке x^k.

В самом деле, имеем

$$f(x) \geqslant f(x^k) + \gamma^T (x - x^k) \quad \forall x,$$

откуда, осуществив замену переменного $x = B_k y$, получим

$$\varphi(y) \geqslant \varphi(y^k) + \gamma^T B_k (y - y^k) \quad \forall y.$$

Это доказывает, что $B_k^T \gamma$ есть субградиент функции φ в точке $y^k = B_k^{-1} x^k$.

Если λ_k — шаг перемещения, выбранный в субградиентной процедуре, примененной к $\varphi(y)$, то

$$y^{k+1} = y^k - \lambda_k B_k^T \gamma$$

и, следовательно, $x^{k+1} = B_k y^{k+1}$ будет получен по формуле

$$x^{k+1} = B_k y^k - \lambda_k B_k B_k^T \gamma = x^k - \lambda_k B_k B_k^T \gamma,$$

где B — неособенная матрица, $B_k B_k^T$ — симметричная положительно определенная матрица, и если функция f непрерывно дифференцируема, то направление $(-B_k B_k^T \gamma)$ есть направление спуска. Значит, формула (26) аналогична той, которая используется в методах с переменной метрикой (квазиньютоновских методах); существенная разница заключается в способе представления матриц B_k на каждой итерации. Для этого было введено понятие оператора растяжения [73].

Определение 1. Пусть $d \in \mathbf{R}^n$, $\|d\| = 1$. *Оператором растяжения* в направлении d с коэффициентом $\alpha \geqslant 0$ называется линейный оператор $R_\alpha(d)$, определяемый соотношением

$$R_\alpha(d) x = x + (\alpha - 1) x_d \quad \forall x \in \mathbf{R}^n,$$

где $x_d = (x^T d) d$ есть проекция вектора x на направление d.

Таким образом, оператор $R_\alpha(d)$ оставляет инвариантным любой вектор, ортогональный d, но умножает на $\alpha \geqslant 0$ всякий вектор, коллинеарный d (α может быть как больше, так и меньше единицы).

Заметим, что $R_\alpha(d)$ может быть представлен также в виде

$$R_\alpha(d)x = x + (\alpha - 1)(dd^T)x, \quad \forall x,$$

и, следовательно, матрица, ассоциированная с этим оператором, есть

$$I + (\alpha - 1)(dd^T),$$

где I — единичная $n \times n$-матрица.

В дальнейшем мы не будем делать различия между понятиями оператора $R_\alpha(d)$ и ассоциированной матрицы.

Действие оператора растяжения иллюстрируется на рис. 10.

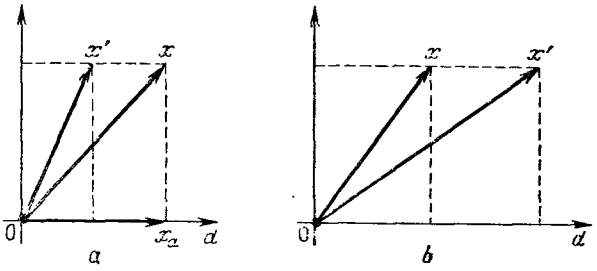


Рис. 10. Иллюстрация понятия оператора растяжения в $\mathbf{R}^2$ для $\alpha = 1/2$ (*a*) и для $\alpha = 2$ (*b*)

В методах растяжения пространства матрицы B_k итеративно изменяются по формулам

$$B_{k+1} = B_k R_{\alpha_k}(d_k),$$

где коэффициент α_k и направление d_k ($\|d_k\| = 1$) зависят от выбранного метода (см. пп. 3.9—3.12).

Исходная матрица, как правило, берется единичной ($B_0 = I$). Следовательно, матрица B_k, полученная на любой k-й итерации, есть произведение k операторов растяжения

$$B_k = R_{\alpha_0}(d_0) R_{\alpha_1}(d_1) \ldots R_{\alpha_{k-1}}(d_{k-1}).$$

Общая структура алгоритма растяжения пространства может быть теперь описана следующим образом.

а. Инициализация:

x^0 — выбранная начальная точка, γ^0 — градиент (или субградиент) функции f в x^0;

$B_0 = I$ (тождественная матрица);

$k = 0$.

b. На k-й итерации x^k — текущая точка. Определить x^{k+1} формулой

$$x^{k+1} = x^k - \lambda_k B_k B_k^T \gamma^k,$$

где λ_k — шаг перемещения. Определить γ^{k+1} — градиент (или субградиент) функции f в x^{k+1}. Пусть d_k ($\|d_k\| = 1$) — направление (оп-

ределенное, вообще говоря, исходя из γ^k или из разности $\gamma^{k+1}-\gamma^k$), α_k — коэффициент растяжения на k-м шаге. Определить

$$B_{k+1}=B_kR_{\alpha_k}(d_k).$$

c. Если тест остановки выполнен, то конец. Иначе присвоить $k\leftarrow k+1$ и возвратиться к b).

Отметим, что эквивалентность методов растяжения пространства с методами переменной метрики, использующими формулы коррекции ранга 1 (п. 2.9), была установлена в [78].

Очевидно, что существует столько же алгоритмов этого типа, сколько и способов выбора коэффициентов λ_k, α_k и направления d_k при каждой итерации. Теперь мы перейдем к рассмотрению основных вариантов, которые были изучены и применены на практике.

3.9. Проектирование на подпространство, порожденное последовательно встречающимися градиентами. Этот метод, связанный с семейством алгоритмов сопряженных градиентов, состоит в выборе:

$$\alpha_k=0 \quad \forall k,$$

$$d_k=\frac{r_k}{\|r_k\|}, \text{ если } r_k=B_k^T[\gamma^{k+1}-\gamma^k]\neq 0;$$

а λ_k выбрано так, чтобы минимизировать f в направлении $-B_kB_k^T\gamma^k$.

Значит, на k-й итерации матрица B_k имеет вид

$$B_k=R_0(d_0)R_0(d_1)\ldots R_0(d_{k-1}),$$

где $R_0(d_i)$ — оператор проектирования на подпространство, ортогональное к d_i $(i=0,\ 1,\ \ldots,\ k-1)$. Тогда легко видеть, что произведение $\prod_{i=0}^{k-1}R_0(d_i)$ есть не что иное, как оператор проектирования на ортогонально дополнительное подпространство (в $\mathbf{R}^n$) к подпространству, порожденному $d_0,\ d_1,\ \ldots,\ d_{k-1}$. Обратимся теперь к случаю, когда минимизируемая функция f непрерывно дифференцируема, и предложим, что на некотором произвольном этапе k $(0\leqslant k\leqslant n-1)$

$$r_k=B_k^T[\gamma^{k+1}-\gamma^k]=0.$$

Отсюда в силу регулярности матрицы B_k имеем $\gamma^{k+1}=\gamma^k$. Покажем теперь, что $\gamma^{k+1}=\gamma^k=0$, если f непрерывно дифференцируема.

Предположим, что $\gamma^k\neq 0$. Поскольку x^{k+1} определяется так, чтобы

$$(\gamma^{k+1})^T[x^{k+1}-x^k]=0$$

и чтобы $\gamma^{k+1}=\gamma^k$, то

$$\lambda_k(\gamma^k)^TB_kB_k^T\gamma^k=0.$$

В силу положительной определенности матрицы BB_k^T направление $-B_kB_k^Ty^k$ будет направлением спуска, и, значит, $x^{k+1}\neq x^k$, т. е. $\lambda_k\neq 0$.

Итак, мы должны иметь

$$(\gamma^k)^T B_k B_k^T \gamma^k = 0,$$

откуда получаем противоречие с тем, что $\gamma^k \neq 0$.

Следовательно, если f — непрерывно дифференцируемая функция и $r_k = 0$ $(0 \leqslant k \leqslant n-1)$, то $\gamma^k = 0$ и x^k есть стационарная точка функции f.

Предположим теперь, что $r_k \neq 0$ $(k = 0, 1, \ldots, n-1)$. Тогда легко видеть, что $B_n = 0$. В самом деле,

$$B_n = \prod_{k=0}^{n-1} R_0(d_k)$$

есть оператор проектирования на дополнительное ортогональное подпространство (в $\mathbf{R}^n$) всего $\mathbf{R}^n$.

Таким образом, ясно, что алгоритм требуется периодически обновлять на каждых n итерациях, принимая в качестве новой начальной точки полученную точку x^k, а в качестве новой матрицы B — тождественную матрицу I.

Можно показать, что когда рассматриваемый метод применяется к непрерывно дифференцируемой функции, он может рассматриваться как вариант метода сопряженных градиентов. Стало быть, мы получаем при классических условиях свойство квадратичной сходимости на n-шагах, как показывает

Теорема 11. *Если функция f дважды непрерывно дифференцируема в окрестности $\mathscr{V}$ оптимума x^*, имеет положительно определенный гессиан и удовлетворяет условию Липшица*

$$\|\nabla^2 f(x) - \nabla^2 f(x')\| \leqslant L\|x - x'\| \qquad \forall x \in \mathscr{V}, \quad \forall x' \in \mathscr{V},$$

то найдутся такие постоянные K и $c > 0$, что при $k > K$

$$\|x^{k+n} - x^*\| \leqslant c\|x^k - x^*\|^2.$$

Доказательство см. в [76].

Только что описанные методы могут быть использованы для функций, не являющихся всюду дифференцируемыми, и, в частности, для выпуклых субдифференцируемых функций.

3.10. Растяжение пространства в направлении разности двух последовательных субградиентов [74, 76]. Речь идет об обобщении предыдущего метода, в котором коэффициент растяжения α берется постоянным и строго положительным:

$$\alpha_k = \alpha, \qquad 0 < \alpha < 1, \quad \forall k,$$

$$d_k = \frac{r_k}{\|r_k\|} \text{ при } r_k = B_k^T[\gamma^{k+1} - \gamma^k],$$

а λ_k определено так, чтобы минимизировать f в направлении $-B_k B_k^T \gamma^k$.

Можно показать [78], что в матричной форме (для матриц $H_k = B_k B_k^T$) этот алгоритм приводит к формуле коррекции ранга 1,

записываемой в виде

$$H_{k+1} = H_k - (1-\alpha^2)\frac{(H_k e_k)(H_k e_k)^T}{e_k^T H_k e_k},$$

где

$$e_k = \gamma^{k+1} - \gamma^k.$$

Определенный таким образом алгоритм применим к функциям, не являющимся всюду дифференцируемыми, и в этом случае его поведение может быть существенно отличным от алгоритма п. 3.9. В самом деле, направление $-B_k B_k^T \gamma^k$ не обязательно будет направлением спуска. Если это произойдет, мы получим $x^{k+1} = x^k$, и алгоритм уже не будет прогрессировать.

Для исправления этой ситуации было предложено [74] модифицировать этап b) общего алгоритма (п. 3.8), заменив его этапом

b′) На k-й итерации текущая точка есть x^k. Определить x^{k+1} по формуле

$$x^{k+1} = x^k - \lambda_k B_k B_k^T \gamma^k,$$

где λ_k выбрано так, чтобы минимизировать f в направлении $-B_k B_k^T \gamma^k$.

Если $\lambda_k = 0$, то положить:

$$x^{k+1} \leftarrow x^k,$$
$$x^k \leftarrow x^{k-1},$$
$$\gamma^{k+1} \leftarrow \gamma^k,$$
$$\gamma^k \leftarrow \gamma^{k-1},$$

перейти к b″).

Если $\lambda_k \neq 0$, то определить γ^{k+1}, субградиент функции f в x^{k+1} и перейти к b″).

b″) Положить

$$d_k = \frac{r_k}{\|r_k\|} \quad \text{при} \quad r_k = B_k^T [\gamma^{k+1} - \gamma^k]$$

и определить

$$B_{k+1} = B_k R_\alpha(d_k),$$

где α — выбранный постоянный коэффициент растяжения.

Таким образом, даже если алгоритм остается неподвижным в одной и той же точке в продолжение нескольких последовательных итераций, матрицы B_k продолжают изменяться (из-за растяжения пространства в направлении разности двух последних различных градиентов), что позволяет определить на каждом шаге итерации новое направление возможного перемещения. Алгоритм деблокируется, как только обнаруживается некоторое направление спуска.

Заметим, что преимущество этого алгоритма перед методами субградиента, описанными в пп. 3.5 и 3.7, состоит в том, что последовательность порожденных им значений $f(x^k)$ монотонно убывает.

Можно доказать их сходимость к минимуму $f(x)$, но необходимы относительно ограничительные условия (см. [74]).

Хотя могут встретиться и патологические случаи, в которых сходимости нет, но численные эксперименты, ссылки на которые можно найти в литературе, позволяют показать, что этот алгоритм действительно представляет собой один из наиболее эффективных известных в настоящее время методов для решения недифференцируемых задач оптимизации (см., например, [44]).

3.11. Растяжение пространства в направлении субградиента в текущей точке [73]. В этом методе на каждой k-й итерации выбираем

$$\alpha_k = \alpha < 1 \quad \forall k,$$

$$d_k = \frac{B_k^T \gamma^k}{\| B_k^T \gamma^k \|}, \quad \lambda_k = \rho \frac{f(x^k) - m}{\| B_k^T \gamma^k \|^2},$$

где ρ (коэффициент релаксации) удовлетворяет условию $0 < \rho \leqslant 2$ и где m — приближение оптимального значения f^*.

Можно показать [78], что в матричной форме $\left(H_k = B_k B_k^T\right)$ этот алгоритм приводит к формуле коррекции ранга 1, записываемой в виде

$$H_{k+1} = H_k - (1 - \alpha^2) \frac{\left(H_k \gamma^k\right)\left(H_k \gamma^k\right)^T}{(\gamma^k)^T H_k \gamma^k}.$$

Относительно сходимости метода справедлива

Теорема 12 [73]. *Пусть f — выпуклая функция (не обязательно дифференцируемая) и пусть существует такая постоянная $M = 1/\omega > 1$, что*

$$f(x) - f^* \geqslant \omega \gamma^T (x - s(x)) \qquad \forall x, \ \forall \gamma \in \partial f(x),$$

где $s(x)$ — проекция точки x на множество оптимальных точек.

Предположим, что выбраны

$$\alpha_k = \alpha = \frac{M-1}{M+1} \quad \forall k$$

и что

$$\rho = \frac{2M}{M+1}.$$

Тогда если $m \geqslant f^*$, то можно выбрать такую подпоследовательность $\{x^i\}$ последовательности $\{x^k\}$, что $\{f(x^i)\}$ сходится к m.

Кроме того, если выбрано $m = f^*$, то найдется такая постоянная c, что

$$f(x^i) - m \leqslant c\alpha^{i/n}$$

(иными словами, сходимость к m линейна с множителем сходимости $q = \sqrt[n]{\alpha}$).

Доказательство см. в [73].

Заметим, что условие теоремы 12 аналогично условию п. 3.7.

Предыдущий результат использован в [73] для построения алгоритма, действующего посредством последовательного установления

значений m (при помощи дихотомии) и позволяющего получить последовательность $\{m^k\}$, сходящуюся к оптимальному (неизвестному) значению $m^* = f(x^*)$.

3.12. Метод отсечения с растяжением пространства [36, 75]. Вариант предыдущего метода, исключительно интересный с точки зрения его теоретических следствий, был предложен в [75].

Пусть R — мажоранта нормы $\|x^0 - x^*\|$ — расстояния между начальной точкой и оптимальным решением. Тогда метод состоит в выборе (n — размерность пространства):

$$\alpha_k = \alpha = \sqrt{\frac{n-1}{n+1}} \quad \forall k,$$

$$d_k = \frac{B_k^T \gamma^k}{\|B_k^T \gamma^k\|}, \quad \lambda_k = \frac{R}{n+1} \frac{(\beta)^k}{\|B_k^T \gamma^k\|},$$

где $\beta = \frac{n}{\sqrt{n^2-1}}$. Это — *метод Шора — Хачияна.*

Можно показать (см. [20, 36]), что алгоритм принимает следующую эквивалентную форму относительно матриц $H_k = B_k B_k^T$:

а) Пусть x^0 — выбранная начальная точка, γ^0 — субградиент функции f в точке x^0. Положить

$H_0 = RI$, где R — мажоранта нормы $\|x^0 - x^*\|$, I — единичная $n \times n$-матрица, $k = 0$.

b) На k-й итерации x^k — текущая точка, γ^k — субградиент в точке x^k.

Определить x^{k+1} по формуле

$$x^{k+1} = x^k - \frac{1}{n+1} \frac{H_k \gamma^k}{\sqrt{(\gamma^k)^T H_k \gamma^k}}.$$

Положить

$$H_{k+1} = \frac{n^2}{n^2-1} \left(H_k - \frac{2}{n+1} \frac{(H_k \gamma^k)(H_k \gamma^k)^T}{(\gamma^k)^T H_k \gamma^k} \right).$$

c) Если проверка теста на остановку удовлетворительна, **то** конец.

Иначе положить $k \leftarrow k+1$ и вернуться к b).

Можно придать этой процедуре следующую геометрическую интерпретацию (иллюстрацию двумерного случая см. на рис. 11, 12). Исходим из того, что искомый оптимум x^* содержится в сфере S^0 с центром x^0 и радиусом R. Если γ^0 — субградиент функции f в x^0, то x^* непременно содержится в полусфере

$$\bar{S}^0 = S^0 \cap \{x \mid (\gamma^0)^T (x - x^0) \leqslant 0\}.$$

Пусть S^1 — эллипсоид минимального объема, содержащий $\bar{S}^0$ (а значит, $x^* \in S^1$). Можно показать, что S^1 — единственный эллипсоид, имеющий одинаковое с S^0 пересечение с гиперплоскостью $\{x \mid (\gamma^0)^T (x - x^0) = 0\}$ и касающийся S^0 в точке A ($x^0 A$ есть направление, сопряженное к подпространству $\{x \mid (\gamma^0)^T x = 0\}$ относительно

эллипсоида S^0) (см. рис. 11). Эллипсоид S^1 есть множество точек, удовлетворяющих неравенству

$$\|H_1^{-1}(x - x^1)\| \leqslant R\frac{n}{\sqrt{n^2-1}}.$$

С другой стороны, если $v(S^0)$ и $v(S^1)$ — объемы эллипсоидов S^0 и S^1, то нетрудно доказать соотношение

$$v(S^1) = qv(S^0),$$

$$q = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n < e^{-1/2(n+1)} < 1.$$

Таким образом, на первой итерации приходим к локализации x^* в эллипсоиде S^1 с центром x^1 и объемом, строго меньшим объема эллипсоида S^0.

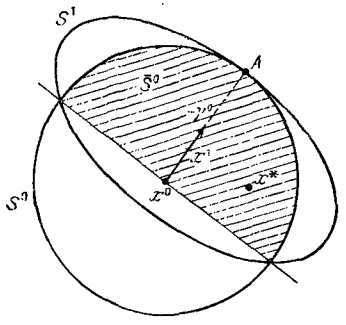


Рис. 11. Иллюстрация метода отсечения с растяжением пространства. Первоначально оптимум x^* располагается в шаре S^0 с центром в x^0 (x^0 — начальная точка) и радиусом R. Если γ^0 — субградиент f в x^0, то x^* обязательно принадлежит заштрихованному полушару $\bar{S}^0$, который является пересечением S^0 с полупространством $\{x|\gamma^0)^T(x - x^0) \leqslant 0\}$. Тогда точка x^1 выбирается как центр эллипсоида S^1 минимального объема, содержащего $\bar{S}^0$. Необходимо $x^* \in S^1$

В общем случае на k-й итерации γ^k есть субградиент функции f в точке x^k (итерация $k = 1$ представлена на рис. 12). Обязательно x^k содержится в полуэллипсоиде

$$\bar{S}^k = S^k \cap \{x|(y^k)^T(x - x^k) \leqslant 0\}.$$

Пусть S^{k+1} — эллипсоид минимального объема, содержащий $\bar{S}^0$ и являющийся единственным эллипсоидом, имеющий одинаковое с

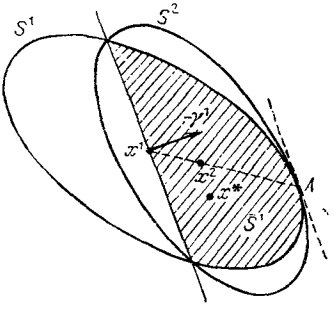


Рис. 12. Если γ^1 — субградиент f в x^1, то оптимум x^* содержится в полуэллипсоиде $\bar{S}^1$ (он заштрихован), который является пересечением S^1 с полупространством $\{x|(\gamma^1)^T(x - x^1) \leqslant 0\}$. Тогда точка x^2 выбирается как центр эллипсоида S^2 минимального объема, содержащего $\bar{S}^1$. По построению $x^* \in S^2$

S^k пересечение с гиперплоскостью $\{x|(\gamma^k)^T(x - x^k) = 0\}$ и касающийся S^k в A (x^kA — направление, сопряженное к подпространству $\{x|(\gamma^k)^Tx = 0\}$ относительно эллипсоида S^k) (см. рис. 12). Точка

x^{k+1} выбирается в качестве центра нового эллипсоида S^{k+1}, и тогда точка x^* локализуется в эллипсоиде с центром x^{k+1} и объемом

$$v(S^{k+1}) = qv(S^k) = q^{k+1}v(S^0).$$

Таким образом, объемы эллипсоидов, порожденных в процессе процедуры, стремятся к нулю со скоростью геометрической прогрессии со знаменателем

$$q = \sqrt{(n-1)/(n+1)}\,(n/\sqrt{n^2-1})^n,$$

зависящим лишь от размерности пространства.

Описанный выше метод имеет своим началом фундаментальный результат по теории сложности. В самом деле, как показано в [36], сложность этого алгоритма, когда он применяется к решению системы линейных уравнений, или, в более общем случае, к решению линейных задач с целыми коэффициентами, есть полиномиальная функция от размера задачи (число бинарных элементов, необходимых для описания задачи).

Однако если этот результат заслуживает большого внимания, ввиду его широкого теоретического применения, то численные расчеты, представленные многими авторами, дают основания подозревать, что алгоритм весьма далек от конкуренции с симплекс-методом (см. гл. 2). Даже для задач очень небольшого объема сходимость исключительно медленна, и алгоритм выглядит весьма неустойчивым численно (матрицы B_k становятся в ходе итераций очень плохо обусловленными).

В противовес симплекс-методу (который в худшем случае требует времени счета, экспоненциально возрастающего с ростом размера, но который очень эффективен в среднем на якобы любых задачах практического происхождения) представляется, что поведение в среднем алгоритма Шора — Хачияна не отличается существенно от его поведения в худшем случае.

В заключение заметим, что алгоритм отсечения с растяжением пространства может быть связан с семейством методов центров [33, 34] или методов центроидов [49, 58]. Методы центроидов могут быть кратко описаны следующим образом.

Предположим, что на k-й итерации минимум функции f может быть локализован в полиэдре π^k. Тогда если выбрать в качестве точки x^{k+1} центр тяжести полиэдра и определить полиэдр π^{k+1} как

$$\pi^{k+1} = \pi^k \cap \{x \mid (\gamma^{k+1})^T(x - x^{k+1}) \leqslant 0\}, \quad \gamma^{k+1} \in \partial f(x^{k+1}),$$

то [49] можно показать, что объемы $v(\pi^{k+1})$, $v(\pi^k)$ будут удовлетворять соотношению

$$v(\pi^{k+1}) \leqslant \left[1 - \left(1 - \frac{1}{n+1}\right)^n\right] v(\pi^k).$$

Стало быть, при любой размерности пространства имеем

$$v(\pi^{k+1}) \leqslant qv(\pi^k), \qquad q = 1 - 1/e.$$

Итак, получен замечательный результат: объемы стремятся к нулю со скоростью геометрической прогрессии, не зависящей одно-

временно ни от обусловленности функции, ни от размерности пространства. Однако поиск центра тяжести полиэдра в $\mathbf{R}^n$ есть задача, которую мы не умеем эффективно решать, поэтому приведенный результат непригоден для прямого построения алгоритма, позволяющего достигать указанной выше скорости сходимости.

Зато с теоретической точки зрения он представляет очень большой интерес, ибо он составляет «предельную теорему»: в самом деле, можно показать, что алгоритм центров тяжести оптимален в том смысле, что не существует алгоритма, использующего ту же информацию и имеющего более быструю сходимость (см. статью Юдина и Немировского (1976), цитируемую в [64]).

3.13. Дальнейшие продвижения в оптимизации функций, не являющихся всюду дифференцируемыми. Область оптимизации функций, не являющихся всюду дифференцируемыми, уже много лет составляет исключительно интенсивный объект исследований в математическом программировании как по числу разработок, так и по их важности (двойственность в целочисленном программировании, методы различных декомпозиций, задачи аппроксимации в смысле Чебышева, задачи на max — min и др.).

Распространение методов сопряженного градиента на выпуклые или вогнутые субдифференцируемые функции было исследовано в [41, 81]. Методы пучков, предложенные в [42—44], могут рассматриваться как расширение методов спуска, обеспечивающее монотонное убывание минимизируемой функции (свойство, которым, вообще говоря, не обладают алгоритмы, изучавшиеся в пп. 3.5—3.7).

Однако кажется, что даже для наиболее разработанных из этих алгоритмов полученные скорости сходимости в лучшем случае сублинейны. Стало быть, эти методы еще далеки от достижения результатов, подобных результатам в дифференцируемой оптимизации.

В результате сравнительного изучения основных алгоритмов недифференцируемой оптимизации [43, 44] можно заключить, что методы растяжения пространства из п. 3.10 наиболее эффективны (но сходимость к оптимальной точке не всегда гарантируется).

Наконец, в [69] мы находим синтез работ, относящихся к распространению понятий субградиента и субдифференциала на функции, не обладающие свойством выпуклости или вогнутости. Эти понятия, по-видимому, будут лежать в основе построения теоретического фундамента новых, самых общих методов в недифференцируемой оптимизации.

§ 4. Методы оптимизации без производных

Какие методы можно использовать, когда функция f недифференцируема или когда нельзя применять алгоритм градиента или субградиента (например, когда невозможно вычислить градиент или субградиент)?

4.1. Циклические методы релаксации. Один из методов, естественно приходящих на ум, состоит в том, чтобы, исходя из начальной точки x^0, минимизировать сначала функцию по переменной x_1,

а другие переменные оставить фиксированными, затем зафиксировать $x_1, x_3, \ldots, x_n$, а минимизировать по x_2, и т. д. Для этого можно воспользоваться одним из методов одномерной оптимизации, описанных в гл. 3.

После того как каждая переменная будет один раз использована, можно повторить один или несколько совершенно таких же циклов.

Можно доказать, что этот метод сходится при условии непрерывной дифференцируемости функции f.

Если переменные слабо взаимодействуют между собой, быстро получаем хорошее приближение оптимума (для разделяемых функций достаточно одного цикла).

Если это не так, то сходимость будет, вообще говоря, гораздо более медленной, чем для метода наискорейшего спуска. Хотя мы не знаем общего результата о сходимости метода этого типа, исследование квадратичного случая показывает, что скорость сходимости для одного цикла (т. е. для n одномерных минимизаций) в худшем случае меньше, чем на единственном шаге метода наискорейшего спуска. Эти теоретические результаты проверены на самых общих классах функций.

Некоторые авторы пытались улучшить метод, располагая переменные в порядке, отличном от естественного. Например, предлагалось минимизировать на каждом шаге по переменной, для которой градиент (если он существует) имеет самое высокое абсолютное

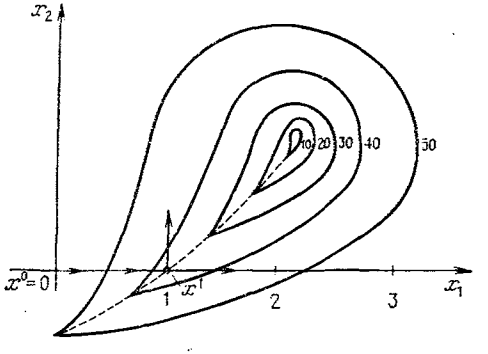


Рис. 13. Отправляясь из $x^0 = (0, 0)$ и минимизируя в направлении x_1, получаем точку $x^1 = (1, 0)$. Точка $x^1 = (1, 0)$ является оптимумом одновременно и в направлении x_1 и в направлении x_2. Следовательно, метод релаксации останавливается. Однако полученная точка отнюдь не является локальным оптимумом задачи

значение. Такие варианты могут достигать лучших результатов на конкретных задачах, но не позволяют преодолеть сформулированную выше сложность в общем виде.

Заметим, наконец, что на некоторых типах функций не только сходимость может быть медленной, но и полученная точка может не оказаться локальным оптимумом. В частности, этот случай может иметь место для функций, не являющихся всюду дифференцируемыми, где алгоритм может остановиться на ребре (см. рис. 13).

С точки зрения всех этих рассуждений алгоритм, который будет описан ниже, вообще говоря, предпочтительнее.

4.2. Алгоритм Пауэлла [65]. Этот метод без производных может быть отнесен к семейству методов сопряженных направлений.

В самом деле, будучи применен к квадратичной функции

$$f(x)=(1/2)x^TAx+b^Tx+c,$$

он осуществляется посредством последовательных одномерных минимизаций вдоль сопряженных направлений (определяемых по мере осуществления процедуры).

Алгоритм может применяться на всех функциях (не обязательно квадратичных), очевидно, без гарантии глобальной оптимальности полученного решения.

Алгоритм Пауэлла существенным образом опирается на следующую идею. Предположим, прежде всего, что отыскивается минимум квадратичной функции f последовательно вдоль p сопряженных направлений $d_1, d_2, \ldots, d_p$ $(p<n)$, *исходя каждый раз из последней* найденной точки.

Таким образом, начиная с точки x^0, шаг за шагом строится последовательность $x^1, x^2, \ldots, x^p$, определенная соотношениями

$$f(x^1)=f(x^0+\lambda_1 d_1)=\min_{\lambda}\{f(x^0+\lambda d_1)\},$$

$$f(x^2)=f(x^1+\lambda_2 d_2)=\min_{\lambda}\{f(x^1+\lambda d_2)\},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$f(x^p)=f(x^{p-1}+\lambda_p d_p)=\min_{\lambda}\{f(x^{p-1}+\lambda d_p)\}.$$

Предположим, далее, что эта операция повторяется, на этот раз с началом в точке $y^0\neq x^0$; получим другую последовательность $y^1, y^2, \ldots, y^p$.

Тогда, как показывает следующий результат, направление y^p-x^p в общем случае будет сопряженным по отношению к p направлениям $d_1, d_2, \ldots, d_p$.

Теорема 13. *Пусть f — квадратичная функция, A — ее гессиан (по предположению положительно определенный). Пусть x^p (соответственно y^p) — точка, полученная одномерной оптимизацией последовательно вдоль p сопряженных направлений $d_1, d_2, \ldots, d_p$, исходя из точки x^0 (соответственно y^0).*

Тогда если $y^p\neq x^p$, то направление $d_{p+1}=y^p-x^p$ сопряжено по отношению к $d_1, d_2, \ldots, d_p$.

Доказательство. По свойству 1 п. 2.5 x^p (соответственно y^p) есть минимум функции $f(x)=(1/2)x^TAx+b^Tx+c$, суженной на аффинное многообразие $V^p(x^0)$ (соответственно $V^p(y^0)$), порожденное набором $(d_1, d_2, \ldots, d_p)$ и проходящее через x^0 (соответственно y^0). Стало быть,

$$d_i^T\nabla f(x^p)=d_i^T(Ax^p+b)=0,$$

$$d_i^T\nabla f(y^p)=d_i^T(Ay^p+b)=0 \qquad \forall i,\quad 1\leqslant i\leqslant p.$$

В результате вычитания сразу получаем

$$d_i^TA(y^p-x^p)=0 \qquad \forall i,\quad 1\leqslant i\leqslant p.$$

Тем самым показано, что если $y^p \neq x^p$, то $d_{p+1} = y^p - x^p$ есть направление, сопряженное относительно $d_1, d_2, \ldots, d_p$.

Заметим, что условие $y^p \neq x^p$ может выполняться только в том случае, если y^0 не принадлежат аффинному многообразию $V^p(x^0)$.

Теперь можно привести описание текущего этапа алгоритма Пауэлла.

а) Выбрать начальную точку x^0 и n линейно независимых направлений $d_1, d_2, \ldots, d_n$ (эти направления не обязательно все попарно сопряжены, но при необходимости могут изменяться в процессе итераций; как именно, мы увидим ниже. Первоначально можно исходить из n направлений, определяемых координатными осями).

Введем теперь семейство $x^1, x^2, \ldots, x^n$ точек, удовлетворяющих соотношениям

$$f(x^i) = \min_{\lambda} f(x^{i-1} + \lambda d_i) = f(x^{i-1} + \lambda_i d_i).$$

b) Заметим, что нельзя без предосторожностей заменять произвольно одно направление новым направлением $x^n - x^0$. В самом деле, некоторые перемещения λ_i могут быть нулевыми (без улучшения в рассматриваемом направлении d_i): тогда, заменив направление $x^n - x^0$ на одно из этих направлений d_i, мы получили бы новое множество линейно зависимых направлений (заметим еще, что $x^n - x^0$ есть линейная комбинация направлений d_i, для которых $\lambda_i \neq 0$).

В квадратичном случае доказано, что, нормируя направления d_i посредством $d_i^T A d_i = 1$ (где A — матрица квадратичной формулы), получаем определитель матрицы $D = [d_1 d_2 \ldots d_n]$, максимальный в том и только том случае, если направления d_i взаимно сопряжены (относительно A). Отсюда заключаем, что нежелательно заменять $x^n - x^0$ на одно из направлений d_i, если эта замена не увеличивает определитель матрицы D. Для вычисления этого определителя при каждой возможной замене, весьма дорогостоящего по времени счета (и в неквадратичном случае невозможного), предложено применить следующее правило.

c) Пусть $f_0 = f(x^0)$, $f_1 = f(x^n)$ и пусть $f_2 = f(2x^n - x^0)$ (значение функции f в точке, симметричной x^0 относительно x^n). Пусть, с другой стороны,

$$\Delta = \max_{i=1,\ldots,n} \{f(x^{i-1}) - f(x^i)\},$$

причем максимум достигается для индекса m (**наибольшее уменьшение значения функции f, полученного в а), в направлении d_m**).

Тогда могут представиться два случая.

d) Если $f_2 \geqslant f_0$ и/или если

$$(f_0 - 2f_1 + f_2)(f_0 - f_1 - \Delta)^2 \geqslant \frac{1}{2}\Delta(f_0 - f_2)^2,$$

то для последующей итерации использовать прежние **направления** $d_1, d_2, \ldots, d_n$ с точкой x^n в качестве новой начальной точки.

Если $f_2 < f_0$ и

$$(f_0 - 2f_1 + f_2)(f_0 - f_1 - \Delta)^2 < \frac{1}{2}\Delta(f_0 - f_2)^2,$$

то снова отыскивать минимум функции f в направлении $x^n - x^0$. Полученная точка будет взята в качестве новой начальной точки x^0 для следующей итерации. С другой стороны, направление заменяется направлением $d = x^n - x^0$, причем порядок, в котором эти направления будут использоваться в процессе итераций, будет следующим:

$$(d_1, d_2, \ldots, d_{m-1}, d_{m+1}, \ldots, d_m, d).$$

При помощи теоремы 13 доказывается, что метод порождает взаимно сопряженные направления, когда он применяется к квадратичной функции.

Для непрерывно дифференцируемых функций f можно показать, что метод Пауэлла сходится к оптимуму функции f, если она строго выпукла. В других случаях он сходится к стационарной точке (локальный оптимум).

Поскольку речь идет о методе сопряженных направлений, то мы имеем асимметрическое поведение метода сопряженных направлений и, при тех же условиях, что в п. 2.12, сходимость будет суперлинейной на n-шагах.

Предлагались и другие методы минимизации без производных. Упомянем, в частности, алгоритм Мифлина [53], относительно которого можно показать, что речь идет о квазиньютоновском методе (отсюда суперлинейная сходимость на одном шаге). За синтезированным изложением отсылаем к [3].

СПИСОК ЛИТЕРАТУРЫ

1. Agmon S. (1954), The Relaxation Method for Linear Inequalities, *Canad. J. Math.* 6, p. 382-392.
2. Best M. J. (1978), A Quasi-Newton method can be obtained from a method of conjugate directions, *Mathematical Programming* 15, p. 189-199.
3. Brent R. T. (1973), *Algorithms for minimization without derivatives*, Prentice Hall.
4. Broyden C. G. (1970), The convergence of a class of double-rank minimization algorithms 2: the new algorithm, *Journal Institute of Math. and its Appl.* 6, p. 222-231.
5. Broyden C. G., Dennis J. E., More J. J. (1973), On the Local and Superlinear Convergence of Quasi-Newton Methods, *J. Institute of Math. and its Appl.* 12, p. 223-245.
6. Camerini P. M., Fratta L., Maffioli F. (1975), On Improving Relaxation Methods by Modified Gradient Techniques, *in*: *Nondifferentiable Optimization*, (M. L. Balinski et P. Wolfe eds.), Mathematical Programming Study 2, Elsevier, New Yark, North Holland.
7. Cauchy A. L. (1987), Méthode générale pour la résolution des systèmes d'équations simultanées, *Comptes Rendus Acad. Sc. Paris*, XXV, p. 536-538.
8. Cheney W., Goldstein A. A. (1959), Newton's Method for Convex Programming and Chebishev Approximation, *Numerische Mathematik* 1, 5, p. 253-268.
9. Clarke F. H. (1975), Generalized Gradients and Applications, *Trans. Am. Math. Soc.* 205, p. 247-262.

10. Curry H. (1944), The Method of Steepest Descent for Nonlinear Minimization Problems, *Quart. Appl. Math.* 2, p. 258-261.
11*. Demyanov V. F. (1971), On the Maximization of a Certain Nondifferentiable Function, *J. Optimization Theory and Appl.* 7, p. 75-89.
12. Dennis J. E. (1976), A brief survey of convergence results for quasi-Newton methods, in: *Nonlinear Programming.* S. I. A. M.-AMS Proceedings, vol. 9 (R. W. Cotte, C. E. Lemke Eds.), p. 185-199.
13. Dennis J. E., More J. J. (1974), A Characterization of superlinear convergence and its application to quasi-newton methods, *Mathematics of Computation*, vol. 28, n° 126, p. 549-560.
14. Dennis J. E., More J. J. (1977), Quasi-Newton methods, motivation and theory *S. I. A. M. Review*, vol. 19, n° 1, p. 46-89.
15. Dixon L. C. W. (1972), The choice of step length, a crucial factor in the performance of variable metric algorithms, in: *Numerical methods for nonlinear optimization* (Lootsma ed.) Academic Press, New York.
16*. Ermoliev Y. M. (1966), Methods for solving nonlinear extremal problems, *Cybernetics* 2, 4, p. 1-17.
17. Fletcher R. (1970), A new approach to variable metric algorithms, *The Computer Journal*, vol. 13, n° 3, p. 317-322.
18. Fletcher R., Reeves C. M. (1964), Function minimization by conjugate gradients, *Computer Journal* 7, p. 149-154.
19. Forsythe G. E. (1968), On the asymptotic directions of the *s*-dimensional optimum gradient method, *Numerische Mathematik* 11, p. 57-76.
20. Gacs P., Lovasz L. (1981), Khachiyan's algorithm for linear programming, *Mathematical Programming Study* 14, p. 61-68.
21. Gill P. E., Murray W. (1972), Quasi-Newton Methods for Unconstrained Optimization, *J. Institute of Math. and Its Appl.* 9, p. 91-108.
22. Goffin J. L. (1977), On convergence rates of subgradient optimization methods, *Mathematical Programming* 13, p. 329-347.
23. Goffin J. L. (1980), The Relaxation Method for solving Systems of Linear Inequalities, *Mathematics of Operations Research* 5, 3, p. 388-414.
24. Goldfarb D. (1970), A Family of variable metric methods derived by variational means, *Mathematics of Computation*, 24, p. 23-26.
25. Goldstein A. A. (1965), On Steepest Descent, *S. I. A. M. J. Control* 3, p. 147-151.
26. Greenstadt J. (1972), A Quasi-Newton Method with no Derivatives, *Mathematics of Computation* 26, p. 145-166.
27. Grinold R. C. (1970), Lagrangian subgradients, *Management Science*, vol. 17, n° 3, p. 185-188.
28*. Gubin L. G., Polyak B. T., Raik E. V. (1967), The Method of Projections for Finding the Common Point of Convex Sets, *USSR Comp. Math. and Math. Phys.* 7, p. 1-24.
29. Held M., Wolfe P., Crowder H. P. (1974). Validation of subgradient optimization, *Mathematical Programming* 6, p. 62-88.
30. Hestenes M. R., Stiefel E. (1952). Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bureau of Standards* 49, p. 409—436.
31*. Himmelblau D. M. (1972), *Applied nonlinear programming*, McGraw-Hill, 498 p.
32. Huang H. Y., Chamblis J. P. (1973), Quadratically Convergent Algorithms and One-Dimensional Search Schemes, *J. Optimization Theory and Appl.* 11, p. 175-188.
33. Huard P. (1967), Resolution of Mathematical Programming with Nonlinear Constraints by the Method of Centers, *in*: *Nonlinear Programming*, (J. Abadie ed.), North Holland, Amsterdam, p. 206-219.
34. Huard P. (1970), A Method of Centers with Upper-Bounding Functions with Applications, *in*: *Nonlinear Programming*, (J. B. Rosen, O. L. Mangasarian, K. Ritter eds.), Academic Press, New York.
35. Kawamura K., Volz R. A. (1973), On the Rate of Convergence of the Conjugate Gradient Reset Methods with Inaccurate Linear Minimizations, *I. E. E. E. Trans. Automatic Control* 18, p. 360-366.

36*. Khachian L. G. (1979), "A polynomial algorithm in linear programming" Soviet Math. Dokl. 20, 1, p. 191-194.
37. Klessig R., Polak E. (1972), Efficient Implementation of the Polak-Ribière Conjugate Gradient Algorithm, *S. I. A. M. J. Control*, 10, p. 524-549.
38. Kowalik J., Osborne M. R. (1968), *Methods for Unconstrained Optimization Problems*, Elsevier, New York.
39*. Lasdon L. S. (1970), *Optimization theory for large systems*, Macmillan, New York, 523 p.
40. Lemarechal C. (1974), Méthodes de sous-gradients, *Bulletin de la Direction des Études et Recherches E. D. F.*, Série C, n° 2, p. 5-14.
41. Lemarechal C. (1975), An extension of Davidon methods to non-differentiable problems, *Mathematical Programming Study* 3, p. 95-109.
42. Lemarechal C. (1978), Nonsmooth optimization and descent methods, International Institute for Applied Systems Analysis, *Research Report*, 78-4, *Laxenburg (Autriche)*.
43. Lemarechal C. (1980*a*), Nondifferentiable optimization, in: *Nonlinear Optimization, Theory and Algorithms*, (Dixon, Spedicato, Szegö eds.), Birkhäuser, Boston.
44. Lemarechal C. (1980*b*), Extensions diverses des méthodes de gradient et applications, *Thèse Doctorat ès Sc., Université Paris IX.*
45. Lemarechal C., Mifflin R. (1978), *Nonsmooth optimization*, IIASA Proceedings Series. Pergamon Press.
46. Lemarechal C., Pironneau O. (1978), *Nonlinear optimization methods and application to some control problems in physics*, Publication of the Instituto de Matematica, Universidade Federal de Rio de Janeiro, (UFRJ), Brésil.
47. Lenard M. L. (1975), Practical convergence condition for the Davidon-Fletcher-Powell method, *Mathematical Programming* 9, p. 69-86.
48. Lenard M. L. (1976), Convergence conditions for restarted conjugate gradient methods with inaccurate line searches, *Mathematical Programming* 10, n° 1, p. 32-51.
49*. Levin A. J. (1965), On an Algorithm for the Minimization of Convex Functions, *Soviet Math. Dokl.* 6, 1, p. 286-290.
50. Luenberger D. G. (1973), *Introduction to linear and nonlinear programming*, Addison Wesley.
51. McCormick G. P., Ritter K. (1972), Methods of conjugate directions versus quasi-Newton methods, *Mathematical Programming*, 3, 1, p.101-116.
52. McCormick G. P., Ritter K. (1974), Alternative Proofs of the Convergence Properties of the Conjugate Gradient Method, *J. Optimization Theory and Appl.* 13, p. 497-515.
53. Mifflin R. (1975), A superlinearly convergent algorithm for minimization without evaluating derivatives, *Mathematical Programming* 9, p. 100-117.
54. Minoux M. (1982), Un algorithme de sous-gradient pour la recherche d'un point satisfaisant les conditions d'optimalité en optimisation non différentiable, *XIth International Symposium on Mathematical Programming*, Bonn, 23-27 août 1982.
55. Motzkin T., Schoenberg I. J. (1954), The Relaxation Method for Linear Inequalities, *Canad. J. Math.*, 6, p. 393-404.
56. Murray W. (1972), *Numerical Methods for Unconstrained Optimization*, Academic Press, New York.
57. Murtagh B. A., Sargent R. W. H. (1970), Computational Experience with Quadratically Convergent Minimization Methods, *Computer Journal* 13, p. 185-194.
58. Newman D. J. (1965), Location of the Maximum on Unimodal Surfaces, *Journal A. C. M.* 12, 3, p. 395-398.
59. Ortega J. M., Rheinboldt W. C. (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.
60. Pierre D. A. (1969), *Optimization theory with applications*, Johs Wiley, New York.
61*. Polak E. (1971), *Computational methods in optimization, a unified approach*, Academic Press, New York.

62. Polak E. (1974), A Modified Secant Method for Unconstrained Minimization, *Mathematical Programming* 6, p. 264-280.
63*. Polyak B. T. (1966), A general method for solving extremum problems, *Soviet Mathematics* n° 8, p. 593-597.
64*. Polyak B. T. (1978), Subgradient methods: a survey of Soviet Research, in: *Nonsmooth Optimization*, (Lemarechal et Mifflin eds.), Pergamon Press, p. 5-30.
65. Powell M. J. D. (1964), An efficient method for finding the minimum of a function of several variables without calculating derivatives, *The Computer Journal*, 7, p. 155-162.
66. Powell M. J. D. (1971), Recent advances in unconstrained optimization, *Mathematical Programming* 1, p. 26-57.
67. Powell M. J. D. (1976), Some global convergence properties of a variable metric algorithm for minimization without exact line searches. in: *Nonlinear Programming*, S. I. A. M., A. M. S. Proceedings, vol. IX (R. W. Cottle, C. E. Lemke eds.), New York, p. 53-72.
68*. Rockafellar R. T. (1970), *Convex analysis*, Princeton University Press, Princeton, New Jersey.
69. Rockafellar R. T. (1981), *The Theory of Subgradients and its Applications to Problems of Optimization. Convex and Nonconvex Functions*, Heldermann-Verlag, Berlin.
70. Rosenbrock H. H. (1960), An Automatic Method for Finding the Greatest or Least Value of a Function, *The Computer Journal* 3, p. 175-184.
71. Shanno D. F. (1979), Conditioning of quasi-newton methods for function minimization, *Mathematics of Computation* 24, p. 641-656.
72*. Shor N. Z. (1968), On the rate of convergence of the generalized gradient method, *Kibernetika* 4, 3, p. 98-99.
73*. Shor N. Z. (1970), Convergence rate of the gradient descent method with dilatation of the space, *Cybernetics* 6, 2, p. 102-108.
74*. Shor N. Z. (1975), Convergence of a gradient method with space dilatation in the direction of the difference between two successive gradients, *Kibernetika* 11, 4, p. 48-53.
75*. Shor N. Z. (1977), Cut-off Methods with Space Extension in Convex Programming Problems, *Kibernetika* 13, 1, p. 94-95, traduit dans *Cybernetics* 13, 1, p. 94-96.
76*. Shor N. Z., Zhurbenko N. G. (1971), A minimization method using the operation of extension of the space in the direction of the difference of two successive gradients, *Cybernetics* 7, 3, p. 450-459.
77*. Shor N. Z., Shabashova L. P. (1972), Solution of minimax problems by the generalized gradient method with space dilatation, *Cybernetics* 8, 1, p. 88-94.
78*. Skokov V. A. (1974), Note on minimization methods using space dilatation, *Cybernetics* 10, 4, p. 689-692.
79. Vignes J. (1978), New methods for evaluating the validity of the results of mathematical computations, *Mathematics and Computers in Simulation*, vol. 20, n° 4, p. 227-249.
80. Vignes J., Alt R., Pichat M. (1980), *Algorithmes Numériques. Analyse et mise en oeuver*, Editions TECHNIP, Publications de l'Institut Français du Pétrole.
81. Wolfe P. (1975), A method of conjugate subgradients for minimizing nondifferentiable functions, in: *Nondifferentiable optimization*, Mathematical Programming Study 3, p. 145-173.
82. Zangwill W. I. (1967), Minimizing a function without calculating derivatives, *The Computer Journal* 10, p. 293-296.
83*. Zangwill W. I. (1969), *Nonlinear Programming: A Unified Approach*, Prentice Hall, New York.

ГЛАВА 5
НЕЛИНЕЙНАЯ ОПТИМИЗАЦИЯ С ОГРАНИЧЕНИЯМИ

Часть 1. ПРЯМЫЕ МЕТОДЫ (ИЛИ МЕТОДЫ РЕШЕНИЯ ИСХОДНОЙ ЗАДАЧИ)

§ 1. Необходимые условия оптимальности

Нас будет интересовать задача

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \quad i \in I = \{1, 2, \ldots, m\}, \qquad (P')$$
$$x \in \mathbf{R}^n,$$

которая есть не что иное, как задача (P) из гл. 1 с условием $S = \mathbf{R}^n$. Все функции f и g_i $(i \in I)$ предполагаются непрерывными и дифференцируемыми. Обозначим через X множество решений задачи (P'), т. е.

$$X = \{x \in \mathbf{R}^n \mid g_i(x) \leqslant 0 \quad \forall i \in I\}.$$

1.1. Допустимые направления и выделение ограничений. Будем предполагать, что множество X непусто; однако оно вполне может иметь пустую внутренность, в частности, если требуется, чтобы некоторые ограничения содержали условие равенства (п. 1.4). Утверждение, что $x^0 \in X$ есть локальный оптимум задачи (P'), влечет, что $f(x)$ не может убывать, когда x описывает дугу кривой Γ (достаточно регулярной), выходящую из x^0 и содержащуюся в множестве решений X. Такая дуга кривой Γ будет называться допустимой и будет определяться посредством непрерывно дифференцируемой функции φ: $\mathbf{R}^+ \to \mathbf{R}^n$ параметра $\theta \geqslant 0$

$$\varphi(0) = [\varphi_1(\theta), \varphi_2(\theta), \ldots, \varphi_n(\theta)],$$

удовлетворяющей условиям

a) $\varphi(0) = x^0$;

b) $\varphi(\theta) \in X$ для достаточно малого $\theta > 0$.

Допустимым направлением в точке x^0 назовем любой вектор

$$y = \frac{d\varphi}{d\theta}(0) = \left[\frac{d\varphi_1}{d\theta}(0), \frac{d\varphi_2}{d\theta}(0), \ldots, \frac{d\varphi_n}{d\theta}(0)\right]^T,$$

касающийся дуги кривой $\varphi(\theta)$, допустимой в x^0.

В дальнейшем будем обозначать через C_{ad} конус, образованный множеством допустимых направлений в точке x^0.

Прежде всего отыщем условие, необходимое для того, чтобы вектор $y \in \mathbf{R}^n$ удовлетворял условию $y \in C_{ad}$.

Обозначим через I^0 множество индексов *насыщенных* ограничений в x^0, т. е. ограничений, выполняющихся в x^0 в форме равенства:

$$I^0 = \{i \in I \mid g_i(x^0) = 0\}.$$

Рассмотрим, кроме того, конус G вида

$$G = \{y \mid \nabla g_i^T(x^0)\, y \leqslant 0 \quad \forall i \in I^0\}.$$

Тогда справедлива

Лемма 1. *Пусть y — допустимое направление в x^0. Тогда необходимо выполняются соотношения*

$$\nabla g_i^T(x^0)\, y \leqslant 0 \quad \forall i \in I^0, \tag{1}$$

иначе говоря, $C_{ad} \subset G$.

Доказательство. Пусть $\varphi(\theta)$ — дуга допустимой кривой в x^0, $y = \frac{d\varphi}{d\theta}(0)$ — допустимое направление в x^0. Поскольку $g_i(x^0) < 0$ $(i \notin I^0)$, то для достаточно малого $\theta > 0$ всегда будет выполняться $g_i(\varphi(\theta)) < 0$. Напротив, для $i \in I^0$ при достаточно малом $\theta > 0$ должно выполняться $g_i(\varphi(\theta)) < 0$.

При помощи разложения Тейлора функции $g_i(\varphi(\theta))$ в окрестности точки $\theta = 0$ это условие записывается в виде

$$g_i(x^0) + 0\nabla g_i^T(x^0)\frac{d\varphi}{d\theta}(0) + 0o(\theta) \leqslant 0,$$

где $o(0) \to 0$ при $\theta \to 0$. Значит, в силу равенства $g_i(x^0) = 0$, необходимо (условие необходимое, но не достаточное), чтобы направление $y = \frac{d\varphi}{d\theta}(0)$ удовлетворяло условию

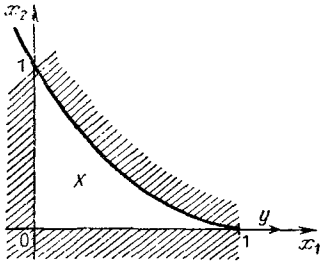


Рис. 1. В точке $x^0 = (1, 0)$ направление $y = (1, 0)$, принадлежащее конусу G, не является допустимым

$$\nabla g_i^T(x^0)\, y \leqslant 0 \quad \forall i \in I^0.$$

Итак,

$$y \in C_{ad} \Rightarrow y \in G,$$

и лемма доказана.

Основной источник трудностей протекает из того, что не все направления y, удовлетворяющие соотношениям (1), обязательно будут допустимыми направлениями, как будет ясно из приводимого ниже примера.

Рассмотрим в $\mathbf{R}^2$ множество X, определяемое ограничениями

$$g_1(x) = -x_1 \leqslant 0,$$
$$g_2(x) = -x_2 \leqslant 0,$$
$$g_3(x) = -(1 - x_1)^3 + x_2 \leqslant 0$$

(см. рис. 1).

В точке $x^0(x_1^0 = 1,\ x_2^0 = 0)$ ограничениями, выполняющимися в виде равенства, будут второе и третье. Значит, $I^0 = \{2, 3\}$.

С другой стороны,

$$\nabla g_2(x^0) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \nabla g_3(x^0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

В этом случае условия (1) запишутся для $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ в виде

$$-y_2 \leqslant 0,$$
$$y_2 \leqslant 0.$$

Вектор $y = (1, 0)$ удовлетворяет этим неравенствам. Однако это направление не будет допустимым, ибо выполнено условие $x^0 + \theta y \notin X \ \forall \theta > 0$. Чтобы исключить такие ситуации, следует наложить дополнительные условия на множество X (т. е. на множество ограничений $g_i(x)$).

Говорят, что область X, определяемая ограничениями $g_i(x) \leqslant 0$ $(i \in I)$, удовлетворяет в точке $x^0 \in X$ *условию выделения ограничений* [34, 1], если

$$\mathrm{cl}(C_{\mathrm{ad}}) = G. \qquad (QC)$$

Таким образом, постановка условия выделения ограничений в x^0 сводится к предположению, что условия (1) необходимы и достаточны для того, чтобы направление y было допустимым в x^0 (или пределом сходящейся последовательности допустимых направлений в x^0).

Очевидно, что прямая проверка условия (QC) может на практике оказаться трудной, поэтому отыскиваются достаточные условия реализуемости соотношения (QC). Наиболее важные результаты содержит приводимая ниже

Лемма 2. *Для того чтобы соотношение (QC) выполнялось в каждой точке $x \in X$, достаточно, чтобы выполнялось одно из условий* a) *или* b):

a) *все функции g_i линейны* [32];

b) *все функции g_i выпуклы и множество X имеет непустую внутренность* [51];

Для того чтобы соотношение (QC) выполнялось в точке $x^0 \in X$, достаточно, чтобы имело место условие:

c) *градиенты $\nabla g_i(x^0)$ $(i \in I^0)$ ограничений в x^0, выполняющихся в форме равенства, линейно независимы* [18].

Доказательство. Прямая проверка п. a) не представляет труда. Для установления b) и c) рассмотрим конус

$$R = \{y \mid \nabla g_i^T(x^0)\, y < 0 \quad \forall i \in I^0\};$$

он называется *конусом возможных направлений* в x^0 (п. 3.2). Легко видеть, что $R \subset C_{\mathrm{ad}}$, откуда следует, что $\mathrm{cl}(R) \subset \mathrm{cl}(C_{\mathrm{ad}})$.

Покажем, что каждое из условий b) или c) влечет равенство $\mathrm{cl}(R) = G$, и, значит, в силу леммы 1 имеем $\mathrm{cl}(C_{\mathrm{ad}}) = G$, а это есть не что иное, как условие выделения ограничений.

Прежде всего заметим, что достаточным условием равенства $\mathrm{cl}(R) = G$ будет $R \neq \varnothing$. В самом деле, допустим, что найдется $\bar{y}$, для которого

$$\nabla g_i^T(x^0)\,\bar{y} < 0 \qquad \forall i \in I^0,$$

и пусть $y \in G$ — произвольный элемент, т. е. удовлетворяет соотношениям

$$\nabla g_i^T(x^0)\,y \leqslant 0 \qquad \forall i \in I^0.$$

Тогда для любого $\lambda \in [0, 1]$

$$\lambda y + (1-\lambda)\bar{y} \in R.$$

Устремив λ к 1 снизу, получим последовательность направлений, содержащихся в $\mathbf{R}$ и имеющих пределом y. Отсюда получаем $\mathrm{cl}(R) = G$.

Предположим теперь, что выполнено условие b). Поскольку множество X имеет непустую внутренность, то найдется такой элемент $\bar{x}$, что

$$g_i(\bar{x}) < 0 \qquad \forall i \in I.$$

Тогда, используя свойство выпуклости функций g_i, для любого $x^0 \in X$ можем написать

$$0 > g_i(\bar{x}) \geqslant g_i(x^0) + \nabla g_i^T(x^0)\,(\bar{x} - x^0),$$

значит, $\bar{y} = (\bar{x} - x^0) \in R$ и, стало быть, $R \neq \varnothing$.

Наконец, предположим, что выполнено условие c). Тогда не найдется таких ненулевых λ_i $(i \in I^0)$, что

$$\sum_{i \in I^0} \lambda_i \nabla g_i(x^0) = 0.$$

Отсюда при помощи теоремы Гордана (см. приложение 1) выводим, что найдется такой элемент $y \in \mathbf{R}^n$, что

$$\nabla g_i^T(x^0)\,y < 0 \qquad \forall i \in I^0,$$

и, значит, $R \neq \varnothing$.

Замечание 1. Условие b) леммы 2 можно расширить, потребовав только, чтобы функции g_i были псевдовыпуклыми (дифференцируемая на $\mathbf{R}^n$ функция называется псевдовыпуклой, если $f(y) \geqslant f(x)$ для любых x, y таких, что $\nabla f^T(x)\,(y-x) \geqslant 0$).

Замечание 2. Интересно отметить, что различные условия леммы 2 допускают возможность комбинирования. Например, выделение ограничений может быть реализовано в любой точке $x \in X$, если функции g_i либо линейны, либо нелинейны и выпуклы и если существует такой элемент $\bar{x} \in X$, что $g_i(\bar{x}) < 0$ (для нелинейной выпуклой функции g_i).

Или еще, выделение ограничений будет реализовано в $x^0 \in X$, если функции g_i либо линейны, либо нелинейны и векторы $\nabla g_i(x^0)$ (для нелинейной g_i) линейно независимы.

1.2. Необходимые условия Куна — Таккера. Приводимая ниже теорема является основополагающей: она дает, при условии выделения ограничений, необходимое условие локальной оптимальности для экстремальной задачи с ограничениями типа (P').

Теорема 1 [34]. *Предположим, что функции f, g_i $(i \in I)$ непрерывно дифференцируемы и выполнено условие выделения ограничений в $x^0 \in X$, где*

$$X = \{x \in \mathbf{R}^n \mid g_i(x) \leqslant 0 \quad \forall i \in I\}.$$

Тогда, для того чтобы точка x^0 была локальным оптимумом задачи (P'), необходимо, чтобы существовали такие числа $\lambda_i \geqslant 0$ $(i \in I)$, что

$$\begin{gathered} \nabla f(x^0) + \sum_{i \in I} \lambda_i \nabla g(x^0) = 0, \\ \lambda_i g_i(x^0) = 0 \qquad \forall i \in I. \end{gathered} \tag{KT}$$

(Числа λ_i называются множителями Куна — Таккера.)

Доказательство. Для того чтобы точка x^0 была локальным оптимумом задачи (P'), необходимо, чтобы для любой дуги допустимой кривой φ выполнялось неравенство

$$f(\varphi(\theta)) \geqslant f(\varphi(0)) = f(x^0)$$

при любом достаточно малом $\theta > 0$. Но

$$f(\varphi(0)) = f(x^0) + 0 \nabla f^T(x^0) y + 0o(\theta),$$

где $o(0) \to 0$ при $0 \to 0$, и $y = \dfrac{d\varphi}{d\theta}(0) \in C_{\text{ad}}$.

Отсюда получаем необходимое условие:

$$\nabla f^T(x^0) y \geqslant 0 \qquad \forall y \in C_{\text{ad}}.$$

Из этого условия следует, что должно выполняться также неравенство

$$\nabla f^T(x^0) y \geqslant 0 \qquad \forall y \in \operatorname{cl}(C_{\text{ad}})$$

(по непрерывности, при рассмотрении последовательности допустимых направлений y^k, сходящейся к y).

По условию выделения ограничений в x^0 имеем $\operatorname{cl}(C_{\text{ad}}) = G$, откуда получаем необходимое условие того, что x^0 — локальный минимум задачи (P'):

$$\nabla f^T(x^0) y \geqslant 0$$

для любого y, удовлетворяющего неравенству

$$\nabla g_i^T(x^0) y \leqslant 0 \qquad \forall i \in I^0.$$

Теперь напомним теорему Фаркаша — Минковского (см. приложение 1).

Пусть A есть $p \times q$-матрица, b — вектор из $\mathbf{R}^p$. Для того чтобы существовал элемент $x \in \mathbf{R}^q$ $(x \geqslant 0)$, удовлетворяющий уравнению $Ax = b$, необходимо и достаточно, чтобы $u^T b \geqslant 0$ для любого $u \in \mathbf{R}^p$, удовлетворяющего неравенству $u^T A \geqslant 0$.

Если отождествить в приведенной формулировке матрицу A^T с матрицей, столбцами которой являются векторы $-\nabla g_i(x^0)$ $(i \in I^0)$, вектор b — с вектором $\nabla f(x^0)$, вектор x^T — с вектором $\lambda = (\lambda_i)$ $(i \in I^0)$, а вектор u^T — с вектором y, то можно сформулировать теорему иначе:

Для того чтобы $\nabla f^T(x^0) y \geqslant 0$ для любого y, удовлетворяющего условию

$$[-\nabla g_i^T(x^0)] y \geqslant 0, \qquad i \in I^0,$$

необходимо и достаточно, чтобы существовали такие числа $\lambda_i \geqslant 0$ $(i \in I^0)$, что

$$\nabla f(x^0) = \sum_{i \in I^0} \lambda_i (-\nabla g_i(x^0)).$$

Выбрав $\lambda_i = 0$ для $i \in I - I^0$, что выражается условиями $\lambda_i g_i(x^0) = 0$, получаем условия Куна — Таккера в точке x^0.

Замечание. Важно отметить, что в том случае, когда градиенты насыщенных ограничений линейно независимы (тогда x^0 называется регулярной точкой), вектор Куна — Таккера λ определен однозначно.

1.3. Геометрическая интерпретация условий Куна — Таккера. Геометрически условия Куна — Таккера интерпретируются при помощи рис. 2. При условии выделения ограничений множество

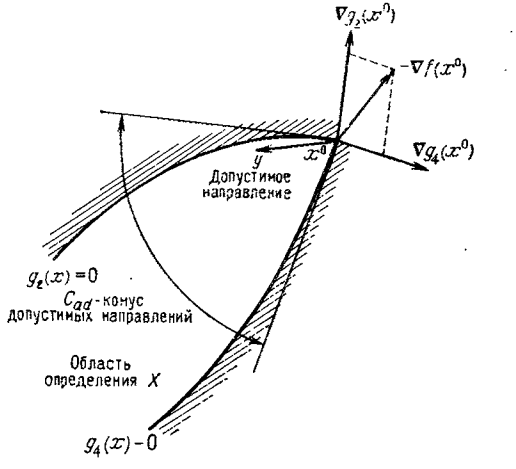


Рис. 2. Иллюстрация условий Куна — Таккера на двумерном примере. В точке x^0 насыщенными ограниченными являются ограничения g_2 и g_4 и $I^0 = \{2, 4\}$.

допустимых направлений y образует конус C_{ad} — пересечение $|I^0|$ полупространств

$$\nabla g_i^T(x^0) y \leqslant 0 \quad \forall i \in I^0.$$

Для того чтобы точка x^0 была локальным оптимумом, необходимо, чтобы вектор $-\nabla f(x^0)$ составлял тупой угол с каждым допустимым направлением y.

При этих условиях можно геометрическими средствами убедиться, что вектор $-\nabla f(x^0)$ должен в действительности выражаться в виде линейной комбинации векторов $\nabla g_i(x^0)$ $(i \in I^0)$ с положительными коэффициентами λ_i.

1.4. Расширение на задачи с ограничениями типа равенств и неравенств. Условия Лагранжа. Условия Куна — Таккера без труда распространяются на задачи, содержащие ограничения одновременно как типа равенств, так и типа неравенств:

$$\begin{aligned} & f(x) \to \min, \\ & g_i(x) \leqslant 0, \quad i \in I = \{1, 2, \ldots, m\}, \\ & h_l(x) = 0, \quad l \in L = \{1, 2, \ldots, p\}, \\ & x \in \mathbf{R}^n. \end{aligned} \tag{P_1'}$$

Все функции f, g_i $(i \in I)$, h_l $(l \in L)$ предполагаются непрерывно дифференцируемыми.

Если обозначить через X множество решений задачи (P_1')

$$X = \{x \mid g_i(x) \leqslant 0 \ (i \in I), \quad h_l(x) = 0 \ (l \in L)\},$$

то понятия дуги допустимой кривой и допустимого направления в точке $x^0 \in X$ будут определяться в точности так же, как в п. 1.1.

Обозначив снова через C_{ad} конус допустимых направлений в x^0, доказываем, что необходимое условие принадлежности $y \in C_{\text{ad}}$ имеет вид

$$y \in G_1 = \{y \mid y \in \mathbf{R}^n; \nabla g_i^T(x^0)\, y \leqslant 0 \, (i \in I^0); \ \nabla h_l^T(x^0)\, y = 0 \, (l \in L)\},$$

где

$$I^0 = \{i \mid g_i(x^0) = 0\}.$$

Сформулированное условие не является достаточным, поэтому мы снова вводим условие выделения ограничений:

$$\operatorname{cl}(C_{\text{ad}}) = G_1, \tag{QC}$$

которое сводится к предположению, что $y \in G_1$ есть необходимое и достаточное условие для того, чтобы направление y было допустимым (или пределом сходящейся последовательности допустимых направлений).

Теперь может быть доказана

Теорема 1′ [34]. *Предположим, что функции f, g_i $(i \in I)$, h_i $(l \in L)$ непрерывно дифференцируемы и что условие (QC) выполнено в точке x^0 — решении задачи (P_1').*

Тогда для того чтобы точка x^0 была локальным оптимумом задачи (P_1'), необходимо, чтобы существовали такие числа $\lambda_i \geqslant 0$ $(i \in I)$ и μ_l $(l \in L)$ (μ_l произвольного знака), что

$$\nabla f(x^0) + \sum_{i \in I} \lambda_i \nabla g_i(x^0) + \sum_{l \in L} \mu_l \nabla h_l(x^0) = 0, \tag{KT}$$

где

$$\lambda_i g_i(x^0) = 0 \quad \forall i \in I.$$

Доказательство на каждом шаге аналогично доказательству теоремы 1.

Основные достаточные условия выполнения условия (QC) в этом случае состоят в следующем.

а) Функции g_i $(i \in I)$ выпуклы, функции h_l $(l \in L)$ линейны, и существует такой элемент $\bar{x} \in X$, что $g_i(\bar{x}) < 0$ $\forall i \in I$ и $h_l(\bar{x}) = 0$ $\forall l \in L$.

b) В точке $x^0 \in X$ градиенты $\nabla g_i(x^0)$ $\forall i \in I^0$ и $\nabla h_l(x^0)$ $\forall l \in L$ линейно независимы.

В случае задачи с ограничениями только типа равенства

$$f(x) \to \min,$$
$$h_l(x) = 0, \quad l \in L = \{1, 2, \ldots, p\},$$
$$x \in \mathbf{R}^n$$

и при условии выделения ограничений в x^0 (линейная независимость градиентов $\nabla h_l(x^0)$ $(l \in L)$) необходимым условием, чтобы точка x^0 была локальным оптимумом, является существование таких чисел μ_l $(l \in L)$ произвольного знака, что

$$\nabla f(x^0) + \sum_{l \in L} \mu_l \nabla h_l(x^0) = 0.$$

Тем самым мы вновь пришли к классическим условиям Лагранжа.

§ 2. Достаточные условия оптимальности. Седловые точки и функция Лагранжа

Теперь мы изучим достаточные условия оптимальности для задач типа

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \quad i \in I, \qquad (P)$$
$$x \in S \subset \mathbf{R}^n.$$

Заметим, что когда $S = \mathbf{R}^n$, то мы вновь приходим к задаче (P') из § 1. Тем не менее все, что будет говориться в этом параграфе, применимо и в более общем случае, к задачам типа (P). Множество S может, например, быть множеством точек с целочисленными коэффициентами (целочисленное программирование см. в гл. 7.).

Поставим в соответствие каждому i-му ограничению $(i \in I)$ действительное число $\lambda_i \geqslant 0$, называемое *множителем Лагранжа*. Функция Лагранжа, отвечающая задаче (P), есть по определению функция

$$L(x, \lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x).$$

Определение 1. Пусть $\bar{x} \in S$, $\bar{\lambda} \geqslant 0$. Говорят, что $(\bar{x}, \bar{\lambda})$ есть седловая точка функции $L(x, \lambda)$, если

$$L(\bar{x}, \bar{\lambda}) \leqslant L(x, \bar{\lambda}) \quad \forall x \in S, \qquad (2)$$
$$L(\bar{x}, \lambda) \leqslant L(\bar{x}, \bar{\lambda}) \quad \forall \lambda \geqslant 0. \qquad (3)$$

Пример седловой точки для функции двух переменных приведен на рис. 3.

2.1. Теорема 2 (характеристическое свойство седловых точек). *Пусть* $\bar{x} \in S$, $\bar{\lambda} \geqslant 0$; *точка* $(\bar{x}, \bar{\lambda})$ *является седловой для функции* $L(x, \lambda)$ *в том и только том случае, если*

$$\bar{L}(\bar{x}, \bar{\lambda}) = \min_{x \in S} L(x, \bar{\lambda}), \tag{a}$$

$$g_i(\bar{x}) \leqslant 0 \quad \forall i \in I, \tag{b}$$

$$\bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i \in I. \tag{c}$$

Доказательство. 1) Если $(\bar{x}, \bar{\lambda})$ — седловая точка, то a) выполняется. С другой стороны,

$$L(\bar{x}, \bar{\lambda}) \geqslant L(\bar{x}, \lambda),$$

откуда

$$f(\bar{x}) + \sum_I \bar{\lambda}_i g_i(\bar{x}) \geqslant f(\bar{x}) + \sum_I \lambda_i g_i(\bar{x}),$$

и мы получаем

$$\sum_{i \in I} (\lambda_i - \bar{\lambda}_i) g_i(\bar{x}) \leqslant 0 \quad \forall \lambda \geqslant 0. \tag{4}$$

Если условие b) не выполняется для некоторого индекса i, то

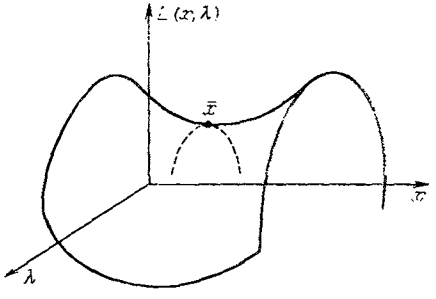


Рис. 3. Иллюстрация понятия седловой точки: $\bar{x}$ минимизирует $L(x, \bar{\lambda})$ на S; $\bar{\lambda}$ максимизирует $L(\bar{x}, \lambda)$ на $\mathbf{R}^+$

всегда можно выбрать достаточно большое $\lambda_i > 0$, чтобы не выполнялось соотношение (4). Значит, b) выполнено.

Наконец, для $\lambda = 0$

$$(4) \Rightarrow \sum_I \bar{\lambda}_i g_i(\bar{x}) \geqslant 0;$$

но $\bar{\lambda}_i \geqslant 0$ и $g_i(\bar{x}) \leqslant 0 \Rightarrow \sum_I \bar{\lambda}_i g_i(\bar{x}) \leqslant 0$. Стало быть,

$$\sum_I \bar{\lambda}_i g_i(\bar{x}) = 0 \Rightarrow \bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i \in I,$$

и c) выполнено.

2) Предположим, что условия a), b), c) выполнены.

$$(a) \Rightarrow L(\bar{x}, \bar{\lambda}) \leqslant L(x, \bar{\lambda}) \quad \forall x \in S;$$

с другой стороны, $c) \Rightarrow L(\bar{x}, \bar{\lambda}) = f(\bar{x})$.

Наконец,

$$L(\bar{x}, \lambda) = f(\bar{x}) + \sum_I \lambda_i g_i(\bar{x}) \leqslant f(\bar{x}) = L(\bar{x}, \bar{\lambda}) \quad \forall \lambda \geqslant 0.$$

Следовательно,

$$L(\bar{x}, \lambda) \leqslant L(\bar{x}, \bar{\lambda}) \leqslant L(x, \bar{\lambda}) \quad \forall \lambda \geqslant 0, \quad \forall x \in S,$$

что и завершает доказательство.

2.2. Теорема 3 (достаточность условия седловой точки). *Если* $(\bar{x}, \bar{\lambda})$ *есть седловая точка функции* $L(x, \lambda)$, *то* $\bar{x}$ *есть глобальный оптимум задачи* (P).

Доказательство. Из условия а) предыдущей теоремы следует, что

$$f(\bar{x}) + \sum_I \bar{\lambda}_i g_i(\bar{x}) \leqslant f(x) + \sum_I \bar{\lambda}_i g_i(x) \quad \forall x \in S.$$

С другой стороны,

$$(c) \Rightarrow \bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i,$$

откуда

$$f(\bar{x}) \leqslant f(x) + \sum_I \bar{\lambda}_i g_i(x) \quad \forall x \in S,$$

а поскольку $\bar{\lambda} \geqslant 0$, то $f(\bar{x}) \leqslant f(x)$ для любого $x \in S$, удовлетворяющего условию $g_i(x) \leqslant 0$ $\forall i \in I$, откуда и следует теорема.

Этот результат, весьма общий, применим к любым выпуклым и невыпуклым экстремальным задачам, с дифференцируемыми или недифференцируемыми функциями, непрерывным или дискретным множеством S, конечным или счетным, и т. п.

Однако для некоторых задач может не существовать седловой точки. Это, вообще говоря, встречается в невыпуклых задачах. В гл. 6 мы покажем, как можно расширить класс задач, допускающих седловые точки, введя обобщенные функции Лагранжа.

2.3. **Пример (невыпуклая экстремальная задача).** Рассмотрим задачу с одной переменной x:

$$f(x) = -x^2 \to \min,$$
$$2x - 1 \leqslant 0,$$
$$0 \leqslant x \leqslant 1.$$

Примем в качестве S множество $\{x | 0 \leqslant x \leqslant 1\}$, в качестве ограничений — неравенство $2x - 1 \leqslant 0$, в качестве переменной Лагранжа — $\lambda \geqslant 0$; тогда функция Лагранжа будет иметь вид

$$L(x, \lambda) = -x^2 + \lambda(2x - 1).$$

Поскольку функция $L(x, \lambda)$ вогнута в x (при любом фиксированном λ), то ее минимум в $x \in [0, 1]$ достигается либо в $x = 0$, либо в $x = 1$. Минимум $x^* = 1/2$ не достигается ни при каком значении λ, поэтому седловая точка не существует.

2.4. Необходимое и достаточное условие существования седловой точки. Функция возмущения. Погружая задачу (P) в более широкое семейство задач, называемых возмущенными задачами, мы покажем, что при такой постановке можно получить необходимое и достаточное условие существования седловой точки, допускающее очень интересную геометрическую интерпретацию.

Пусть $y=(y_1, y_2, \ldots, y_m)^T$ — m-вектор, каждая компонента которого соответствует ограничению $g_i(x)\leqslant 0$ задачи (P). Рассмотрим для $y\in \mathbf{R}^n$ семейство (возмущенных) задач

$$\begin{aligned} &f(x)\to \min (=\Phi(y)),\\ &g_i(x)\leqslant y_i \quad \forall i\in I, \qquad (P_y)\\ &x\in S. \end{aligned}$$

Ясно, что (P_y) сводится к (P) при помощи модификации (возмущения) правой части в ограничениях.

Для $y\in \mathbf{R}^m$ оптимальным значением для возмущенных задач (P_y) будет функция $\Phi(y)$, называемая *функцией возмущений*. При $y=0$ вновь приходим к исходной задаче и получаем $\Phi(0)=\min(P)$.

Важно отметить свойство монотонности функции возмущений:

$$y'\leqslant y \Rightarrow \Phi(y')\geqslant \Phi(y),$$

следующее непосредственно из определения (множество решений задачи $(P_{y'})$ включается в множество решений задачи (P_y)).

Теперь может быть сформулирована

Теорема 4. *Если задача (P) имеет конечный оптимум $\bar{x}$, то значение $\bar{\lambda}$ будет множителем седловой точки в том и только том случае, если гиперплоскость*

$$z=\Phi(0)-\bar{\lambda}y$$

будет опорной в точке $y=0$ графика функции возмущений $\Phi(y)$, т. е. в том и только том случае, если

$$\Phi(y)\geqslant \Phi(0)-\bar{\lambda}y \qquad \forall y\in \mathbf{R}^n. \tag{5}$$

Доказательство. Необходимость. В самом деле, рассмотрим такое $\bar{\lambda}$, что $(\bar{x}, \bar{\lambda})$ есть седловая точка. По определению $\bar{\lambda}\geqslant 0$ и

$$f(x)+\bar{\lambda}g(x)\geqslant f(\bar{x})+\bar{\lambda}g(\bar{x}) \qquad \forall x\in S.$$

Но, с другой стороны, $\bar{\lambda}g(\bar{x})=0$, поэтому

$$f(x)+\bar{\lambda}g(x)\geqslant f(\bar{x})=\Phi(0)=\min(P);$$

стало быть,

$$f(x)\geqslant \Phi(0)-\bar{\lambda}g(x) \qquad \forall x\in S.$$

В частности, для любого y, при котором задача (P_y) имеет решение, и для любого решения x этой задачи

$$f(x)\geqslant \Phi(0)-\bar{\lambda}g(x)\geqslant \Phi(0)-\bar{\lambda}y.$$

Отсюда следует неравенство

$$\Phi(y) \geqslant \Phi(0) - \bar{\lambda} y.$$

Если y таково, что (P_y) не имеет решения, то $\Phi(y) = +\infty$, и соотношение (5) снова выполнено.

Достаточность. Предположим, что

$$\Phi(y) \geqslant \Phi(0) - \bar{\lambda} y \qquad \forall y \in \mathbf{R}^m.$$

Докажем сначала, что $\bar{\lambda} \geqslant 0$. Допустив, что $\bar{\lambda}_i < 0$, и устремив $y_i \to +\infty$ (когда другие компоненты нули), получим

$$\Phi(0) - \bar{\lambda} y \to +\infty,$$

откуда следует противоречие со свойством монотонности:

$$\Phi(y) \leqslant \Phi(0) \qquad \forall y \geqslant 0.$$

Значит, $\bar{\lambda} \geqslant 0$.

По определению функции $\Phi(y)$

$$f(x) \geqslant \Phi(g(x)) \qquad \forall x \in S.$$

Стало быть,

$$f(x) \geqslant \Phi(g(x)) \geqslant \Phi(0) - \bar{\lambda} g(x) \qquad \forall x \in S,$$

откуда следует, что

$$f(x) + \bar{\lambda} g(x) \geqslant f(\bar{x}) \quad (= \Phi(0)) \qquad \forall x \in S.$$

В частности, для $x = \bar{x}$ получаем

$$f(\bar{x}) + \bar{\lambda} g(\bar{x}) \geqslant f(\bar{x}),$$

откуда $\bar{\lambda} g(\bar{x}) \geqslant 0$. А поскольку также $\bar{\lambda} g(\bar{x}) \leqslant 0$ (ибо $\bar{\lambda} \geqslant 0$, $g(x) \leqslant \leqslant 0$), то $\bar{\lambda} g(\bar{x}) = 0$. Тогда соотношение

$$f(x) + \bar{\lambda} g(x) \geqslant f(\bar{x}) = f(\bar{x}) + \bar{\lambda} g(\bar{x}) \qquad \forall x \in S$$

приводит к равенству

$$L(\bar{x}, \bar{\lambda}) = \min_{x \in S} L(x, \bar{\lambda}),$$

которое с учетом того, что $\bar{\lambda} \geqslant 0$, $\bar{\lambda} g(\bar{x}) = 0$, показывает (см. теорему 2), что $(\bar{x}, \bar{\lambda})$ есть седловая точка. Следовательно, $\bar{\lambda}$ есть множитель седловой точки.

Доказанная теорема без труда распространяется на случай, когда задача (P) содержит ограничения типа равенства, что приводит к исключению условий неотрицательности на соответствующие переменные λ.

Теперь мы приведем геометрическую иллюстрацию теоремы 4. Прежде всего рассмотрим следующий пример (который будет воспроизведен в гл. 6, п. 2.4).

Пример. Рассмотрим задачу

$$x_1^2 + x_2^2 \to \min,$$
$$2x_1 + x_2 \leqslant -4.$$

Оптимум этой (выпуклой) задачи можно найти либо при помощи условий Куна — Таккера, либо геометрически. Получим $x_1 = -8/5$, $x_2 = -4/5$ (см. рис. 1 гл. 6 п. 2.4). В этой точке значение минимума равно $x_1^2 + x_2^2 = 16/5$.

Возмущенная задача записывается в виде

$$\Phi(y) = \min x_1^2 + x_2^2,$$
$$2x_1 + x_2 + 4 \leqslant y$$

и имеет в качестве решения

$$x_1 = \frac{2y-8}{5}, \quad x_2 = \frac{y-4}{5}, \quad \text{если} \quad y \leqslant 4,$$
$$x_1 = 0, \quad x_2 = 0, \quad \text{если} \quad y \geqslant 4.$$

Отсюда получаем

$$\Phi(y) = 0, \quad \text{если} \quad y \geqslant 4,$$
$$\Phi(y) = \left(\frac{2y-8}{5}\right)^2 + \left(\frac{y-4}{5}\right)^2 = \frac{1}{5}y^2 - \frac{8}{5}y + \frac{16}{5}, \quad \text{если} \quad y \leqslant 4.$$

Эта функция возмущений представлена на рис. 4. Ясно, что речь идет о выпуклой функции, имеющей в точке $y = 0$ касательную с угловым коэффициентом $-8/5 = -\bar{\lambda}$.

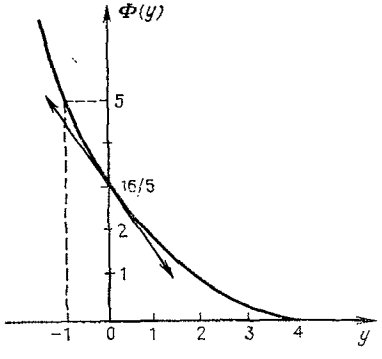


Рис. 4. Пример, иллюстрирующий необходимое и достаточное условие существования седловой точки (теорема 4)

Из теоремы 4 следует существование для этой задачи седловой точки с множителем $\bar{\lambda} = 8/5$.

Этот пример будет изучен снова в гл. 6, п. 2.4 в связи с двойственностью; будет показано, что на самом деле значение $\bar{\lambda} = 8/5$ соответствует оптимуму двойственной функции и может быть получено в результате решения двойственной задачи.

Предыдущий пример иллюстрировал ситуацию, когда решаемая задача выпукла и функция возмущений выпукла. На рис. 5 показаны несколько случаев, которые также могут представиться. Для наглядности предполагаем, что задача содержит единственное ограничение и, таким образом, λ и y — скаляры.

Существуют многочисленные невыпуклые задачи, для которых функция Φ не выпукла и для которых, вообще говоря, не существует опорная плоскость, а значит, и седловая точка (рис. 5*а*).

Однако в некоторых невыпуклых случаях (как дифференцируемых, так и недифференцируемых) функция $\Phi(y)$ может иметь опорную гиперплоскость в точке $(0, \Phi(0))$, откуда будет следовать существование седловой точки (рис. 5*b*, 5*c*). Эти случаи трудны для рассмотрения.

Наконец, в выпуклом случае (как дифференцируемом, так и недифференцируемом) при некоторых условиях (см. п. 2.5 ниже)

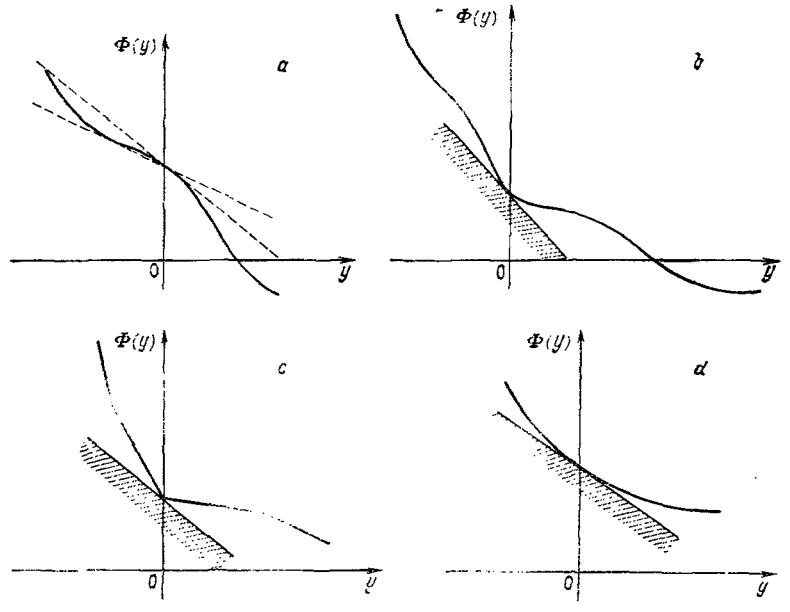


Рис. 5. *a*) Седловой точки нет. *b*) Функция Φ не выпукла, но седловая точка существует. *c*) Функция Φ не выпукла и не дифференцируема, но седловая точка существует. *d*) Функция Φ выпукла и седловая точка существует

функция $\Phi(y)$ имеет опорную гиперплоскость в точке $(0, \Phi(0))$, и существование седловой точки обеспечивается автоматически (рис. 5*d*).

2.5. Существование седловой точки в выпуклом случае. Множители Лагранжа как субградиенты функции возмущения. Как мы только что видели из геометрических рассмотрений, выпуклость функции возмущений Φ является, вообще говоря, достаточным условием существования седловой точки. Более точно, имеет место

Теорема 5. *Пусть функции* f, g_i $(i \in I)$ *выпуклы, множество* $S \subset \mathbf{R}^n$ *выпукло и существует такое* $x \in S$, *что*

$$g_i(x) < 0 \quad \forall i \in I.$$

Тогда если задача (P) *имеет оптимальное решение* $\bar{x}$, *то существует такой вектор множителей* $\bar{\lambda} \geqslant 0$, *что* $(x, \bar{\lambda})$ *есть седловая точка функции Лагранжа* $L(x, \lambda)$.

Доказательство. Классическое доказательство этой теоремы [32] использует теорему об отделимости выпуклых множеств (см. приложение 1, теорема 3) и имеет то неудобство, что не дает достаточных возможностей для интуитивного восприятия. Однако оно представлено в приложении 2.

В действительности можно привести прямое доказательство, исходя из теоремы 4 и заметив, что из ее условий следует, что:

— $\Phi(y)$ — выпуклая собственная функция;

— $y=0$ принадлежит внутренности $\mathrm{dom}(\Phi)$.

Тогда отсюда вытекает, что $\Phi(y)$ имеет субградиент γ в точке $y=0$ (см. гл. 1, теорема 6), удовлетворяющий соотношению

$$\Phi(y)\geqslant\Phi(0)+\gamma y \qquad \forall y\in\mathbb{R}^m.$$

Положив $\bar{\lambda}=-\gamma$, выводим существование седловой точки.

Выполнение сформулированных выше условий проверить легко.

Выпуклость функции Φ очевидна.

Если задача (P) имеет оптимальное решение $\bar{x}$ с конечным значением $f(\bar{x})=\Phi(0)$, то отсюда следует, что $0\in\mathrm{dom}(\Phi)$.

С другой стороны, существует такое $x\in S$, что $g_i(\hat{x})<0$ $\forall i\in I$, поэтому найдется такое $\varepsilon>0$, что

$$\max_{i\in I}|y_i|<\varepsilon\Rightarrow g_i(\hat{x})\leqslant y_i \qquad \forall i\in I.$$

Следовательно, для достаточно малого $\|y\|$ множество решений задачи (P_y) непусто, значит, $y\in\mathrm{dom}(\Phi)$, и отсюда следует, что $0\in\mathrm{int}\,\mathrm{dom}(\Phi)$.

Наконец, чтобы показать, что функция собственная, заметим, что для любого $y\in\mathrm{dom}(\Phi)$ и $y\leqslant 0$

$$-\infty<\Phi(y)\geqslant\Phi(0)>-\infty.$$

В частности, для y', удовлетворяющих условию

$$y'_i=-\varepsilon/2 \qquad \forall i\in I,$$

имеем

$$+\infty>\Phi(y')\geqslant\Phi(0)>-\infty.$$

Пусть y'' — точка с коэффициентами $y''_i=+M$ $\forall i\in I$, где M — большая константа, $M>0$.

Поскольку 0 принадлежит внутренности сегмента $[y', y'']$, то соотношение $\Phi(y'')=-\infty$ выполняться не может (ибо в этом случае в силу выпуклости $\Phi(0)=-\infty$ и мы приходим к противоречию).

Значит,

$$\Phi(y'')>-\infty.$$

Для любой точки $y\in\mathrm{dom}(\Phi)$ положим $M=\max\{y_i;\ 0\}$ и определим y'' посредством $y''_i=M$ $\forall i\in I$; тогда $\Phi(y)\geqslant\Phi(y'')>-\infty$, откуда получаем требуемое.

Приведенное доказательство интересно тем, что оно выявляет основное свойство множителей Лагранжа в седловой точке:

Свойство 1. *В выпуклом случае* $\bar{\lambda}$ *есть множитель седловой точки в том и только том случае, если* $-\bar{\lambda}$ *есть субградиент функции возмущений* Φ *в точке* $y=0$.

Мы вернемся к этой интерпретации множителей седловой точки как субградиентов функции возмущения в п. 2.8 при постоптимальном анализе.

2.6. Связь с условиями Куна — Таккера в дифференцируемом выпуклом случае. Если функции f и g_i выпуклы и дифференцируемы и если $S=\mathbf{R}^n$, то предыдущий результат может быть связан с условиями Куна — Таккера следующим образом:

Теорема 6 (необходимость и достаточность условий Куна — Таккера в выпуклом случае).

Рассмотрим случай $S=\mathbf{R}^n$ (*задача* (P') § 1). *Если при условиях теоремы* 5 *функции* f *и* g_i *дифференцируемы, то, для того чтобы точка* $\bar{x}$ *была глобальным оптимумом задачи* (P'), *необходимо и достаточно, чтобы в точке* $\bar{x}$ *выполнялись условия Куна — Таккера, т. е.:*

существует такое $\bar{\lambda}\geqslant 0$, *что*

$$\nabla_x L(x,\bar{\lambda})=0,$$

$$\bar{\lambda}g_i(x)=0 \quad \forall i\in I.$$

Доказательство. В силу теоремы 5 точка $\bar{x}$ есть глобальный оптимум в том и только том случае, если $\bar{\lambda}\geqslant 0$ таково, что $(\bar{x},\bar{\lambda})$ есть седловая точка, т. е.

a) $x\to \min L(x,\bar{\lambda})$ на $\mathbf{R}^n$;

b) $g_i(\bar{x})\leqslant 0$;

c) $\bar{\lambda}_i g_i(\bar{x})=0 \ \forall i\in I$.

Функции f и g_i $(i\in I)$ выпуклы и дифференцируемы, поэтому функция $L(x,\bar{\lambda})$ выпукла и дифференцируема по x, и, значит, условие a) равносильно равенству

$$\nabla_x L(\bar{x},\bar{\lambda})=0,$$

которое в комбинации с b) и c) дает условия Куна — Таккера в $\bar{x}$.

Итак, множители Куна — Таккера отождествляются с множителями Лагранжа в седловой точке.

2.7. Достаточные условия локальной оптимальности в невыпуклом случае. Очевидно, существует много задач, к которым предыдущий результат не может быть применен, ибо условие выпуклости является довольно обременительным. Тогда, как интересно заметить, для получения достаточного условия локальной оптимальности можно довольствоваться условием локальной выпуклости.

Таким образом, если функции $f(x)$ и $g_i(x)$ дифференцируемы, то для того чтобы точка $\bar{x}$ была локальным оптимумом задачи (P''), достаточно, чтобы:

a) функции $f(x)$ и $g_i(x)$ были выпуклы в некоторой окрестности точки $\bar{x}$;

b) в точке x были выполнены условия Куна — Таккера.

В общем случае, когда функции $f(x)$ и $g_i(x)$ произвольны (не обязательно дифференцируемы), для того чтобы точка $\bar{x}$ была локальным оптимумом задачи (P), достаточно, чтобы:

a) существовала окрестность $\mathscr{V}(\bar{x})$ точки $\bar{x}$, в которой функции $f(x)$ и $g_i(x)$ выпуклы;

b) существовало такое $\bar{\lambda}$, что $(\bar{x}, \bar{\lambda})$ есть седловая точка, при дополнительном ограничении принадлежности $\mathscr{V}(\bar{x})$, т. е. точка, удовлетворяющая ограничениям

$$L(\bar{x}, \bar{\lambda}) \leqslant L(x, \bar{\lambda}) \qquad \forall x \in S \cap \mathscr{V}(x),$$
$$L(\bar{x}, \lambda) \leqslant L(\bar{x}, \bar{\lambda}) \qquad \forall \lambda \geqslant 0.$$

2.8. Множители Лагранжа и постоптимальный анализ. Рассмотрим задачу (P):

$$\begin{aligned} f(x) &\to \min, \\ g(x) &\leqslant 0, \\ x &\in S \subset \mathbf{R}^n. \end{aligned} \qquad (P)$$

Постоптимальным анализом задачи (P) будем называть исследование вариаций оптимального значения задачи при малых возмущениях правых частей ограничений.

Иными словами, речь идет об исследовании функции возмущений $\Phi(y)$ в окрестности точки $y = 0$.

Предположим, что функция $\Phi(y)$ дифференцируема в $y = 0$ и что задача (P) имеет седловую точку $(\bar{x}, \bar{\lambda})$. В этом случае опорная гиперплоскость к графику функции $\Phi(y)$ в точке $(0, \Phi(0))$ есть касательная гиперплоскость, уравнение которой имеет вид

$$z = \Phi(0) + [\nabla_y \Phi(0)](y - 0).$$

Тогда $\bar{\lambda} = -[\nabla_y \Phi(0)]^T$, или, иначе,

$$\bar{\lambda}_i = -\frac{\partial \Phi(0)}{\partial y_i}, \qquad i = 1, \ldots, m.$$

Итак, множитель $\bar{\lambda}_i$ седловой точки интерпретируется (с точностью до знака) как частная производная оптимального значения задачи (P) относительно возмущения y_i правой части i-го ограничения.

Можно придать этому следующую экономическую интерпретацию. Рассмотрим, например, исходную экономическую задачу, в которой речь шла о максимизации доходов предприятия при различных ограничениях на ресурсы (бюджетных, на первичные материалы и т. д.). Множитель $\bar{\lambda}_i$, соответствующий ограничению, относящемуся к i-му ресурсу, представляет собой дополнительный доход от дополнительной i-й единицы ресурса, которым располагает предприятие.

§ 3. Оптимизация с ограничениями. Прямые методы (решение исходной задачи)

Большинство существующих методов в нелинейном программировании можно разделить на два больших семейства:

— прямые методы (методы непосредственного решения исходной задачи);

— методы, использующие понятие двойственности.

Прямые методы характеризуются тем, что они имеют дело непосредственно с заданной задачей (называемой исходной задачей в противоположность двойственной задаче, которая будет определена в следующей главе). Эти методы порождают последовательность решений (т. е. точек, удовлетворяющих ограничениям), обеспечивая монотонное убывание минимизируемой функции. Стало быть, они обладают важным свойством: если итерационный процесс прерывается, эти методы обеспечивают приближенное решение, удовлетворяющее ограничениям. И напротив, они, вообще говоря, неудобны тем, что связаны с некоторыми тонкостями и что бывает затруднительно получить свойство глобальной сходимости (яркий пример см. в п. 3.2 ниже). В противоположность этому двойственные методы — более сильные, и глобальную сходимость часто бывает легче получить; зато они представляют то неудобство, что они не дают решения исходной задачи в ходе решения — оно реализуемо лишь в конце сходимости.

Здесь мы ограничимся изучением лишь первого семейства, отсылая читателя для изучения другого семейства к следующей главе.

3.1. Метод замены переменных. Иногда можно исключить ограничения типа неравенств при помощи специально подобранных преобразований. Например, если ограничение имеет вид

$$a \leqslant x \leqslant b \qquad \text{(типа границ)},$$

то преобразование $x = a + (b-a)\sin^2 y$ позволяет заменить x на неограниченную переменную y.

Можно придумать много подобных преобразований.

Точно так же часто можно, по крайней мере теоретически, исключить ограничения типа равенства:

$$g_i(x) = 0.$$

Когда выполняется теорема о неявных функциях, то можно выразить одну переменную (например, x_j) через другие:

$$x_j = \psi_i(x_1, x_2, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n).$$

Однако в практическом плане этот метод применим лишь в случае *линейных ограничений*; это происходит из-за трудности определения обратных функций ψ_i.

В случае линейных ограничений, когда переменные — произвольного знака, метод позволяет сразу заменить задачу на экстремальную задачу без ограничений; тогда этот метод можно особенно рекомендовать.

3.2. Методы возможных направлений. Идея состоит в применении принципа методов оптимизации без ограничений, чтобы учесть требования, диктуемые наличием ограничений.

Более точно, выбираем исходную точку, удовлетворяющую ограничениям, и отыскиваем возможное направление перемещения y, т. е. такое, что:

a) малое перемещение в этом направлении не выводит из множества X решений;

b) функция $f(x)$ в этом направлении строго убывает.

Затем производим перемещение в полученном направлении на некоторое расстояние до получения новой точки, лучшей, чем предыдущая; например, отыскиваем минимум функции $f(x)$ в направлении y с ограничением не выходить из множества X.

Для анализа метода такого типа мы обратимся к результатам гл. 3, п. 3.3, показывающим, что сходимость зависит, весьма решительным образом, от выбора направления перемещения на каждом этапе. Яркой иллюстрацией этому будет служить метод Зойтендейка [65], который будет сейчас изложен и в котором будет показано, почему алгоритм не сходится в его примитивной форме и какие изменения необходимо произвести, чтобы обеспечить его сходимость.

Рассмотрим задачу

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \qquad i = 1, 2, \ldots, m, \qquad (P')$$
$$x \in \mathbf{R}^n,$$

и пусть X есть область решений:

$$X = \{x \mid g_i(x) \leqslant 0,\ i = 1, \ldots, m\}.$$

Будем исходить из точки $x^0 \in X$. Задача состоит в выборе направления перемещения y, которое является *возможным* (удовлетворяющим a) и b)).

Пусть

$$I^0 = \{i \in I \mid g_i(x^0) = 0\}$$

есть множество индексов насыщенных ограничений в x^0.

Вектор y должен быть таким, чтобы малое перемещение в этом направлении не делало никакое насыщенное ограничение положительным, т. е.

$$\left[\frac{d}{d\theta}\, g_i(x^0 + \theta y)\right]_{\theta=0} = \nabla g_i^T(x^0)\, y \leqslant 0, \qquad i \in I^0.$$

Кроме того, малое перемещение вдоль y должно вести к уменьшению функции f, т. е.

$$\left[\frac{d}{d\theta}\, f(x^0 + \theta y)\right]_{\theta=0} = \nabla f^T(x^0)\, y < 0.$$

Отсюда следует идея определить направление перемещения, отыскивая единичный вектор y, обращающий в минимум выражение

$\nabla f^T(x^0)y$ при ограничениях

$$\nabla g_i^T(x^0)y \leqslant 0 \qquad \forall i \in I^0.$$

Это приводит к решению задачи

$$\nabla f^T(x^0)y \to \min,$$
$$\nabla g_i^T(x^0)y \leqslant 0, \qquad i \in I^0, \qquad (I)$$
$$y^Ty = 1.$$

Заменяя ограничение $y^Ty = 1$ (которое нелинейно) другим условием нормализации типа

$$\sum_{i=1}^{n} |y_i| = 1,$$

можно свести все к решению задачи линейного программирования:

$$\nabla f^T(x^0)y \to \min,$$
$$\nabla g_i^T(x^0)y \leqslant 0, \qquad i \in I^0, \qquad (I')$$
$$\sum |y_i| = 1$$

(но задача (I') не эквивалентна задаче (I)).

Этот выбор, обладая преимуществом простоты, тем не менее может привести к серьезным затруднениям.

Прежде всего, в случае нелинейных ограничений направление y, определенное задачей (I) или (I'), может быть таким, чтобы бесконечно малое перемещение в этом направлении приводило к немедленному выходу из множества решений.

На рис. 6 представлено в пространстве двух переменных x_1 и x_2 нелинейное ограничение $g_i(x) \leqslant 0$, насыщенное в x^0.

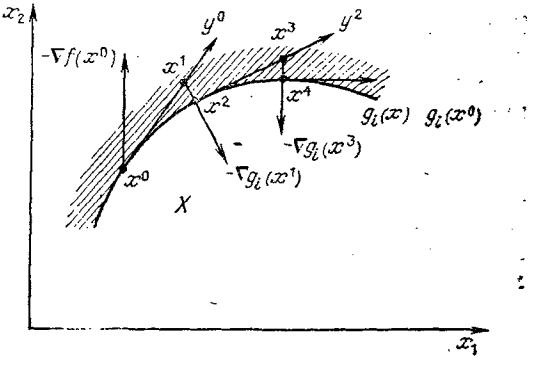


Рис. 6. В методе Зойтендейка малое перемещение в направлении y (расположенном здесь в касательной плоскости к ограничению в точке x^0) выводит точку из множества решений X

Исходящий из x^0 вектор y, минимизирующий $\nabla f^T(y^0)y$, есть проекция $-\nabla f(x^0)$ на плоскость, касающуюся ограничения в x^0.

Из-за кривизны границы (с уравнением $g_i(x) = g_i(x^0)$) видим, что всякое, даже очень малое, перемещение вдоль этого вектора y выводит точку из области X; тогда для возвращения в X после каж-

дого перемещения необходимо осуществить специальную процедуру, и это может повлечь несходимость метода.

Другая, более основательная трудность заключается в том, что отображение D (которое ставит в соответствие точке x направление перемещения) не замкнуто.

Это проистекает из того, что в (I) участвуют лишь ограничения, насыщенные в x^0. Следовательно, как только в процессе перемещения новое ограничение становится насыщенным, направление y может испытать резкий разрыв. Пример на рис. 7 ясно показывает это. Область X определяется неравенствами

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_1 + x_2 \leqslant 1;$$

имеем

$$-\nabla f = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \text{const}.$$

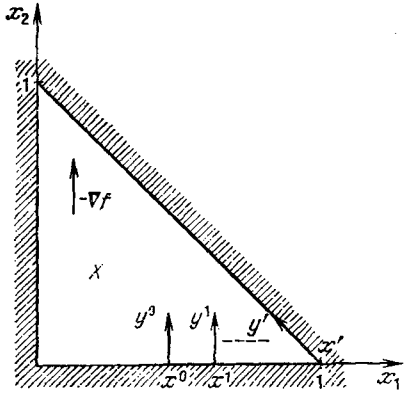


Рис. 7. Пример, показывающий, что отображение D: $x \to y$ не замкнуто

Для внутренних точек отрезка $[0, 1]$ оси x_1 единственным насыщенным ограничением будет ограничение $x_2 \geqslant 0$, и оптимальным решением задачи (I) будет направление $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, параллельное $-\nabla f$. В точке $x' = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ограничение $x_1 + x_2 \leqslant 1$ также становится насыщенным, и оптимальным решением задачи (I) будет направление

$$y' = \begin{bmatrix} -\sqrt{2}/2 \\ +\sqrt{2}/2 \end{bmatrix}.$$

Это объясняет, почему метод Зойтендейка, в его исходной форме, может не сходиться.

Чтобы избавиться от этих неудобств, воспользуемся идеей поиска на каждом этапе направления перемещения, учитывающего все ограничения задачи, а не только насыщенные ограничения.

Предположим, что для достаточно малого $\|y\|$ (для определенности $\|y\|^2 < \alpha$) функция f и ограничения g_i $(i \in I)$ могут быть аппроксимированы их разложением Тейлора первого порядка. Поиск направления перемещения, учитывающего множество ограничений, может производиться при помощи решения задачи

$$\begin{gathered} \nabla f^T(x^0)\, y \to \min, \\ g_i(x^0) + \nabla g_i^T(x^0)\, y \leqslant 0 \quad \forall i \in I, \\ y^T y \leqslant \alpha. \end{gathered} \tag{II}$$

Из-за ограничения нормализации эта задача нелинейна. С другой стороны, если для насыщенного ограничения $i \in I^0$ решение задачи

(*II*) удовлетворяет условию

$$\nabla g_i^T(x^0)\, y = 0,$$

то ясно, что можно найти задачу того же типа, что и упомянутая выше (когда любое малое перемещение выводит точку из множества решений). Поэтому было предложено [55] заменить задачу (*II*) (очень близкой) задачей линейного программирования

$$\begin{gathered} \xi \to \min, \\ \nabla f^T(x^0)\, y - \xi \leqslant 0, \\ g_i(x^0) + \nabla g_i^T(x^0)\, y - u_i \xi \leqslant 0 \quad \forall i \in I, \end{gathered} \tag{III}$$

в которой неизвестными являются y_i $(i \in I)$, ξ, а u_i $(i \in I)$ — параметры, которые фиксированы и положительны. Это — *метод Топкиса и Вейнотта.*

Задача (*III*) снова состоит в минимизации критерия спуска $\nabla f^T(x^0)\, y$, но с требованием на этот раз, чтобы в направлении y удаляться от границы области X, по крайней мере локально. В самом деле, если оптимум задачи (*III*) соответствует некоторому значению $\xi^* < 0$, то ясно, что $\nabla f^T(x^0)\, y < 0$ и, значит, y есть направление спуска.

С другой стороны, из соотношения

$$g_i(x^0) + \nabla g_i^T(x^0)\, y \leqslant u_i \xi^* < 0$$

выводим, что все ограничения будут оставаться выполненными для достаточно малых перемещений в направлении y.

Параметры u_i используются в качестве коэффициентов нормализации: они теоретически должны быть выбраны в виде функции кривизны функций g_i в x^0 (а стало быть, соответственных значений гессианов в x^0). Практически при отсутствии этой информации часто выбирают $u_i = 1$ $(\forall i \in I)$.

Поскольку в (*III*) входят все ограничения (даже те, которые не являются насыщенными), то найденное направление не испытывает резкого разрыва, когда ограничение перестает быть активным или когда становится активным новое ограничение (в противовес этому решение задачи (*III*) является более дорогостоящим, поскольку содержит дополнительные ограничения).

Теперь можно сформулировать следующий результат.

Лемма 3. *Предположим, что:*

— *функции f и g_i непрерывно дифференцируемы;*

— *множество решений*

$$X = \{x \mid g_i(x) \leqslant 0,\ i \in I\}$$

выпукло и имеет непустую внутренность, т. е.

$$(\exists x:\ g_i(x) < 0\ \forall i);$$

— *все точки $x \in X$ регулярны (для всех насыщенных ограничений в x градиенты $\nabla g_i(x)$ линейно независимы).*

Тогда отображение D, которое каждой точке $x^0 \in X$ ставит в соответствие направление перемещения y — решение задачи (III),—

замкнуто и порождает равномерно возможные направления (см. гл. 3, § 3).

Доказательство см. в [35].

Из леммы 3 и теоремы 3 гл. 3 выводим, что произведение отображений $U_x \circ D$, где U_x соответствует одномерной оптимизации с ограничением:

$$\min_{\substack{\alpha \geqslant 0 \\ x^0 + \alpha y \in X}} \{f(x^0 + \alpha y)\},$$

замкнуто.

Чтобы применить теорему 10 гл. 1 и доказать тем самым глобальную сходимость метода Тонкиса и Вейнотта, достаточно, следовательно, показать, что f есть функция спуска. Поскольку y есть направление спуска для f до тех пор, пока для оптимума задачи (*III*) выполняется $\xi^* < 0$, достаточно убедиться в том, что если $\xi^* = 0$, то текущая точка будет локальным оптимумом. Справедливость этого утверждения показывает

Теорема 7. *Если в некоторой точке $x^0 \in X$ оптимальное значение задачи (III) удовлетворяет условию $\xi^* = 0$ и если x^0 есть регулярная точка множества X, то в x^0 выполняются условия Куна — Таккера.*

Доказательство. Предположим, что $\xi^* = 0$. Связывая с ограничениями задачи (*III*) двойственные переменные $\mu_0 \geqslant 0$, $\mu_i \geqslant 0$ $(i \in I)$, запишем задачу, двойственную задаче (*III*):

$$\begin{gathered} \sum_{i \in I} \mu_i g_i(x^0) \to \max, \\ -\mu_0 \nabla f^T(x^0) - \sum_{i \in I} \mu_i \nabla g_i^T(x^0) = 0, \\ \mu_0 + \sum_{i \in I} \mu_i u_i = 1 \\ \mu_0 \geqslant 0, \quad \mu_i \geqslant 0 \end{gathered} \tag{IV}$$

(где ограничения являются ограничениями типа равенства, ибо в (*III*) переменные y и ξ — произвольного знака).

Согласно теореме двойственности в линейном программировании имеем

$$\xi^* = \sum_{i \in I} \mu_i g_i(x^0) = 0.$$

Поскольку $\mu_i \geqslant 0$ и $g_i(x^0) \leqslant 0$ $\forall i \in I$, то $\mu_i g_i(x^0) = 0$ $\forall i \in I$; это показывает, что лишь те μ_i, которые соответствуют ограничениям, насыщенным в x^0, могут быть отличны от нуля.

Если x^0 — регулярная точка, т. е. если $\nabla g_i(x^0)$ линейно независимы для $i \in I^0 = \{i | g_i(x^0) = 0\}$, то

$$\mu_0 = 0 \Rightarrow \sum_{i \in I^0} \mu_i \nabla g_i(x^0) = 0 \Rightarrow \mu_i = 0 \qquad \forall i \in I^0 \Rightarrow \mu_i = 0 \quad \forall i \in I_1$$

что противоречит второму ограничению задачи (*IV*).

Следовательно, $\mu_0 \neq 0$.

Положив теперь $\lambda_i = \mu_i/\mu_0 \geqslant 0$, приходим к тому, что $\lambda_i \geqslant 0$ удовлетворяют равенствам

$$\nabla f(x^0) + \sum_{i \in I} \lambda_i \nabla g_i(x^0) = 0,$$

$$\lambda_i g_i(x^0) = 0 \qquad \forall i \in I.$$

Следовательно, при обычных условиях регулярности алгоритм Зойтендейка оканчивается, когда $\xi^* = 0$, при помощи выделения точки, удовлетворяющей условиям Куна — Таккера.

3.3. Метод проекционного градиента [48]. *Вполне естественная идея применения методов оптимизации без ограничений к задачам с ограничениями состоит в проектировании перемещения на каждом этапе на границу области, с тем чтобы убедиться в том, что вновь полученная точка принадлежит множеству возможных решений.*

Модифицируя таким способом, по возможности незначительно, исходный метод, можно надеяться сохранить его эффективность. Были предложены многочисленные алгоритмы, основанные на этой общей схеме.

Например, для градиентного метода приходим к проектированию градиента на границу области. Это дает продвижение вдоль границы в направлении «относительно» самого сильного спуска, т. е. требуемого ограничениями; один из первых алгоритмов, построенных по этому принципу, есть проекционный градиент [48].

Как мы покажем, метод проекционного градиента особенно интересен в случае линейных ограничений.

Итак, пусть задача поставлена в виде

$$f(x) \to \min,$$

$$a_i x \leqslant b_i, \qquad i \in I_1,$$

$$a_i x = b_i, \qquad i \in I_2.$$

(Заметим, что если имеются ограничения типа положительности на переменные, то они рассматриваются как включенные в ограничения типа I_1.)

Предположим, что нам известно решение x (такая точка всегда может быть получена при помощи решения задачи линейного программирования). Пусть

$$I^0(x) = \{i \mid i \in I_1;\ a_i x = b_i\} \cup I_2$$

есть множество индексов ограничений, насыщенных в точке x. Найдем в x направление перемещения d ($\|d\| = 1$), которое позволяет уменьшить максимально возможно $f(x)$ (и, значит, делает $\nabla f^T(x) d$ минимальным), но которое, по крайней мере для малых перемещений, позволяет оставаться в множестве реализуемых решений. Это приводит к тому, что направление d должно удовлетворять соотношениям

$$a_i d = 0 \qquad \forall i \in I^0(x). \tag{6}$$

Обозначим через A^0 подматрицу матрицы A, состоящую из строк $i \in I^0(x)$ матрицы A. Введем условие, что A^0 есть матрица полного ранга, т. е. $(A^0) = |I^0(x)| = q$, что сводится к предположению об отсутствии вырожденности в точке x^0. Ясно, что множество векторов d, удовлетворяющих соотношениям (6), составляет векторное подпространство размерности $n - q$, определенное следующим образом: $S^0 = \{y | A^0 y = 0\}$.

Стало быть, речь идет об отыскании такого $d \in S^0$, что $\nabla f^T(x) d$ минимально, с ограничением нормализации $\|d\| = 1$.

Приводимый ниже результат дает явное решение этой задачи, получаемое проектированием вектора $-\nabla f(x)$ на подпространство S^0.

Теорема 8. *Если A^0 есть $q \times n$-матрица полного ранга $q \leqslant n$, то оптимальное решение задачи*

$$\nabla f^T(x) d \to \min,$$
$$A^0 d = 0,$$
$$\|d\| = 1,$$

есть $d = \bar{y}/\|\bar{y}\|$, где $\bar{y}$ — проекция вектора $-\nabla f(x)$ на $S^0 = \{y | A^0 y = 0\}$, задаваемая формулой

$$\bar{y} = -P^0 \nabla f(x) = -(I - A^{0T}[A^0 A^{0T}]^{-1} A^0) \nabla f(x)$$

(P^0 называется матрицей проектирования на S^0).

Доказательство. Любой вектор из $\mathbf{R}^n$, и в частности $-\nabla f(x)$, может быть записан в виде

$$-\nabla f(x) = \bar{y} + \bar{z}, \quad \bar{y} \in S^0, \quad z \in S^{0\perp},$$

где $S^{0\perp}$ — ортогональное дополнение к S^0 в $\mathbf{R}^n$. Значит (учитывая, что $\bar{z} \in S^{0\perp}$, $d \in S^0$),

$$-\nabla f^T(x) d = \bar{y}^T d + \bar{z}^T d = \bar{y}^T d.$$

Поскольку d — произвольный единичный вектор из S^0 и $\bar{y} \in S^0$, то величина $\nabla f^T(x) d$ будет минимальной, когда будет максимальной величина $y^T d$, т. е. когда $d = \bar{y}/\|y\|$. Значит, искомое направление d параллельно $\bar{y}$ — проекции вектора $-\nabla f(x)$ на S^0.

Для нахождения $\bar{y}$ заметим, что $S^{0\perp}$ есть векторное подпространство пространства $\mathbf{R}^n$, порожденное столбцами матрицы A^{0T}; тогда $\bar{z}$ можно взять в виде $\bar{z} = A^{0T} u$, где u — вектор из $\mathbf{R}^l$, который требуется найти. Записывая

$$-\nabla f(x) = \bar{y} + A^{0T} u$$

и используя тот факт, что $\bar{y} \in S^0$ (а значит, что $A^0 \bar{y} = 0$), получаем

$$A^0 \bar{y} = -A^0 \nabla f(x) - A^0 A^{0T} u = 0.$$

Если A^0 — матрица полного ранга (т. е. $\operatorname{rang}(A^0) = q$), то матрица $(A^0 A^{0T})$ обратима, что позволяет найти вектор u:

$$u = -[A^0 A^{0T}]^{-1} A^0 \nabla f(x),$$

а, стало быть, $\bar{y}$ задается в виде

$$\bar{y} = -\nabla f(x) + A^{0T}[A^0 A^{0T}]^{-1} A^0 \nabla f(x) = -P^0 \nabla f(x),$$

где матрица

$$P^0 = I - A^{0T}[A^0 A^{0T}]^{-1} A^0$$

есть матрица проектирования на подпространство S^0.

Заметим, что если $\bar{y} \neq 0$, то $\bar{y}$ есть направление спуска, ибо

$$\nabla f^T(x)\, y = -\bar{y}^T(\bar{y} + \bar{z}) = -\bar{y}^T\bar{y} < 0.$$

Получив направление перемещения $d = \bar{y}/\|\bar{y}\|$, находим требуемое в ограничениях максимальное перемещение, т. е. такое, что

$$\alpha_{\max} = \max\{\alpha \mid \alpha \geqslant 0;\ x + \alpha\bar{y} \in X\},$$

где

$$X = \{x \mid x \in \mathbf{R}^n;\ a_i x \leqslant b_i\ (i \in I_1);\ a_i x = b_i\ (i \in I_2)\}.$$

Тогда следующая точка x' выбирается как точка, минимизирующая $f(x + \alpha\bar{y})$ на отрезке $[0, \alpha_{\max}]$.

В точке x' требуется определить новое множество I^0 насыщенных ограничений и новую матрицу проектирования P^0, позволяющую вычислить новое направление перемещения $\bar{y}' = -P^0\nabla f(x')$, и т. д.

Стало быть, алгоритм действует таким образом, пока $\bar{y} = -P^0\nabla f(x) \neq 0$.

Посмотрим теперь, что происходит, когда $\bar{y} = -P^0\nabla f(x) = 0$ в текущей точке x. В этом случае имеем

$$\nabla f(x) + A^{0T}u = 0, \tag{7}$$

причем

$$u = -[A^0 A^{0T}]^{-1} A^0 \nabla f(x). \tag{8}$$

Поскольку столбцы матрицы A^{0T} являются градиентами насыщенных ограничений в точке x, то ясно, что при $u \geqslant 0$ соотношение (7) выражает не что иное, как условия Куна — Таккера в текущей точке x, а это равносильно утверждению (с привлечением обычных условий), что x есть локальный оптимум задачи.

В случае, когда $\bar{y} = 0$, но вектор u, заданный соотношением (8), имеет строго отрицательные компоненты, текущая точка может не быть оптимальной и необходимо искать другое направление перемещения. Для этого исключим в I^0 одно из t ограничений, для которых $u_i < 0$ (например то, для которого u_i отрицательно с наибольшим модулем). Получим новую матрицу A'^0 и новую матрицу проектирования P'^0, которая позволит найти новое направление перемещения $\bar{y}' = -P'^0\nabla f(x)$.

Легко показать, что направление $\bar{y}'$, полученное таким образом, всегда удовлетворяет условиям:

1) $\bar{y}' \neq 0$;

2) $\bar{y}'$ есть направление спуска для f.

После этого итерации могут производиться с новым направлением $\bar{y}'$.

Таким образом, алгоритм проекционного градиента составляет следующую процедуру.

a) На итерации $k=0$ мы находимся в точке x^0.

b) Текущая k-я итерация — в x^k.

Определить множество $I^0(x^k)$ индексов насыщенных ограничений. Положить $L^0=I^0(x^k)$.

c) Пусть A^0 — матрица, строки которой соответствуют ограничениям $i\in L^0$.

Вычислить матрицу проектирования

$$P^0=I-A^{0T}[A^0A^{0T}]^{-1}A^0.$$

Затем вычислить $y^k=-P^0\nabla f(x^k)$.

Если $y^k=0$, то перейти к e).

d) Если $y^k\neq 0$, то найти $\alpha_{\max}=\max\{\alpha\,|\,x^k+\alpha y^k\in X\}$.

Затем найти такое x^{k+1}, что

$$f(x^{k+1})=\min_{0\leqslant\alpha\leqslant\alpha_{\max}}\{f(x^k+\alpha y^k)\}.$$

Положить $k\leftarrow k+1$ и возвратиться к b).

e) Пусть $u=-[A^0A^{0T}]^{-1}A^0\nabla f(x^k)$.

Если $u\geqslant 0$, то конец: x^k удовлетворяет условиям Куна — Таккера. Иначе пусть u_i — наибольшая по модулю из отрицательных компонент вектора u.

Положить $L^0\leftarrow L^0-\{i\}$ и возвратиться к c).

Метод обладает одним интересным свойством: между k-й и $k+1$-й итерациями множества $I^0(x^k)$ и $I^0(x^{k+1})$ индексов насыщенных ограничений *различаются*, вообще говоря, самое большее одним элементом. Следовательно, матрицы A^0 различаются не более чем одной строкой, и новая матрица проектирования может быть вычислена непосредственно исходя из прежней, а это существенно снижает объем требуемых вычислений на каждой итерации.

Исследуем глобальную сходимость метода. Известно, что сходимость существенно зависит от свойств отображения D, которое точке x^k ставит в соответствие искомое направление y^k: D должно быть замкнутым (гл. 1, § 4) и должно порождать равномерно допустимые направления (гл. 3, § 3).

Но легко видеть, что отображение D не является замкнутым, поскольку направление y^k, вообще говоря, испытывает резкий разрыв, когда новое ограничение становится активным. Стало быть, *глобальная сходимость метода проекционного градиента не гарантируется.*

Тем не менее не представляется возможным указать явно практический случай *отсутствия* сходимости, а это, в соединении с простотой постановок, позволяет рассматривать метод как интересный инструмент, по крайней мере для решения задач нелинейной оптимизации с линейными ограничениями.

В случае нелинейных ограничений метод проекционного градиента может быть обобщен, но, как мы покажем, появляются

трудности, которые делают его использование гораздо более тонким. Рассмотрим задачу типа

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \qquad i \in I_1,$$
$$g_i(x) = 0, \qquad i \in I_2.$$

Пусть x — реализуемая (текущая) точка и $I^0(x)$ — множество индексов ограничений, насыщенных в x. Направление перемещения y в точке x вычисляется здесь как проекция вектора $-\nabla f(x)$ на векторное подпространство S^0, касательное к поверхности, определяемой равенством $g_i(x) = 0$ $(i \in I^0(x))$.

Если обозначить через $g^0(x)$ вектор $[g_{i_1}(x), g_{i_2}(x), \ldots, g_{i_q}(x)]^T$, где $\{i_1, i_2, \ldots, i_q\} = I^0(x)$, а через $\frac{\partial g^0}{\partial x}(x)$ — матрицу $\left[\frac{\partial g_{i_1}}{\partial x}, \frac{\partial g_{i_2}}{\partial x}, \ldots, \frac{\partial g_{i_q}}{\partial x}\right]^T$ (якобиан функции g^0, вычисленный в x), то ясно, что подпространство S^0 определяется посредством множества векторов y, удовлетворяющих условию $[\partial g^0/\partial x]\, y = 0$.

Иными словами, вычисление проекции вектора $-\nabla f(x)$ на S^0 будет производиться в точности так же, как и в линейном случае, при условии замены матрицы A^0 матрицей $[\partial g^0/\partial x]$. Получаем

$$\bar{y} = -P^0 \nabla f(x),$$
$$P^0 = I - \left[\frac{\partial g^0}{\partial x}\right]^T \left(\left[\frac{\partial g^0}{\partial x}\right]\left[\frac{\partial g^0}{\partial x}\right]^T\right)^{-1} \left[\frac{\partial g^0}{\partial x}\right].$$

Итак, вычисление проекции не представляет существенной трудности по сравнению с линейным случаем.

Однако главная трудность проистекает из того, что, вообще говоря, направление перемещения $\bar{y}$ сразу выводит точку из области X решений. Тогда необходимо предусмотреть специальные процедуры, позволяющие вернуться в X, когда малое перемещение уже осуществлено в направлении $\bar{y}$. В этом случае выбор наилучшего перемещения следует из тонкого компромисса между убыванием функции f (что требует больших перемещений) и необходимостью оставаться достаточно близко к границе множества X (что требует малых перемещений). Отсюда следует, что всегда в высшей степени трудно строить на этом принципе общие программы, применимые к произвольным задачам нелинейного программирования.

С такого рода ситуацией мы вновь столкнемся в п. 3.5 при рассмотрении обобщенного метода приведенного градиента.

3.4. Метод приведенного градиента. Другой метод, применимый к случаю линейных ограничений, есть метод приведенного градиента [62]. Хотя на уровне принципов этот метод очень близок методам проекционного градиента, на уровне техническом он существенно отличается от них и, вообще говоря, признан более эффек-

тивным. Это, вне сомнения, объясняется тем фактом, что он составляет прямое расширение симплексного метода.

Задача, которую требуется решить, предполагается взятой в стандартной форме, т. е. (гл. 2) с ограничениями только типа равенства:

$$f(x) \to \min,$$
$$Ax = b,$$
$$x \geqslant 0.$$

Предположим, что A — $m \times n$-матрица, $x \in \mathbf{R}^n$, $b = (b_1, b_2, \ldots, b_m)^T$.

Пусть B есть базис, т. е. регулярная $m \times m$-матрица, извлеченная из A. Положим $A = [B, N]$, $x = [x_B, x_N]$. Ограничения можно записать в виде

$$Bx_B + Nx_N = b,$$

что позволяет выразить базисные переменные как функцию внебазисных переменных:

$$x_B = \bar{b} - \bar{N}x_N, \quad \text{где } \bar{b} = B^{-1}b, \quad \bar{N} = B^{-1}N.$$

В дальнейшем будем предполагать базис B невырожденным, т. е. таким, что $\bar{b} > 0$ (и это условие должно выполняться в течение всей процедуры).

Вариация df функции f для перемещения dx, согласованного с ограничениями, записывается в виде

$$df = \frac{\partial f}{\partial x_B}\, dx_B + \frac{\partial f}{\partial x_N}\, dx_N,$$

$$dx_B = -\bar{N}\, dx_N,$$

откуда следует

$$df = \left(\frac{\partial f}{\partial x_N} - \frac{\partial f}{\partial x_B}\bar{N}\right) dx_N = u_N\, dx_N,$$

где полагаем

$$u_N = \frac{\partial f}{\partial x_N} - \frac{\partial f}{\partial x_B}\bar{N}.$$

По определению вектор u_N называется *приведенным градиентом* относительно базиса B.

Пусть на текущем этапе алгоритма $x = [x_B, x_N]$ — полученное реализуемое решение. Чтобы его улучшить, перейдем к направлению $y = [y_B, y_N]$, где y_N определяется равенствами

$$y_j = 0, \text{ если } u_j > 0 \text{ и } x_j = 0,$$
$$y_j = -u_j \text{ в противном случае}$$

для всех индексов j, соответствующих всебазисным переменным, и

$$y_B = -\bar{N}y_N.$$

Иначе говоря, в пространстве внебазисных переменных (независимых переменных) перемещаются в направлении, противоположном направлению приведенного градиента, и, если это направление

стремится сделать переменную x_j отрицательной, вводят требование $y_j = 0$ (это есть не что иное, как проекция на положительный ортант в пространстве внебазисных переменных).

Если $y_N = 0$, то, как будет показано ниже, условия Куна — Таккера выполняются. Для выпуклой функции f это означает, что оптимум достигается. Для невыпуклой функции f имеем локальный оптимум.

Если $y_N \neq 0$, то находим $\theta \geqslant 0$, минимизирующее функцию $g(\theta) = f(x + \theta y)$. Поскольку ограничения положительности переменных должны всегда выполняться, то должно выполняться неравенство $x_j + \theta y_j \geqslant 0 \ \forall j$, что приводит к соотношениям

$$0 \leqslant \theta_{\max} = \min_{y_j < 0} \left\{ -\frac{x_j}{y_j} \right\}.$$

Таким образом, находим $\bar{\theta}$, минимизирующее функцию $g(\theta)$ на $[0, \theta_{\max}]$ (одномерная минимизация).

Могут представиться два случая.

а) $\bar{\theta} < \theta_{\max}$. В этом случае никакая переменная не уничтожается. Процедура производится, начиная с новой точки $x' = x + \bar{\theta} y$. Снова вычисляем вектор u_N, используя тот же базис B.

в) $\bar{\theta} = \theta_{\max}$. В этом случае одна переменная аннулируется, допустим x_s. Если x_s — внебазисная переменная, то продолжаем процедуру без замены базиса. Если же x_s — базисная переменная, то осуществляем замену базиса: любую внебазисную переменную $x_t > 0$ можно заменить на x_s при условии, что разрешающий элемент $\bar{A}_{st}$ строго положителен. После этого процедура продолжается с этим новым базисом.

В заключение убедимся в том, что условие $y_N = 0$ влечет условие Куна — Таккера (ибо ограничения линейны и условие выделения ограничений выполнено).

Сопоставим ограничениям $Ax = b$ множители Куна — Таккера v (любого знака), а ограничениям $x \geqslant 0$ — множители $w \geqslant 0$; условия Куна — Таккера записываются в виде

$$-\frac{\partial f}{\partial x} + vA + wI = 0, \tag{9}$$

$$wx = 0 \tag{10}$$

(I — единичная $n \times n$-матрица).

Положим

$$v = (\partial f / \partial x_B) B^{-1}, \qquad w = [w_B, \ w_N]$$

при $w_B = 0$ и

$$w_N = \frac{\partial f}{\partial x_N} - \frac{\partial f}{\partial x_B} \bar{N} = u_N;$$

тогда соотношение (9) выполнено.

Заметим, что равенство $y_N = 0$ влечет $u_N \geqslant 0$, откуда $w_N \geqslant 0$ и, значит, $w \geqslant 0$. С другой стороны, поскольку для любого внебазисного j имеем $y_j = 0$, то $u_j > 0 \Rightarrow x_j = 0$, и мы приходим к тому, что условия дополнительности (10) выполнены.

Итак, условия Куна — Таккера выполняются при $y_N = 0$.

Метод приведенного градиента, в его примитивной форме, как он здесь изложен, не сходится глобально. В самом деле, легко видеть, что направление перемещения y может испытывать резкие разрывы, например, когда ограничение перестает быть активным. Тем не менее можно получить глобальную сходимость, модифицируя метод и определяя направление перемещения y равенствами

$$y_j = -u_j \quad \text{при} \quad u_j < 0,$$
$$y_j = -x_j u_j \quad \text{при} \quad u_j > 0$$

для всех индексов j, соответствующих внебазисным переменным (см., например, [35]).

3.5. Обобщенный приведенный градиент. Метод обобщенного приведенного градиента [3, 4] содержит комбинацию идей методов линеаризации и методов приведенного градиента. Долгое время метод рассматривался как один из наиболее эффективных для исследования общего случая, когда как оптимизируемая функция, так и ограничения нелинейны (однако при этом делалось предположение об их непрерывной дифференцируемости).

Всегда можно допустить, что задача имеет стандартную форму:

$$f(x) \to \min,$$
$$g(x) = [g_1(x),\ g_2(x),\ \ldots,\ g_m(x)]^T = 0,$$
$$x \geqslant 0.$$

Обозначим через $J = \{1, 2, \ldots, n\}$ множество индексов переменных. Пусть x^0 — реализуемое (текущее) решение, т. е. решение, удовлетворяющее условиям $g(x^0) = 0$, $x^0 \geqslant 0$.

Подмножество переменных $B \subset J$, где $|B| = m$, называется *базисом* в том и только том случае, если матрица (якобиан)

$$\frac{\partial g}{\partial x_B}(x^0) = \left[\frac{\partial g_i}{\partial x_j}(x^0)\right]_{\substack{i=1,\ldots,m \\ j \in B}}$$

регулярна. Тогда вектор x можно представить в виде

$$x = [x_B,\ x_N],$$

где x_B — базисные переменные (зависимые переменные), x_N — внебазисные (независимые) переменные.

Исходя из точки x^0, будем отыскивать направление перемещения, позволяющее уменьшить функцию f, оставаясь в пределах выполнения ограничений.

Вариация df функции f для бесконечно малого перемещения $dx = [dx_B,\ dx_N]$, согласующегося с ограничениями, записывается в виде

$$df = \left[\frac{\partial f}{\partial x_N} - \left(\frac{\partial f}{\partial x_B}\right)\left(\frac{\partial g}{\partial x_B}\right)^{-1}\left(\frac{\partial g}{\partial x_N}\right)\right] dx_N = u_N\, dx_N,$$

где вектор

$$u_N = \frac{\partial f}{\partial x_N} - \left(\frac{\partial f}{\partial x_B}\right)\left(\frac{\partial g}{\partial x_B}\right)^{-1}\left(\frac{\partial g}{\partial x_N}\right)$$

называется *обобщенным приведенным градиентом.* (NB: все якобианы вычислены в текущей точке x^0, и это останется так всюду в дальнейшем.)

После этого находим направление перемещения y_N относительно независимых переменных:

$$y_j = 0, \quad \text{если } u_j > 0 \text{ и } x_j = 0,\ j \in N,$$

$$y_j = -u_j \text{ в противном случае } (j \in N).$$

Тогда направление перемещения y_B относительно базисных переменных выводится из y_N по формуле

$$y_B = -\left(\frac{\partial g}{\partial x_B}\right)^{-1}\frac{\partial g}{\partial x_N}\, y_N.$$

При $y_N = 0$ доказано, что условия Куна — Таккера в текущей точке x^0 выполнены.

При $y_N \neq 0$ находим $\bar{\theta} \geqslant 0$, минимизирующее функцию переменного θ:

$$\psi(\theta) = f(x^0 + \theta y).$$

Поскольку условия положительности переменных всегда должны быть выполнены, то должно выполняться неравенство

$$x_j + \theta y_j \geqslant 0 \quad \forall j,$$

и, значит,

$$\theta \leqslant \theta_{\max} = \min_{y_j < 0}\left\{-\frac{x_j}{y_j}\right\}.$$

Таким образом, находим $\bar{\theta}$, минимизирующее $\psi(\theta)$ на $[0, \theta_{\max}]$ (одномерная минимизация).

После нахождения таким способом значения θ точка $\hat{x} = x^0 + \bar{\theta} y$, очевидно, удовлетворяет ограничениям положительности $(x \geqslant 0)$.

Однако поскольку перемещение y_B базисных переменных было получено, исходя из y_N, линеаризацией ограничений, то нет никаких оснований для того, чтобы $\hat{x}$ удовлетворяло ограничениям $g(x) = 0$. Для получения допустимой точки x^1, исходя из $\hat{x}$, применяем метод Ньютона решения нелинейных систем (в переменных x_B): $g(x_B, \hat{x}_N) = 0$, где точка $\hat{x}_N = x_N^0 + \bar{\theta} y_N$ остается фиксированной и где $\hat{x}_B = x_B^0 + \bar{\theta} y_B$ — начальная точка итераций. Это приводит к рекуррентной формуле

$$x_B^{t+1} = x_B^t - \left[\frac{\partial g}{\partial x_B}(x^0)\right]^{-1} g\left(x_B^t, \hat{x}_N\right)$$

(t есть номер итерации).

Отметим, что в принципе обращение якобиана $[\partial g/\partial x_B]^{-1}$ должно быть оценено на каждой итерации t в точке x_B^t. Однако в большинстве случаев достаточно осуществить это вычисление с обращением якобиана в точке x_B^0; получаем значительный выигрыш во времени. Стало быть, общий метод будет резервирован на случай, когда упрощенный метод не сходится.

А тогда могут представиться две ситуации.

a. Метод Ньютона *сходится к точке* $\bar{x}_B \geqslant 0$. Тогда вновь повторяют итерационную процедуру (нахождения обобщенного приведенного градиента и направления перемещения), с сохранением того же базиса B и с началом в новой точке $x^1 = [x_B, \hat{x}_N]$.

b. Поскольку метод Ньютона не учитывает ограничений положительности переменных x_B, может оказаться, если исходить из $\hat{x}_B \geqslant 0$, что после некоторого числа итераций одна из базисных переменных, например x_r, обращается в нуль или становится отрицательной. Тогда следует осуществить замену базиса. Процедура аналогична той, которая была реализована в линейном случае (алгоритм приведенного градиента): выбираем внебазисную переменную x_s, которая может быть заменена на x_r, причем критерий выбора переменной x_s, вообще говоря, основан на абсолютном значении члена (разрешающего элемента), соответствующего r-й строке и s-му столбцу матрицы вида

$$\left[\frac{\partial g}{\partial x_B}\right]^{-1}\left(\frac{\partial g}{\partial x_N}\right)$$

(якобианы оцениваются в x^0).

После вычисления обращения нового базиса $B' = B + \{s\} - \{r\}$ в x^0 (можно воспользоваться формой обратного произведения) вновь вычисляем обобщенный приведенный градиент (в x^0) относительно нового базиса и находим новое направление перемещения, исходя из x^0.

Сходимость метода к точке, удовлетворяющей условиям Куна — Таккера, установить очень трудно. Лишь недавно удалось получить доказательство сходимости [52, 53, 40] при помощи некоторых модификаций исходной процедуры и при помощи условий, вообще говоря, исключительно трудных для их практической проверки. По-видимому, не существует результата, касающегося скорости сходимости, но можно ожидать, что таковая подвержена тем же ограничениям, что и в методе проекционного градиента (п. 3.7).

Имеется улучшение метода обобщенного приведенного градиента [2] при помощи функции Лагранжа и двойственности. При этом кажущийся рост преимуществ, отсюда вытекающих (алгоритм еще не прошел систематических проверок), существенным образом обязан дополнительной информации, доставляемой двойственной задачей, и при этих условиях можно задаться вопросом — не будут ли методы, которые систематически используют двойственность, во всех случаях предпочтительнее с тройной точки зрения — свойств

сходимости, простоты применения и общности изучаемых задач (гл. 6).

При ближайшем рассмотрении основное преимущество метода обобщенного приведенного градиента кажется основанным на том, что он может эффективно использовать малую плотность матриц (якобианов) и тем самым позволяет решать некоторые задачи относительно большого объема (в несколько десятков переменных и ограничений).

3.6. Методы линеаризации. Успехи симплекс-метода при решении линейных задач, даже больших размеров, побудили многих авторов к развитию методов, называемых *линеаризацией.*

Общий принцип прост: заменяем решение нелинейной задачи решением последовательности линейных задач, аппроксимирующих, в некотором смысле, заданную задачу.

Эти методы существенно используют тот или другой из следующих технических приемов (а возможно и оба).

a) Касательная аппроксимация. Нелинейная непрерывно дифференцируемая функция $h(x)$ заменяется в окрестности точки $x = x^0$ аффинной функцией

$$z = h(x^0) + \nabla h^T(x^0)(x - x^0)$$

(уравнение касательной плоскости в x^0 к графику функции h).

b) Барицентрическая аппроксимация. Пусть имеется нелинейная (выпуклая) функция $h(x)$. Зададим K точек $y^1, y^2, \ldots, y^K$ и заменим функцию $h(x)$ функцией $\widehat{h}$, определяемой соотношениями

$$\widehat{h}(x) = \min\left\{\sum_{k=1}^{K} \lambda_k h(y^k)\right\},$$

$$\sum_{k=1}^{K} \lambda_k y^k = x,$$

$$\lambda \geqslant 0;$$

точно так же, если функция h вогнута, то заменим ее функцией $\widehat{h}$, определяемой соотношениями

$$\widehat{h}(x) = \max\left\{\sum_{k=1}^{K} \lambda_k h(y^k)\right\},$$

$$\sum_{k=1}^{K} \lambda_k y^k = x,$$

$$\lambda \geqslant 0.$$

Процесс, состоящий в замене одной нелинейной задачи последовательностью линейных задач, полученной в результате касательной аппроксимации в единственной текущей точке, вообще говоря, не сходится, в чем легко убедиться на примере (рис. 8).

Область есть выпуклый полиэдр (линейные ограничения), линии постоянного значения — концентрические круги. Взяв x^0 в качестве отправной точки, получим последовательно x^1 и x^2, затем

эти две точки будут последовательно чередоваться до бесконечности.

Для избежания этих неудобств можно воспользоваться различными техническими приемами:

— ограничение перемещений (например, применение одномерного поиска);

— использование кратных точек линеаризации.

Среди методов касательной линеаризации мы опишем последовательно: метод Франка и Вольфе [20], который применяется в случае линейных ограничений; метод Келли [33] секущих плоскостей, который применяется к выпуклым задачам с нелинейной целевой функцией и нелинейными ограничениями. Наконец, будет описан метод Данцига барицентрической аппроксимации.

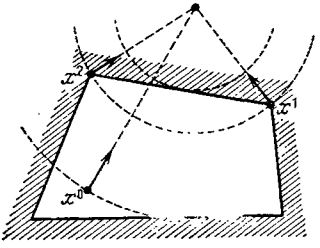


Рис. 8. Отсутствие сходимости метода касательной линеаризации

3.6.1. Метод Франка и Вольфе [20]. Рассматриваются задачи с линейными ограничениями

$$f(x) \to \min,$$
$$Ax = b,$$
$$x \geqslant 0.$$

Метод является итерационным; он порождает последовательность точек $x^0, x^1, \ldots, x^k$, где x^{k+1} определяется (при любом k) по x^k следующим образом.

Начинаем с решения линейной задачи

$$\nabla f^T(x^k)x \to \min,$$
$$Ax = b, \qquad PL(x^k)$$
$$x \geqslant 0.$$

Пусть y^k — крайняя точка множества X — есть оптимальное решение задачи $PL(x^k)$. Тогда x^{k+1} выбрана так, чтобы минимизировать f на отрезке $[x^k, y^k]$. Авторы прежде всего применили свой метод к квадратичному случаю, но можно доказать глобальную сходимость для произвольной непрерывно дифференцируемой функции, как показывает

Теорема 9. *Предположим, что функция f непрерывно дифференцируема и что выполняется одно из двух условий*:

1) *полиэдр $X = \{x \mid Ax = b,\ x \geqslant 0\}$ ограничен*;

2) $f(x) \to +\infty$ *при* $\|x\| \to +\infty$.

Тогда для любой начальной точки $x^0 \in X$ метод Франка — Вольфе сходится к локальному оптимуму задачи $\min\limits_{x \in X} f(x)$.

Доказательство. Обозначим через Ω множество точек из X, удовлетворяющих условиям Купа — Таккера. По условиям теоремы все точки x^k содержатся в компакте. При этом

$$x^k \notin \Omega \Rightarrow f(x^{k+1}) < f(x^k) \quad \forall k$$

и f есть функция спуска. Значит, достаточно доказать, что отображение $U \circ D$ замкнуто, где D — отображение, дающее направление перемещения, U — отображение, представляющее собой одномерную оптимизацию.

Легко показать, что отображение U замкнуто в любой точке (x, d), где $x \in \Omega$, $d = y - x$, причем y — крайняя точка множества X (всегда $d \neq 0$, ибо $d = 0 \Rightarrow x \in \Omega$). С другой стороны, для заданного $\bar{x}$ $(\bar{x} \notin \Omega)$ множество $D(\bar{x})$ есть множество направлений вида $\bar{y} - \bar{x}$, где $\bar{y}$ пробегает множество крайних точек множества X — оптимальных решений задачи

$$\begin{gathered} \nabla f^T(\bar{x})\, x \to \min, \\ Ax = b, \qquad\qquad PL(\bar{x}) \\ x \geqslant 0. \end{gathered}$$

Покажем, что D замкнуто. Пусть $\{x^k\}$ — последовательность точек $(x^k \in X)$, в которой $x^k \to \bar{x}$; рассмотрим такую последовательность $\{y^k\}$, что

$$y^k - x^k \in D(x^k), \qquad y^k \to \bar{y} \quad \forall k.$$

Условие $y^k - x^k \in D(x^k)$ равносильно неравенству

$$\nabla f^T(x^k)\, y^k \leqslant \nabla f^T(x^k)\, x \qquad \forall x \in X.$$

При фиксированном $x \in X$ устремим $k \to +\infty$. Используя непрерывность градиента ∇f, получим

$$\nabla f(x)\, \bar{y} \leqslant \nabla f(\bar{x})\, x.$$

Это соотношение выполняется при любых $x \in X$, поэтому $\bar{y}$ есть оптимальное решение задачи

$$\begin{gathered} \nabla f(\bar{x})\, x \to \min, \\ Ax = b, \\ x \geqslant 0. \end{gathered}$$

Тем самым показано, что $\bar{y} - \bar{x} \in D(\bar{x})$, и, значит, D замкнуто.

Применим теперь предложение 2 гл. 1, п. 4.4 (всегда можно предположить, что полученные направления нормированны, значит, множество Y — образ множества X при отображении D — компактно); отсюда следует, что отображение $U \circ D$ замкнуто. После этого глобальная сходимость алгоритма Франка — Вольфе вытекает из теоремы 10 гл. 1.

3.6.2. Метод Келли секущих плоскостей [33]. Этот метод применяется к общим выпуклым задачам вида

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \qquad i = 1, 2, \ldots, m, \qquad\qquad (P') \\ x \in \mathbf{R}^n, \end{gathered}$$

где функции f и g_i выпуклы и дифференцируемы. Заметим, что всегда можно предположить минимизируемую функцию f в (P') линейной. В самом деле, в противном случае (P') может быть

записана, с введением дополнительной переменной y, в виде

$$y \to \min,$$
$$f(x) - y \leqslant 0,$$
$$g_i(x) \leqslant 0, \qquad i = 1, \ldots, m,$$
$$x \in \mathbf{R}^n, \qquad y \in \mathbf{R},$$

а это — выпуклая задача с линейной целевой функцией в $\mathbf{R}^{n+1}$.

Итак, в дальнейшем мы ограничимся изучением выпуклых задач вида

$$cx = \sum_{j=1}^{n} c_j x_j \to \min,$$
$$g_i(x) \leqslant 0, \qquad i = 1, \ldots, m, \qquad (P'')$$
$$x \in \mathbf{R}^n,$$

где функции g_i выпуклы. Пусть $X = \{x | x \in \mathbf{R}^n,\ g_i(x) \leqslant 0,\ i = 1, \ldots, m\}$ — множество решений задачи (P''). Принцип метода секущих плоскостей состоит, на k-й итерации, в следующем.

a) Аппроксимируем множество решений X таким политопом Q^k, что $X \subset Q^k$.

b) Решаем задачу линейного программирования

$$cx \to \min,$$
$$x \in Q^k,$$

и пусть x^k — оптимальное решение этой задачи.

c) Если x^k не является решением задачи (P''), добавляем к ограничениям, определяющим Q^k, дополнительное ограничение

$$g_i(x^k) + \nabla g_i^T(x^k)(x - x^k) \leqslant 0$$

(где i — индекс, при котором $g_i(x^k) > 0$, например индекс, при котором $g_i(x^k)$ максимально).

d) Переходим к $k+1$-й итерации с

$$Q^{k+1} = Q^k \{x | g_i(x^k) + \nabla g_i^T(x^k)(x - x^k) \leqslant 0\}.$$

Заметим, что в силу выпуклости g_i

$$g_i(x^k) + \nabla g_i^T(x^k)(x - x^k) \leqslant 0,$$

и, значит, $X \subset Q^{k+1}$.

Таким образом, показано, что гиперплоскость с уравнением

$$g_i(x^k) + \nabla g_i^T(x^k)(x - x^k) = 0$$

строго отделяет x^k от X и на этом основании называется *секущей плоскостью*.

Сходимость метода секущих плоскостей обеспечивает следующая

Теорема 10. *Если функции g_i $(i = 1, \ldots, m)$ выпуклы и непрерывно дифференцируемы и если задача (P'') имеет оптимум на конечном расстоянии, то всякая точка сгущения последователь-*

ности $\{x^k\}$, *порожденной методом секущих плоскостей, есть оптимальное решение задачи* (P'').

Доказательство. Если задача (P'') имеет оптимум на конечном расстоянии, то, начиная с некоторого номера K, последовательность точек x^k содержится в ограниченном множестве. Пусть $\{x^t\}_{t\in L}$ $(L\subset \mathbf{N})$ — подпоследовательность, сходящаяся к $\bar{x}$. Пусть i $(1\leqslant i\leqslant m)$ — индекс произвольного ограничения. Рассмотрим подпоследовательность $\{x^t\}_{t\in T}$ $(T\subset L)$ точек, для которых секущая плоскость порождена относительно i-го ограничения (если на каждой итерации добавлять секущую плоскость относительно самого жесткого ограничения, то можно заметить, что либо $g_i(x^t)\leqslant 0$, начиная с некоторого номера $l\geqslant l_0$, либо подпоследовательность $\{x^t\}_{t\in T}$ бесконечна).

В случае, когда подпоследовательность $\{x^t\}_{t\in T}$ бесконечна, для любого $t'\in T$ $(t'>t)$ имеем

$$g_i(x^t)+\nabla g_i^T(x^t)(x^{t'}-x^t)\leqslant 0,$$

откуда

$$g_i(x^t)\leqslant \|\nabla g_i(x^t)\|\,\|x^{t'}-x^t\|.$$

А поскольку

$$\|x^{t'}-x^t\|\to 0,\qquad \|\nabla g_i(x^t)\|\to\|\nabla g_i(\bar{x})\|,$$

то отсюда следует, что

$$g_i(x^t)\to g_i(\bar{x})\leqslant 0,$$

а следовательно, $\bar{x}$ — решение задачи (P'').

С другой стороны, если x^* — оптимальное решение задачи (P''), то на каждой итерации имеем $cx^t\leqslant cx$, откуда следует $c\bar{x}\leqslant cx^*$. А это показывает, что решение $\bar{x}$ оптимально.

3.6.3. Алгоритм Данцига порождения столбцов [14]. Этот метод барицентрической аппроксимации в общем виде применяется лишь к выпуклым задачам, но как целевая функция, так и ограничения могут быть нелинейными.

Итак, ставится задача

$$\begin{gathered} f(x)\to\min,\\ g_i(x)\leqslant 0,\qquad i=1,\ \ldots,\ m,\\ x\in\mathbf{R}^n. \end{gathered}\qquad (P')$$

Обратимся к k-му текущему этапу, когда некоторое число точек $x^1, x^2, \ldots, x^k$ уже найдены. Заменив функции f и g_i их барицентрической аппроксимацией в точках $x^1, \ldots, x^k$, получим линейную задачу

$$\begin{aligned} &\min_{\lambda}\ \sum_{j=1}^{k}\lambda_j f(x^j),\\ &\sum_{j=1}^{k}\lambda_j g_j(x^j)\leqslant 0,\qquad i=1,\ldots,m,\\ &\sum_{j=1}^{K}\lambda_j=1,\quad \lambda_j\geqslant 0\quad \forall j. \end{aligned}\qquad PL(k)$$

Решение этой линейной задачи даст оптимальное решение λ^k и оптимальные двойственные переменные (оптимальные симплекс-множители):

$$\bar{u}^k = (\bar{u}_1^k, \bar{u}_2^k, \ldots, \bar{u}_m^k),$$

соответствующие первым m ограничениям, равно как и оптимальную двойственную переменную u_0, соответствующую последнему ограничению. Тогда точка x^{k+1} будет найдена как оптимальное решение нелинейной задачи оптимизации без ограничений:

$$\min_{x \in R^n} \left\{ f(x) + \sum_{i=1}^{m} \left(-\bar{u}_i^k\right) g_i(x) \right\} - u_0,$$

которая есть не что иное, как минимизация по x функции Лагранжа $L(x, -\bar{u}^k)$. Заметим, что выпуклость по x функции $L(x, -\bar{u}^k)$ позволяет без труда получить глобальный оптимум.

После этого задача $PL(k)$ может быть заменена другой линейной задачей барицентрической аппроксимации $PL(k+1)$ на $k+1$ точках $x^1, x^2, \ldots, x^{k+1}$. Задача содержит на один столбец больше, чем $PL(k)$, откуда и название метода.

Интересно заметить, что этот алгоритм может быть равным образом интерпретирован как метод касательной линеаризации в применении к двойственной функции задачи (P') (гл. 6, п. 3.2). Тогда сходимость двойственных решений $-\bar{u}^k$ к оптимуму u^* двойственной задачи вытекает из теоремы 1 гл. 8, п. 1.6.

На каждом k-м этапе оптимальное решение $\bar{\lambda}^k$ задачи $PL(k)$ позволяет выписать решение $\bar{x}^k$ задачи (P') в виде равенства

$$\bar{x}^k = \sum_{j=1}^{k} \bar{\lambda}_j^k x^j$$

(тот факт, что x есть решение задачи (P'), следует из выпуклости функций g_i), и выполняются соотношения

$$f(x^*) \leqslant f(\bar{x}^k) \leqslant \sum_{j=1}^{k} \bar{\lambda}_j^k f(x^j).$$

Полученная таким способом последовательность точек $\bar{x}^k$ сходится к оптимуму x^* задачи (P'), и в каждый момент значение

$$L(x^{k+1}, -\bar{u}^k) = \min_x L(x, -\bar{u}^k)$$

есть миоранта функции $f(x^*)$. Стало быть, можно прервать вычисления, как только рамки точности

$$L(x^{k+1}, -\bar{u}^k) \leqslant f(x^*) \leqslant f(\bar{x}^k)$$

станут меньше заданного заранее $\varepsilon > 0$.

Когда задача (P') строго выпукла (например, когда функция f строго выпукла, а g_i выпуклы), то последовательность полученных точек x^k сама сходится к оптимуму x^* задачи (P'). Это следует из свойства 5 гл. 6, п. 2.9.

3.7. Анализ сходимости. Оценка алгоритмов. Главными алгоритмами, описанными в этом пункте и применимыми к общему случаю (не обязательно линейных функций и ограничений), являются следующие:

— методы возможных направлений;

— методы проекционного градиента и обобщенного приведенного градиента.

В применении этих методов часто встречаются характерные тонкости. В самом деле, когда мы пытаемся получить конкретные и, что особенно важно, надежные процедуры, согласно этим принципам, то мы приходим к необходимости многочисленных мер предосторожности для спасения от всех встречающихся типов трудностей. Это приводит к высшей степени сложным процедурам — результатам всего накопленного в рассматриваемых методах эмпирического знания.

Это не упрощает теоретический анализ достоинств этих алгоритмов. Этим же объясняется недостаточное число результатов, относящихся к свойствам сходимости рассматриваемых методов, что делает также очень трудным сравнение методов между собой.

В анализе, которому мы будем следовать [35, 36], мы, стало быть, ограничимся включением различных методов в общую схему, немного «идеализированную», но, однако, привлекательную выделением идеи их действия, которую можно извлечь из этих различных алгоритмов.

Прежде всего мы будем предполагать, что мы имеем дело с задачей с ограничениями типа равенства:

$$f(x) \to \min,$$
$$g_i(x) = 0, \qquad i = 1, \ldots, m,$$
$$x \in \mathbb{R}^n.$$

Это предположение оправдано в той мере, в какой представляет интерес именно асимптотическое поведение алгоритмов. В самом деле, в случае ограничений типа $\leqslant$ после некоторого числа итераций активные ограничения известны и все сводится к задаче, в которой должны выполняться только активные ограничения (со знаком равенства).

С другой стороны, мы будем предполагать, что последовательность точек x^k, порожденная различными методами, сходится к x^*, что в x^* выполнены условия Куна — Таккера и что λ^* есть вектор множителей Куна — Таккера в x^*.

Общая основная идея для различных алгоритмов состоит в следующем. На каждой k-й итерации мы перемещаемся из одной допустимой точки x^k в другую допустимую точку x^{k+1}. В x^k вычисляем $-\nabla f(x^k)$ и отсюда получаем направление перемещения y^k. В методах проекционного градиента или приведенного градиента y^k есть проекция вектора $-\nabla f(x^k)$ на гиперплоскость, касательную в x^k к многообразию

$$V = \{x \in \mathbb{R}^n \mid g_i(x) = 0 \quad \forall i\}.$$

В методах возможных направлений типа метода Зойтендейка (п. 3.2) y^k слегка отлично от проекционного градиента, но различие между двумя семействами методов тем более мало, чем меньше шаг перемещения; стало быть, они — с точки зрения асимптотического поведения — эквивалентны.

Поскольку ограничения нелинейны, то полученная перемещением в направлении y^k точка $z = x^k + \alpha^k y^k$, вообще говоря, не принадлежит многообразию V. Чтобы вернуться в V, исходя из z, требуется специальная процедура (например, типа

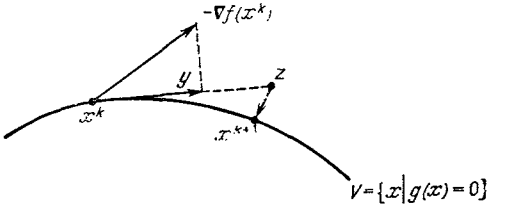


Рис. 9. Поведение методов проекционного градиента или приведенного градиента при нелинейных ограничениях

метода Ньютона). Тогда получаем x^{k+1} (рис. 9). Шаг перемещения α^k должен быть выбран так, чтобы $f(x^{k+1}) < f(x^k)$; стало быть, может понадобиться большое число шагов.

Такая общая схема, хотя она и очень упрощена, все еще слишком сложна для алгоритмического анализа.

Чтобы продвигаться дальше, предположим на мгновение, что можно получить явное описание многообразия V. Тогда задача сведется к оптимизационной задаче без ограничений, на этот раз в многообразии V (нелинейном).

Предположим тогда, что мы применяем к этой задаче градиентный метод типа «наискорейшего спуска». Начав с точки x^k, мы придем к замене некоторой величины вдоль кривой Γ^k (целиком содержащейся в многообразии V), соответствующей самому сильному убыванию функции f.

Заметим, что в этом случае направление перемещения y^k есть не что иное, как касательный вектор к Γ^k в x^k.

Иными словами, для достаточно малых шагов перемещения процедура метода проекционного градиента практически эквивалентна оптимизации функции f на многообразии V методом наискорейшего спуска.

Отсюда при помощи результатов гл. 4, п. 2.12 сразу выводим, что методы проекционного градиента или приведенного градиента демонстрируют линейную сходимость с множителем асимптотической сходимости $\left(\frac{A-a}{A+a}\right)^2$. Здесь A и a — соответственно наибольшее и наименьшее собственное значение гессиана функции f, суженной на многообразие V (с точкой x^* — решением задачи).

Но, как легко видеть, этот гессиан есть не что иное, как гессиан функции Лагранжа:

$$\nabla^2 f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 g_i(x^*),$$

суженной на гиперплоскость, касающуюся многообразия V в точке x^* [36] (λ_i^* — множители Куна — Таккера в точке x^*). Стало быть, скорость сходимости зависит от обусловленности гессиана функции $L(x, \lambda)$ в точке (x^*, λ^*).

Таким образом, как и градиентные методы (прямым расширением которых являются рассматриваемые алгоритмы), основные методы, изучавшиеся в § 3, могут испытывать очень медленную сходимость на плохо обусловленных задачах.

Если оставить в стороне некоторые специальные задачи (например, с линейными ограничениями), указанное неудобство, заставляющее часто применять некоторые тонкости, приводит к тому, что методы, описываемые в следующем параграфе и в гл. 6, будут, вообще говоря, предпочтительнее. Мы покажем, в частности, что некоторые из этих методов позволяют получить сверхлинейную сходимость или даже квадратичную. Одна из причин этого состоит в том, что они используют информацию о двойственной задаче, в частности, через функцию Лагранжа.

§ 4. Оптимизация с ограничениями при помощи решения уравнений Куна — Таккера

В этом параграфе мы будем изучать один класс методов, общий принцип которых состоит в решении уравнений Куна — Таккера методом Ньютона. Эти методы могут рассматриваться как исходно-двойственные, в том смысле, что они действуют одновременно в пространстве исходных переменных и в пространстве множителей Куна — Таккера (двойственных переменных). Их асимптотическое поведение очень интересно, поскольку при обычных условиях регулярности функций f и g_i получаем суперлинейную или квадратичную скорость сходимости.

Для обеспечения же глобальной сходимости эти методы должны, вообще говоря, комбинироваться с другими методами.

4.1. Метод Ньютона. Прежде всего мы рассмотрим только случай ограничений типа равенства.

Рассмотрим задачу

$$f(x) \to \min,$$
$$g_i(x) = 0, \quad i = 1, \ldots, m,$$
$$x \in \mathbf{R}^n.$$

Поиск точки Куна — Таккера сводится к решению системы $n + m$ уравнений с $n + m$ неизвестными (x, λ):

$$\nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0, \tag{11}$$

$$g_i(x) = 0, \quad i = 1, \ldots, m. \tag{12}$$

Метод Ньютона состоит в том, чтобы, исходя из точки (x^k, λ^k), линеаризовать (11) и (12) в окрестности точки x^k и определить точку (x^{k+1}, λ^{k+1}) как решение линеаризованной системы

$$\nabla f(x^k) + \sum_i \lambda_i^k \nabla g_i(x^k) + \left[\nabla^2 f(x^k) + \sum_i \lambda_i^k \nabla^2 g_i(x^k)\right](x^{k+1} - x^k) + \\ + \sum_i (\lambda_i^{k+1} - \lambda_i^k) \nabla g_i(x^k) = 0,$$

$$g_i(x^k) + \nabla g_i^T(x^k)(x^{k+1} - x^k) = 0$$

(градиенты и гессианы берутся относительно переменных x).

Введем функцию Лагранжа

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

и получим систему

$$\begin{array}{c|cc} & n & m \\ \hline n & \nabla_x^2 L(x^k, \lambda^k) & \nabla g_1 \dots \nabla g_m \\ \hline m & \begin{matrix}\nabla g_1^T \\ \dots \\ \nabla g_m^T\end{matrix} & 0 \end{array} \times \begin{bmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla_x L(x^k, \lambda^k) \\ -g(x^k) \end{bmatrix}.$$

Если взять

$$H^k = \nabla_x^2 L(x^k, \lambda^k) = \nabla^2 f(x^k) + \sum_{i=1}^{m} \lambda_i^k \nabla^2 g_i(x^k)$$

(гессиан по x от функции Лагранжа в точке (x^k, λ^k)),

$$J^k = [\nabla g_1(x^k), \nabla g_2(x^k), \dots, \nabla g_m(x^k)]^T = \frac{\partial g}{\partial x}(x^k)$$

(якобиан от g в точке x^k), то система запишется в виде

$$\begin{bmatrix} H^k & (J^k)^T \\ J^k & 0 \end{bmatrix} \times \begin{bmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla f(x^k) - (J^k)^T \lambda^k \\ -g(x^k) \end{bmatrix}.$$

Если предположить выделяемость ограничений и существование решения (x^*, λ^*) и, с другой стороны, предположить существование такой положительной постоянной q, что

$$y^T \nabla_x^2 L(x^*, \lambda^*) \geqslant q \| y \|^2$$

для любого y, удовлетворяющего условию $y^T \nabla g_i(x^*) = 0$, $i = 1, \dots, m$, то можно доказать, что метод Ньютона квадратично сходится к (x^*, λ^*) (см. [41, 45]).

Ясно, что для применения метода необходимо на каждом этапе вычислять:

— градиенты функций f и g_i;

— гессиан по x функции Лагранжа $L(x, \lambda)$ (но нет необходимости во вторых производных функций f и g_i по отдельности).

С другой стороны, заметим, что в первых n уравнениях член $(J^k)^T\lambda^k$ исключается и остается

$$\begin{bmatrix} H^k & (J^k)^T \\ J^k & 0 \end{bmatrix} \times \begin{bmatrix} x^{k+1} - x^k \\ \lambda^{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f(x^k) \\ -g(x^k) \end{bmatrix}. \tag{13}$$

Если предположить матрицу H^k обратимой, то явное решение системы (13) запишется в виде

$$\begin{bmatrix} x^{k+1} - x^k \\ \lambda^{k+1} \end{bmatrix} = \left[\begin{array}{c|c} H^{-1} - H^{-1}J^T[JH^{-1}J^T]^{-1}JH^{-1} & H^{-1}J^T[JH^{-1}J^T]^{-1} \\ \hline [JH^{-1}J^T]^{-1}JH^{-1} & -[JH^{-1}J^T]^{-1} \end{array}\right] \times \begin{bmatrix} -\nabla f(x^k) \\ -g(x^k) \end{bmatrix},$$

где для упрощения записи опущен итерационный индекс k при записи матриц H^k и J^k.

Однако метод Ньютона обладает некоторыми недостатками:

— система (11), (12) удовлетворяется не только для минимумов с ограничением, но и для максимумов или седловых точек с ограничениями;

— последовательность (x^k, λ^k) может не сходиться, если начальная точка выбрана слишком далеко от решения (x^*, λ^*).

Нужно исходить из начальной точки, близкой к оптимуму, а это — весьма стеснительное условие, чтобы строить общие алгоритмы, работающие на принципе метода Ньютона.

Если мы хотим избавиться от этих недостатков, все еще сохраняя полученные свойства сходимости в окрестности оптимума, то мы придем к использованию метода Ньютона в соединении с другими методами.

В первой фазе мы попытаемся отыскать хорошую аппроксимацию точки (x^*, λ^*). Наиболее приемлемы для этого методы штрафа (изучаемые в гл. 6). Тогда метод Ньютона будет применяться, начиная с точки, полученной для улучшения точности (см., например, [8]). Эта идея составляет также одну из мотиваций методов расширенного лагранжиана (см. гл. 6, § 4).

В заключение заметим, что решение уравнений Куна — Таккера может также производиться квазиньютоновскими методами, где на каждом этапе применяется аппроксимация обращения матрицы $\begin{bmatrix} H & J^T \\ J & 0 \end{bmatrix}$. Это позволяет избежать вычисления вторых производных.

4.2. Расширение метода Ньютона: метод Вильсона. Случай ограничений типа неравенства. Положив $y^k = x^{k+1} - x^k$, запишем систему (13) в виде

$$H^k y^k + (J^k)^T\lambda^{k+1} = -\nabla f(x^k),$$
$$J^k y^k = -g(x^k),$$

после этого замечаем, что y^k есть решение задачи квадратичной

оптимизации

$$\min \frac{1}{2} y^T H^k y + \nabla f^T(x^k)\, y, \qquad (Q)$$
$$J^k y + g(x^k) = 0$$

и что λ^{k+1} есть не что иное, как оптимальный двойственный вектор этой квадратичной задачи.

Значит, вместо использования классической формулы Ньютона можно на каждой итерации решать задачу (Q) для получения x^{k+1} и λ^{k+1}, исходя из x^k и λ^k. Тем самым мы избавляемся от стационарных точек, которые не являются локальными минимумами. Это расширение метода Ньютона принадлежит Вильсону [59].

Для решения задачи (Q) можно воспользоваться, например, алгоритмом приведенного градиента (п. 3.4).

Метод Вильсона обладает теми же свойствами квадратичной сходимости оптимума, что и метод Ньютона. Необходимо, однако, отметить трудности случая, когда матрица H^k не является положительно определенной.

Но зато метод Вильсона обладает преимуществом очень легкого обобщения на случай ограничений типа неравенства.

Если вместо ограничений типа равенства имеем ограничения типа неравенства $g_i(x) \leqslant 0$, то легко показать, что достаточно рассмотреть в Q ограничения типа

$$\nabla g_i^T(x^k)\, y + g_i(x^k) \leqslant 0.$$

Тогда на каждом этапе надо решать квадратичную задачу

$$\min \frac{1}{2} y^T H^k y + \nabla f^T(x^k)\, y, \qquad (Q')$$
$$J^k y + g(x^k) \leqslant 0.$$

4.3. Связь с методами возможных направлений. Интересно установить связь между методом Вильсона и методами возможных направлений из § 3. Рассмотрим случай ограничений типа неравенства.

Один из методов возможных направлений, изучавшийся в п. 3.2, состоит в нахождении направления перемещения y, исходя из текущей точки x^k, посредством решения задачи типа

$$\nabla f^T(x^k)\, y \to \min,$$
$$g_i(x^k) + \nabla g_i^T(x^k)\, y \leqslant 0, \qquad i = 1, \ldots, m,$$
$$y^T y \leqslant \alpha$$

(см. задачу (II) § 3.2).

Эта задача нелинейна, в силу ограничения перемещений $y^T y \leqslant \alpha$, но можно учитывать это ограничение, включая его в форме штрафного члена в целевую функцию (гл. 6, § 1). Тогда получаем

$$\nabla f^T(x^k)\, y + \mu y^T I y \to \min,$$
$$g_i(x^k) + \nabla g_i^T(x^k)\, y \leqslant 0, \qquad i = 1, \ldots, m, \qquad (V)$$

где $\mu > 0$ есть коэффициент штрафа, I — единичная $n \times n$-матрица.

Ясно, что задача (V) отличается от задачи (Q') лишь квадратичным членом в целевой функции.

Вообще, более интересно бывает решать задачу (Q') вместо задачи (II): с одной стороны, таким образом мы избегаем одномерной оптимизации, поскольку решение задачи (Q') сразу дает x^{k+1}, а с другой стороны, если иметь дело с (Q'), то результаты сходимости вытекают непосредственно из результатов, относящихся к методу Ньютона.

4.4. Связь с двойственными методами. Метод Вильсона может быть еще улучшен, если заметить, что задача (Q) эквивалентна задаче

$$\min \frac{1}{2} y^T H^k y + \nabla f^T(x^k)\, y + \lambda^k J^k y,$$

$$J^k y + g(x^k) = 0$$

(таким образом, мы оказываемся в случае ограничений типа равенства). В самом деле, целевые функции различаются лишь величиной $\lambda^k g(x^k)$, которая не зависит от y.

В целевой функции мы узнаем аппроксимацию второго порядка (с точностью до постоянной) функции Лагранжа $L(x, \lambda)$ в окрестности точки (x^k, λ^k). Это наводит на мысль о методе [50, 47], в котором для заданных x^k, λ^k на каждой итерации решается нелинейная задача с линейными ограничениями:

$$L(x^k + y, \lambda^k) \to \min_y,$$

$$\nabla g_i^T(x^k)\, y + g_i(x^k) = 0, \qquad i = 1, \ldots, m. \tag{VI}$$

Возьмем $x^{k+1} = x^k + y$, где y — оптимальное исходное решение задачи (VI), и λ^{k+1} — оптимальное двойственное решение задачи (VI).

Поскольку разность между полученными решениями задачи (VI) и задачи (Q) имеет порядок $\|y\|^2$, то квадратичная сходимость метода выводится из результатов, полученных для метода Вильсона.

Таким образом, мы пришли к итерационному методу, в котором на каждом этапе минимизируется функция Лагранжа с учетом ограничений в линеаризованной форме.

Этот метод на самом деле очень близок к методам, использующим понятие двойственности,— методам, которые будут развернуты в следующей главе.

СПИСОК ЛИТЕРАТУРЫ

1. Abadie J. (1967), On the Kuhn-Tucker Theorem, in: *Nonlinear Programming*, (J. Abadie ed.), North Holland, Amsterdam.
2. Abadie J. (1978), Un nouvel algorithme pour la programmation non-linéaire. *R.A.I.R.O. Recherche Opérationnelle*, vol. 12, n° 2, p. 233-238.
3. Abadie J., Carpentier J. (1969), Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, in: *Optimization*, chap. 4, (R. Fletcher ed.), Academic Press.
4. Abadie J., Guigou J. (1970), Numerical experiments with the G.R.G. method, in: *Integer and nonlinear programming*, (J. Abadie ed.), North Holland Publishing Company, Amsterdam, p. 529-536.

5. Arrow K. J., Hurwicz L., Uzawa H. (1961), Constraint qualification in maximization problems, *Nav. Res. Log. Quart.*, 8, p. 175-191.
6. Avriel M. (1976), *Nonlinear Programs: Analysis and Methods*, Prentice Hall.
7. Bazaraa M. S., Goode J. J., Shetty C. M. (1972), Constraint qualification revisited, *Management Science*, 18, p. 567-573
8. Best M. J., Braüninger J., Ritter K., Robinson S. M. (1981), A Globally and Quadratically Convergent Algorithm for General Nonlinear Programming Problems, *Computing*, 26, p. 141-153.
9. Best M. J., Ritter K. (1976), A Class of Accelerated Conjugate Direction Methods for Linearly Constrained Minimization Problems, *Mathematics of Computation*, 30, p. 478-504.
10. Cabay D., Luenberger D. G. (1976), Efficiently converging methods for nonlinear constrained minimization methods based on the Reduced Gradient, *S. I. A. M., Journal Control and Optimization*, vol. 14, n° 1.
11. Canon M. D., Cullum C. D. (1968), A tight upper bound on the rate of convergence of the Frank-Wolfe Algorithm, *S. I. A. M. J., Control* 6, p. 509-516.
12. Canon M. D., Cullum C., Polak E. (1970), *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill, New York.
13. Cottle R. W. (1968), The principal pivoting method of quadratic programming, in: *Mathematics of The Decision Sciences*. (Dantzig G. B. Veinott A. F. eds.), American Mathematical Society, Providence, R. I.
14*. Dantzig G. B. (1963), *Linear programming and extensions*, Princeton University Press, Princeton.
15. Eaves B. C., Zangwill W. I. (1971), Generalized Cutting Plane Algorithms, *S. I. A. J., Control* 9, p. 529-542.
16. Evans J. P. (1970), On Constraint Qualifications in Nonlinear Programming, *Nav. Res. Log. Quart.* 17, p. 281-286.
17. Faure P., Huard P. (1965), Résolution de programmes mathématiques à fonction non linéaire par la méthode du gradient réduit, *Rev. Fr. Rech. Opérationnelle*, n° 36, p. 167-206.
18*. Fiacco A. V., McCormick G. P. (1968), *Nonlinear programming*, John Wiley, New York.
19. Fletcher R. (1972), Minimizing General Functions Subject to Linear Constraints, in: *Numerical Methods for Nonlinear Optimization*, (Lootsma F. A. ed.), Academic Press, New York.
20. Frank M., Wolfe P. (1956), An algorithm for quadratic programming, *Naval Research Logistics Quarterly* 3, p. 95-110.
21*. Gill P. E., Murray W. (1974), *Numerical Methods for Constrained Optimization*, Academic Press.
22. Gill P. E., Murray W. (1974), Newton-Type Methods for Unconstrained and Linearly constrained Optimization, *Mathematical Programming* 7, p. 311-350.
23. Goldfarb D. (1972), Extensions of Newton's Method and Simplex Methods for solving Quadratic Programs, in: *Numerical Methods for Nonlinear Optimization*, (Lootsma F. A. ed.), Academic Press, New York.
24. Gould F. J., Tolle J. W. (1971), A Necessary and Sufficient Qualification for Constrained Optimization, *S. I. A. M. J., Applied Mathematics* 20, p. 164-172.
25. Guigou J. (1975), Une nouvelle variante du gradient réduit, *Bulletin Dir. Et. Rech. E. D. F.*, Série C, n° 1, p. 47-58.
26. Hartley H. O., Hocking R. R. (1963), Convex Programming by Tangential Approximation, *Management Science* 9, p. 600-612.
27. Han S. P. (1976), Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Mathematical Programming* 11, p. 263-282.
28. Hogan W. W. (1973), The continuity of the perturbation function of a convex program, *Operations Research* 21, p. 351-352.
29. Huard P. (1967), Resolution of mathematical programming with nonlinear constraints by the method of centres, in: *Nonlinear programming*, (Abadie ed.), North Holland Publishing Company, Amsterdam, p. 206-219.

30. Huard P. (1970), A method of centers by upper-bounding functions with applications, in: *Nonlinear Programming*, (J. B. Rosen, O. L. Mangasarian, K. Ritter eds.), Academic Press, New York.
31. Huard P. (1971). Tour d'horizon en programmation non-linéaire, *Bulletin Dir. Et. Rech E. D. F.*, Série C, n° 1, p. 35-70.
32*. Karlin S. (1959), *Mathematical methods and theory in games, programming and economics*, Vol. 1, Addison Wesley.
33. Kelley J. E. (1960), The cutting plane method for solving convex programs, *S. I. A. M. Journal* 8, p. 703-712.
34. Kuhn W. W., Tucker A. W. (1951), Nonlinear programming, *Proc. 2nd Berkeley Symp. on Mathematical Statistics and Probability, University of California Press, Berkeley*, p. 481-492.
35. Luenberger D. G. (1973), *Introduction to linear and nonlinear programming*, Addison Wesley.
36. Luenberger D. G. (1976), Algorithmic analysis in constrained optimization, in: *Nonlinear Programming, S. I. A. M.*, AMS Proceedings, Vol. 9, (R. W. Cottle, C. E. Lemke eds.), p. 39-51.
37. McCormick G. P. (1967), Second order conditions for constrained minima, *S. I. A. M. J. Appl. Math.* 15, p. 641-652.
38. Meyer G. G. L. (1974), Accelerated Frank-Wolfe Algorithms, *S. I. A. M. J. Control* 12, p. 655-663.
39. Mokhtar-Kharroubi H. (1979), Sur quelques méthodes de gradient réduit sous contraintes linéaires, *R. A. I. R. O. Analyse Numérique*, V13, 2, p. 167-180.
40. Mokhtar-Kharroubi H. (1980), Sur la convergence théorique de la méthode du gradient réduit généralisé, *Numerische Mathematik*, 34, p. 73-85.
41. Ortega J. M., Rheinboldt W. C. (1970), *Iterative solution of nonlinear equations in several variables*, Academic Press, New York.
42. Nguyen V. H., Strodiot J. J., Mifflin R. (1980), On conditions to have bounded multipliers in locally Lipschitz programming, *Mathematical Programming*, 18. p. 100-106.
43. Peterson D. W. (1973), A Review of Constraint Qualifications in Finite-Dimensional Spaces, *S. I. A. M. Review* 15. p. 639-654.
44. Pironneau O., Polak E. (1973), Rate of convergence of a class of methods of feasible directions, *S. I. A. M. J. Numerical Analysis*, 10, p. 161-174.
45. Powell M. J. D. (1978), Algorithms for nonlinear constraints that use Lagrangian functions, *Mathematical Programming*, 14, p. 224-248.
46. Ritter K. (1973), A superlinearly convergent method for minimization problems with linear inequality constraints, *Mathematical Programming* 4, p. 44-71.
47. Robinson S. M. (1972), A quadratically convergent algorithm for general nonlinear programming problems, *Mathematical Programming* 3, p. 145-156.
48. Rosen J. B. (1960), The gradient projection method for nonlinear programming, part 1: linear constraints, *Journal S. I. A. M.*, 8, p. 181-217.
49. Rosen J. B. (1961), The gradient projection method for nonlinear programming, part 2: nonlinear constraints, *Journal S.I.A.M.*, 9, p. 514-432.
50. Rosen J. B., Kreuser J. (1972), A gradient projection algorithm for nonlinear programming, in: *Numerical methods for nonlinear optimization*, (Lootsma ed.), Academic Press, Londres.
51. Slater M. (1950), Lagrange Multipliers Revisited: A Contribution to Nonlinear Programming, *Cowles Commision Discussion Paper, Mathematics* 403.
52. Smeers Y. (1974), A convergence proof of a special version of the Generalized Reduced Gradient method (GRGS), *R. A. I. R. O.*, vol. 5, n° 3.
53. Smeers Y. (1977), Generalized Reduced Gradient method as an extension of feasible directions methods, *Journal of Optimization Theory and Appl.*, vol. 22, n° 2, p. 209—226.
54. Strodiot J. J., Nguyen V. H. (1982), Kühn-Tucker Multipliers and Nonsmooth Programs, in: *Mathematical Programming Study* 19: *Optimality, Duality and Stability*, M. Guignard ed., p. 222-240.

55. Topkis D. M., Veinott A. F. (1967), On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming, *S.I.A.M. J. Control* 5, p. 268-279.
56. Tui H. (1964), Concave Programming Under Linear Constraints, *Soviet Mathematics* 5, p. 1437-1440.
57. Vajda S. (1961), *Mathematical Programming*, Addison Wesley.
58. Whittle P. (1971), *Optimization Under Constraints: Theory and Applications of Nonlinear Programming*, Wiley, New York.
59. Wilson R. B. (1963), *A simplicial method for convex programming*, Ph. D. Thesis, Harvard University, Cambridge, Mass.
60. Wolfe P. (1959), The Simplex Method for Quadratic Programming, *Econometrica* 27, p. 382-398.
61. Wolfe P. (1961), Accelerating The Cutting Plane Method for Nonlinear Programming, *S.I.A.M. J. Appl. Math.* 9, 3, p. 481-488.
62. Wolfe P. (1963), Methods of nonlinear programming: the reduced gradient method, in: *Recent advances in mathematical programming*, (Graves and Wolfe eds.), McGraw-Hill, New York, p. 67-86.
63. Zangwill W. I. (1967), The Convex Simplex Method, *Management Science* 14, p. 221-283.
64*. Zangwill W. I. (1969), *Nonlinear Programming: A Unified Approach*, Prentice Hall.
65*. Zoutendijk G. (1960), *Methods of feasible directions*, Elsevier Publishing Company, Amsterdam.
66. Zoutendijk G. (1970), Some Algorithms based on the Principle of Feasible Directions, in: *Nonlinear Programming*, (Rosen J. B., Mangasarian O. L., Ritter K. eds), Academic Press, New York.
67. Zoutendijk G. (1976), *Mathematical Programming Methods*, North Holland, Amsterdam.

ГЛАВА 6
НЕЛИНЕЙНАЯ ОПТИМИЗАЦИЯ С ОГРАНИЧЕНИЯМИ

Часть 2. ДВОЙСТВЕННЫЕ МЕТОДЫ (МЕТОДЫ, ИСПОЛЬЗУЮЩИЕ ПОНЯТИЕ ДВОЙСТВЕННОСТИ)

В этой главе мы продолжим изучение задач оптимизации с ограничениями. Нас будут интересовать методы, существенно отличающиеся от методов, описанных выше. Их общий принцип заключается в замене исходной задачи на решение последовательности экстремальных задач без ограничений.

В качестве введения будут описаны так называемые методы штрафа.

Далее будут изучаться методы, основанные на понятии двойственности.

§ 1. Введение. Методы штрафа

Методы штрафа составляют семейство алгоритмов, интересное с двух точек зрения — простоты принципа и практической эффективности. Из большого количества работ, посвященных этим методам, можно указать работы [11, 1, 19]. Мы начнем с изложения основного принципа всех этих методов, а затем приступим к более детальному рассмотрению двух наиболее употребительных вариантов: методов внешних штрафов и методов внутренних штрафов.

1.1. Общий принцип методов штрафа. Рассмотрим задачу

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \quad i = 1, \ldots, m, \\ x \in \mathbf{R}^n. \end{gathered} \tag{P'}$$

Пусть h: $\mathbf{R} \to \mathbf{R}$ — функция вида

$$\begin{aligned} h(y) &= 0, \quad y \leqslant 0, \quad y \in \mathbf{R}, \\ h(y) &= +\infty, \quad y > 0. \end{aligned}$$

Рассмотрим задачу без ограничений (задача со штрафом):

$$\begin{gathered} \varphi(x) = f(x) + H(x) \to \min, \\ x \in \mathbf{R}^n, \end{gathered} \tag{PP}$$

где функция H, называемая *функцией штрафа*, определяется соотношением

$$H(x) = \sum_{i=1}^{m} h(g_i(x)) \quad \forall x.$$

Если задача (P') имеет непустое множество решений $X = \{x | g_i(x) \leqslant 0 \ (i = 1, \ldots, m)\}$, ясно, что оптимум функции $\varphi(x)$ не может достигаться в точке, не принадлежащей X (в этом случае $\varphi(x) = +\infty$). С другой стороны, $H(x) = f(x)$ для любого $x \in X$. Итак, отыскание решения задачи (P') равносильно решению задачи со штрафом (PP), которая является уже задачей без ограничения.

Указанный подход, хотя и привлекательный на первый взгляд, не находит непосредственного применения, ибо очевидно, что функция штрафа H по построению разрывна, и то же самое можно сказать относительно функции φ. А тогда решение задачи (PP) уже само по себе представляет трудную работу, для которой неприменимы методы оптимизации без ограничения, описанные в гл. 4.

1.2. Методы внешних штрафов. Упомянутую выше трудность можно обойти, если использовать непрерывные и непрерывно дифференцируемые функции штрафа. Так, в методе, называемом *методом внешних штрафов*, было предложено [19] принять

$$h(y) = 0, \quad y \leqslant 0,$$
$$h(y) = y^2, \quad y > 0.$$

Положим

$$H(x) = \sum_{i=1}^{m} h(g_i(x)) = \sum_{i=1}^{m} [g_i^+(x)]^2.$$

Тогда задача (P') заменяется на экстремальную задачу без ограничений (задача со штрафом)

$$\varphi(x, r) = f(x) + rH(x) \to \min, \quad x \in \mathbf{R}^n, \qquad (PP)$$

здесь r — *коэффициент штрафа*.

Функция H называется *функцией внешнего штрафа*.

Обозначим через $\bar{x}(r)$ минимум функции $\varphi(x, r)$ при любом $r > 0$. Выбор значения, придаваемого коэффициенту штрафа r, обусловлен следующим компромиссом: с одной стороны, коэффициент r должен быть достаточно большим, чтобы полученная точка $\bar{x}(r)$ была близка к множеству решений X (иначе говоря, чтобы $H(\bar{x}(r))$ было достаточно малым); с другой стороны, если коэффициент r выбран слишком большим, то функция φ может оказаться плохо обусловленной (см. п. 5.2), откуда проистекают вычислительные трудности отыскания оптимума в задаче без ограничений.

Все это объясняет причину того, что методы штрафа используются вообще говоря, в итерационной форме.

Начинаем выбор коэффициента штрафа со значения r_1, не слишком высокого (чтобы избежать вычислительных затруднений), затем решаем задачу без ограничений:

$$\min_{x \in \mathbf{R}^n} \varphi(x, r_1) = f(x) + r_1 H(x).$$

Пусть $\bar{x}(r_1)$ — полученная точка.

Если значение $H(\bar{x}(r_1))$ достаточно мало, то $\bar{x}(r_1)$ будет хорошим приближением для оптимума, и вычисления заканчиваются. В противном случае штраф, соответствующий нарушению ограничений, недостаточно высок. Тогда выбираем коэффициент штрафа $r_2 > r_1$ (например, $r_2 = 10r_1$) и решаем новую задачу без ограничений:

$$\min_{x \in \mathbf{R}^n} \varphi(x, r_2) = f(x) + r_2 H(x).$$

Получим новую точку $\bar{x}(r_2)$, и т. д.

Заметим, что на каждом k-м этапе указанного процесса предпочтительно использовать точку $\bar{x}(r_{k-1})$, полученную на предыдущем этапе, в качестве начальной точки выбранного алгоритма оптимизации без ограничений (метод сопряженного градиента, квазиньютоновский метод, ...).

Метод внешних штрафов сходится к оптимальному решению задачи (P') при весьма малоограничительных условиях, как показывает

Теорема 1. *Пусть* $H\colon \mathbf{R}^n \to \mathbf{R}$ — *функция внешнего штрафа, удовлетворяющая условиям*

$$H(x) \geqslant 0 \quad \forall x,$$

$$H(x) = 0 \Leftrightarrow x \in X = \{x \mid g_i(x) \leqslant 0 \qquad (i = 1, \ldots, m)\},$$

H *непрерывна.*

Пусть f — *непрерывная функция, множество* X *замкнуто и выполняется одно из двух условий*:

1) $f(x) \to +\infty$ *при* $\|x\| \to +\infty$;

2) X *ограничено и* $H(x) \to +\infty$ *при* $\|x\| \to +\infty$.

Тогда при $r \to +\infty$:

a) *последовательность* $\bar{x}(r)$ *имеет по крайней мере одну точку сгущения и всякая точка сгущения этой последовательности есть (глобальное) оптимальное решение задачи* (P');

b) $H(\bar{x}(r)) \to 0$.

Доказательство. Из условий теоремы следует существование оптимального решения x^* задачи (P) (см. теорему Вейерштрасса, гл. 1, п. 2.3). Значит, $f(x) \geqslant f(x^*)$ $\forall x \in X$.

Рассмотрим бесконечную возрастающую последовательность $\{r_k\}$ значений r, удовлетворяющих условию $r_k \to +\infty$ (при $k \to +\infty$), и последовательность $\{x^k\}$ соответствующих значений x $(x^k = \bar{x}(r_k))$.

Выведем несколько полезных соотношений.

Прежде всего заметим, что

$$f(x^{k+1}) + r_{k+1} H(x^{k+1}) > f(x^{k+1}) + r_k H(x^{k+1}) \geqslant f(x^k) + r_k H(x^k),$$

откуда

$$\varphi(x^{k+1}, r_{k+1}) > \varphi(x^k, r_k) \quad \forall k. \tag{1}$$

Далее, по определению x^k и x^{k+1} имеем

$$f(x^k) + r_k H(x^k) \leqslant f(x^{k+1}) + r_k H(x^{k+1}),$$

$$f(x^{k+1}) + r_{k+1} H(x^{k+1}) \leqslant f(x^k) + r_{k+1} H(x^k),$$

откуда в результате сложения

$$(r_{k+1}-r_k)H(x^{k+1})\leqslant(r_{k+1}-r_k)H(x^k),$$

отсюда в силу неравенства $r_{k+1}>r_k$ получаем

$$H(x^{k+1})\leqslant H(x^k)\quad\forall k. \tag{2}$$

Наконец, для любого k можем написать

$$f(x^k)\leqslant f(x^k)+r_kH(x^k)\leqslant f(x^*)+r_kH(x^*),$$

где x^* — оптимальное решение задачи (P'). Отсюда в силу равенства $H(x^*)=0$ $(x^*\in X)$ следует, что

$$f(x^k)\leqslant\varphi(x^k,\ r_k)\leqslant f(x^*)\quad\forall k. \tag{3}$$

В этом случае из условий теоремы вытекает, что любая последовательность $\{x^k\}$ содержится в ограниченном множестве. В самом деле, если выполнено условие 1), то это свойство следует из соотношения (3): $f(x^k)\leqslant f(x^*)$ $\forall k$.

Если выполнено условие 2), то свойство ограниченности вытекает из соотношения (2), которое показывает, что $H(x^k)\leqslant H(x^1)$ $\forall k$, где x^1 — первый член последовательности $\{x^k\}$.

Значит, во всех случаях из $\{x^k\}$ можно извлечь подпоследовательность $\{x^l\}_{l\in L}(L\subset N)$, сходящуюся к x.

В силу непрерывности функции f имеем

$$\lim_{l\in L}f(x^l)=f(\widehat{x}),$$

а согласно (3) имеем $f(\widehat{x})\leqslant f(x^*)$. Последовательность значений $\varphi(x^k,\ r_k)$ в силу (1) монотонно возрастает и согласно (3) мажорируется значением $f(x^*)$. Значит, она имеет предел $\varphi^*\leqslant f(x^*)$. Отсюда следует, что

$$f(x^k)+r_kH(x^k)\to\varphi^*,$$

$$\lim r_lH(x^l)=\varphi^*-f(\widehat{x}),\qquad l\in L.$$

Отсюда, поскольку $H(x^l)\geqslant 0$ и $r_l\to+\infty$, выводим, что

$$H(x^l)\to 0\quad\text{при}\quad l\to\infty,\quad l\in L.$$

После этого, используя непрерывность функции H, приходим к равенству $H(\widehat{x})=0$, и тем самым доказано, что $\widehat{x}\in X$. С другой стороны, в силу неравенства $f(\widehat{x})\leqslant f(x^*)$ из доказанного следует, что $f(\widehat{x})=f(x^*)$, и, значит, $\widehat{x}$ есть оптимальное решение задачи (P').

В более общем виде этот результат будет приведен в гл. 10, п. 4.4, для бесконечномерных экстремальных задач.

1.3. Методы внутренних штрафов. Главное неудобство предыдущего метода заключается в том, что оптимум x^* аппроксимируется снаружи, т. е. различные промежуточные решения x^1, x^2, ..., x^k, полученные при коэффициентах штрафа r_1, r_2, ..., r_k, не принадлежат X (множеству решений). Именно это привело к тому,

что были предложены [19] другие методы штрафа, в которых оптимум аппроксимируется изнутри. (Отсюда название — *методы внутренних штрафов.*)

Предположим, что множество X решений задачи (P') удовлетворяет условиям:

— X имеет непустую внутренность;

— каждая граничная точка множества X есть предел последовательности точек, лежащих внутри X.

Пусть $B(x)$ — функция вида

$$B(x) = -\sum_{i=1}^{m} \frac{1}{g_i(x)}.$$

Она удовлетворяет условиям:

$-B(x) \geqslant 0 \ \forall x \in \operatorname{int} X$;

$-B(x) \to +\infty$, если x стремится к границе множества X;

— при непрерывных функциях g_i $(i = 1, \ldots, m)$ функция $B(x)$ непрерывна во внутренности X.

Такая функция называется *функцией внутреннего штрафа* (говорят еще *барьерная функция*).

Рассмотрим функции

$$\psi(x, t) = f(x) + tB(x);$$

здесь $t > 0$ — *коэффициент штрафа.*

Пусть функция f непрерывна на X и выполняется одно из условий:

1) $f(x) \to +\infty$ при $\|x\| \to +\infty$,

2) X ограничено.

Тогда при любом $t > 0$ функция $\psi(x, t)$ имеет минимум на X, а именно $\bar{x}(t)$, причем $\bar{x}(t) \in \operatorname{int} X$ (это может быть выведено из теоремы Вейерштрасса, гл. 1, п. 2.3).

Принцип методов внутреннего штрафа состоит тогда в следующем.

Выбираем значение $t_1 > 0$ и ищем (итеративным методом оптимизации без ограничений) минимумы функции $\psi(x, t)$, исходя из $x^0 \in \operatorname{int}(X)$. О процессе минимизации для $\psi(x, t_1)$ заметим, что мы никогда не сможем пересечь границу X, потому что при приближении к ней имеем $\psi(x, t_1) \to +\infty$.

Таким образом, мы получаем точку $x^1 = \bar{x}(t_1) \in \operatorname{int}(X)$. Если величина $t_1 B(x^1)$ достаточно мала, то (см. теорему 2 ниже) точка x^1 является хорошим приближением оптимума f на X, и вычисления останавливаются.

В противном случае мы выберем значение $t_2 < t_1$ и будем заново искать точку $x^2 = \bar{x}(t_2)$ минимума $\psi(x, t_2)$, исходя из $x^1 = \bar{x}(t_1)$ — из предыдущей точки. Процесс повторяется вплоть до получения приемлемой аппроксимации оптимума для задачи (P').

Следующий результат показывает, что, принимая несколько менее ограничительные предположения, можно обеспечить сходимость метода внутренних штрафов к оптимуму задачи (P').

Теорема 2. *Пусть* $X = \{x | g_i(x) \leqslant 0 \ \ (i = 1, \ldots, m)\}$ — *множество решений задачи* (P').

Предположим, что X *замкнуто, имеет непустую внутренность и что любая точка* $x \in X$ *есть предел последовательности точек, принадлежащих внутренности* X.

Пусть $B(\mathbf{R}^n \to \mathbf{R})$ — *функция внутреннего штрафа, удовлетворяющая следующим условиям:*

— $B(x) \geqslant 0 \ \forall x \in \operatorname{int} X$;

— $B(x) \to +\infty$ *при* x, *стремящемся к границе* X;

— $B(x)$ *непрерывна на* $\operatorname{int} X$.

Предположим, с другой стороны, что f — *непрерывная функция и выполнено хотя бы одно из следующих двух условий:*

1) $f(x) \to +\infty$ *при* $\|x\| \to +\infty$;

2) X *ограничено.*

Тогда при коэффициенте штрафа t, *стремящемся к нулю, имеем:*

— *последовательность* $\bar{x}(t)$ *имеет по крайней мере одну точку сгущения, и каждая точка сгущения последовательности* $\bar{x}(t)$ *есть (глобальный) оптимум задачи* (P');

— *величина* $tB(\bar{x}(t))$ *стремится к нулю.*

Доказательство. Заметим, что из условий теоремы 2 следует существование точки $x^* \in X$, являющейся оптимальным решением задачи (P') (см. теорему Вейерштрасса, гл. 1, п. 2.3).

Рассмотрим тогда бесконечную убывающую последовательность $\{t_k\}$ положительных значений t ($t_k \to 0$ при $k \to \infty$), и пусть $\{x^k\}$ — последовательность соответствующих значений x (так что $x^k = \bar{x}(t_k)$).

Для каждого k можно написать

$$f(x^*) \leqslant f(x^k) \leqslant f(x^k) + t_k B(x^k) = \psi(x^k, t_k). \qquad (4)$$

Вследствие непрерывности f и в соответствии с предположениями, сделанными относительно множества X, для любого $\varepsilon > 0$ существует такой $\tilde{x} \in \operatorname{int} X$, что

$$f(\tilde{x}) \leqslant f(x^*) + \varepsilon.$$

Отсюда следует, что для любого k

$$f(x^*) + \varepsilon + t_k B(\tilde{x}) \geqslant f(\tilde{x}) + t_k B(\tilde{x}) \geqslant \psi(x^k, t_k).$$

И, следовательно,

$$\lim_{k \to \infty} \psi(x^k, t_k) \leqslant f(x^*) + \varepsilon.$$

Это соотношение остается справедливым $\forall \varepsilon > 0$, откуда, используя (4), выводим, что

$$\lim_{k \to \infty} \psi(x^k, t_k) = f(x^*),$$

$$\lim_{k \to \infty} f(x^k) = f(x^*),$$

$$\lim_{k \to \infty} t_k B(x^k) = 0.$$

Но теперь из условий теоремы 2 следует, что вся последовательность $\{x^k\}$ содержится в ограниченном множестве. Следовательно, можно выделить подпоследовательность $\{x^l\}_{l\in L}$ $(L\subset \mathbf{N})$, сходящуюся к некоторому $\bar{x}$. Вследствие непрерывности f имеем $f(\bar{x})=f(x^*)$, а вследствие замкнутости X имеем $x^l\in X\Rightarrow \bar{x}\in X$. Отсюда получаем, что *любая точка сгущения последовательности* $\{x^k\}$ есть (глобальный) оптимум для (P').

Замечание. Как и в доказательстве теоремы 1, можно без труда установить следующие соотношения:

$$\psi(x^{k+1},\ t_{k+1})<\psi(x^k,\ t_k) \quad \forall k,$$
$$B(x^k)\leqslant B(x^{k+1}) \quad \forall k,$$
$$f(x^{k+1})\leqslant f(x^k) \quad \forall k.$$

Последнее неравенство показывает, в частности, что последовательность полученных значений целевой функции монотонно не возрастает.

Методы внутренних штрафов представляют большой интерес при исследовании задач оптимизации с сильно нелинейными ограничениями, для которых большая часть прямых методов, описанных в гл. 5 (например, метод возможных направлений), встречает большие трудности. Но, со своей стороны, эти методы предполагают, что известно исходное решение x^0, принадлежащее внутренности множества решений X.

Кроме того, их запуск в работу требует некоторых предосторожностей — в частности, на уровне процедур одномерного поиска,— чтобы избежать выхода за пределы множества решений X.

1.4. Оптимальная аппроксимация множителей Куна — Таккера. Одной из интересных особенностей методов штрафов является их способность давать хорошие аппроксимации оптимальных множителей Куна — Таккера.

Рассмотрим сначала метод внешних штрафов (п. 1.2), в котором точка x^k, полученная на k-й итерации, дает минимум функции

$$\varphi(x, r_k)=f(x)+r_k\sum_{i=1}^{m}\left[g_i^+(x)\right]^2.$$

Если функции f и g_i непрерывно дифференцируемы, то точка x^k удовлетворяет, таким образом, условию стационарности для $\varphi(x, r_k)$, что записывается в виде

$$\nabla f(x^k)+2r_k\sum_{i=1}^{m}g_i^+(x^k)\nabla g_i(x^k)=0. \tag{5}$$

В предположениях теоремы 1 имеем $x^k\to x^*$, где x^* — оптимум задачи (P').

Предположим, с другой стороны, что x^* — *регулярная точка*, т. е. что градиенты ограничений, удовлетворяемых точкой x^*, линейно независимы. Тогда вследствие замечания, сделанного в гл. 5, п. 1, 2, существует такой *однозначно определенный вектор Куна* —

Таккера $\lambda^* \geqslant 0$, что

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0, \tag{6}$$

$$\lambda_i^* g_i(x^*) = 0. \tag{7}$$

Пусть $I^0 = \{i | g_i(x^*) = 0\}$ — множество индексов насыщенных в x^* ограничений, т. е. удовлетворяемых точкой x^* в виде равенства. Для любого $i \notin I^0$ имеем $g_i(x^*) < 0$ и, следовательно, существует такое K, что

$$\forall i \notin I^0, \quad k > K \Rightarrow g_i(x^k) < 0.$$

Следовательно, $k > K \Rightarrow 2r_k g_i^+(x^k) = 0$.

Тогда для $\forall k > K$ можно написать

$$\nabla f(x^k) + \sum_{i \in I^0} 2r_k g_i^+(x^k) \nabla g_i(x^k) = 0,$$

откуда выводим (вследствие непрерывности ∇f и ∇g_i), что для $i \in I^0$ величина $2r_k g_i^+(x^k)$ стремится к λ_i^*, т. е. к единственному решению задачи (6) — (7).

Мы видим также, что для всех $i = 1, \ldots, m$ величина $2r_k g_i^+(x^k)$ образует приближение оптимального множителя Куна — Таккера λ_i^*.

Рассмотрим теперь метод внутренних штрафов (п. 1.3), в котором точка x^k, получаемая на k-й итерации, дает минимум функции

$$\psi(x, t_k) = f(x) - t_k \sum_{i=1}^{m} \frac{1}{g_i(x)}.$$

Если предположить, что функции f и g_i непрерывно дифференцируемы, то условие стационарности ψ можно записать в виде

$$\nabla f(x^k) + t_k \sum_{i=1}^{m} \left(\frac{1}{g_i(x^k)} \right)^2 \nabla g_i(x^k) = 0.$$

Рассуждение, аналогичное предыдущему, показывает тогда, что если $x^k \to x^*$ (x^* — оптимум задачи (P')), и если x^* — *регулярная точка*, то для любого $i = 1, \ldots, m$ величина $t_k (1/g_i(x^k))^2$ стремится к λ_i^* — к оптимальному множителю Куна — Таккера, связанному с i-м ограничением.

1.5. Комбинация методов штрафа с другими методами. Методы штрафов являются простыми и эффективными методами как для быстрого получения хороших приближенных решений задачи типа (P'), так и для хорошей аппроксимации оптимальных множителей Куна — Таккера. Однако они, вообще говоря, не позволяют достичь заданной точности, так как это требовало бы выбора либо слишком больших (внешние штрафы), либо слишком малых (внутренние штрафы) коэффициентов штрафа, что приводит к плохой обуслов-

ленности φ и ψ и к численным трудностям. Именно поэтому методы штрафа часто используются в сочетании с другими методами.

Например, исходя из хороших приближений для x^* (оптимальных исходных переменных) и λ^* (оптимальных дуальных переменных), полученных методом штрафа, можно применить метод Ньютона (или один из его вариантов) для решения уравнений Куна — Таккера (см. гл. 5, § 4): свойство квадратичной сходимости в окрестности решения (x^*, λ^*) позволяет тогда получить заданную точность (см., например, [6]).

Как мы увидим в § 4 настоящей главы, комбинируя методы штрафов с изложенными ниже, в § 2, 3, двойственными методами, можно получить другие, также высокоэффективные алгоритмы (методы расширенных лагранжианов).

§ 2. Классическая лагранжева двойственность

Рассмотрим следующую задачу математического программирования:

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \qquad i \in I = \{1, \ldots, m\}, \\ x \in S \subset \mathbf{R}^n. \end{gathered} \tag{P}$$

Функция Лагранжа $L(x, \lambda)$, введенная нами в предыдущей главе, имеет вид

$$L(x, \lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x).$$

Мы видели в предыдущей главе, что задачу (P) можно решить, если определить седловую точку функции Лагранжа, т. е. такую пару $(\bar{x}, \bar{\lambda})$, что

$$\begin{gathered} L(\bar{x}, \bar{\lambda}) = \min_{x \in S} L(x, \bar{\lambda}), \\ g(\bar{x}) \leqslant 0, \\ \bar{\lambda}_i g_i(\bar{x}) = 0 \qquad (i = 1, \ldots, m). \end{gathered}$$

Определим теперь в области $\lambda \geqslant 0$ функцию $w(\lambda)$, полагая

$$w(\lambda) = \inf_{x \in S} \{L(x, \lambda)\}.$$

В дальнейшем мы будем предполагать, что функции f и g_i и множество S таковы, что в любой точке λ, в которой $w(\lambda)$ имеет конечное значение, существует такой $\bar{x} \in S$, что $w(\lambda) = L(\bar{x}, \lambda)$. Тогда можно написать

$$w(\lambda) = \min_{x \in S} \{L(x, \lambda)\}.$$

Заметим, что это условие выполняется, в частности, если функции f и g_i непрерывны на S, причем S компактно (см. теорему Вейерштрасса, гл. 1, п. 2.3).

Мы сейчас увидим, что поиск седловой точки, если она существует, можно осуществить, решая задачу

$$\max_{\lambda} w(\lambda) = \max_{\lambda} \{\min_{x \in S} L(x, \lambda)\},$$

$$\lambda \in \mathbf{R}^{m+}; \qquad (D)$$

задача (D) называется *дуальной (двойственной) задачей* для (P). Наоборот, (P) называется тогда *исходной задачей* или *прямой задачей*, w называется *дуальной (двойственной) функцией*.

Заметим — и это важно,— что дуальная функция w и дуальная задача (D) определены также и тогда, когда седловая точка не существует.

2.1. Свойство 1 (слабая теорема двойственности). *Для любого $\lambda \in \mathbf{R}^{m+}$ значение дуальной функции $w(\lambda)$ не превосходит абсолютного оптимума $f(x^*)$ задачи (P), иначе говоря, если $w(\lambda^*)$ — оптимальное значение двойственной задачи, то*

$$w(\lambda) \leqslant w(\lambda^*) \leqslant f(x^*) \qquad \forall \lambda \geqslant 0.$$

Доказательство. Возьмем некоторое $\lambda \geqslant 0$.

По определению $x \in S \Rightarrow$

$$w(\lambda) \leqslant f(x) + \sum_{i \in I} \lambda \cdot g_i(x).$$

Если x — решение (P), то имеем $g(x) \leqslant 0$, поэтому $\lambda g(x) \leqslant 0$ и, следовательно, $w(\lambda) \leqslant f(x)$.

В частности, если x^* — оптимальное решение (P), то имеем $\forall \lambda \geqslant 0$:

$$w(\lambda) \leqslant w(\lambda^*) \leqslant f(x^*).$$

2.2. Свойство 2 (вогнутость дуальной функции). *Дуальная функция w вогнута как функция λ.*

Доказательство. Возьмем некоторые λ^1 и λ^2 и для некоторого $\theta \in [0, 1]$ положим $\lambda = \theta\lambda^1 + (1-\theta)\lambda^2$.

Тогда существует такое $\bar{x}$, что

$$w(\lambda) \leqslant f(\bar{x}) + \sum_{i=1}^{m} \lambda_i g_i(\bar{x}).$$

Тогда по определению $w(\lambda^1)$ и $w(\lambda^2)$ имеем

$$w(\lambda^1) \leqslant f(\bar{x}) + \sum_{I} \lambda_i^1 g_i(\bar{x}),$$

$$w(\lambda^2) \leqslant f(\bar{x}) + \sum_{I} \lambda_i^2 g_i(\bar{x});$$

умножим первое неравенство на $\theta \geqslant 0$, второе на $1-\theta \geqslant 0$ и сложим.

Получаем:

$$\theta w(\lambda^1) + (1-\theta) w(\lambda^2) \leqslant f(\bar{x}) + \sum_{I} [\theta\lambda_i^1 + (1-\theta)\lambda_i^2] g_i(\bar{x}) =$$

$$= f(\bar{x}) + \sum_{I} \lambda_i g_i(\bar{x}) = w(\lambda),$$

Это последнее свойство — совершенно общее, и оно не предполагает ни выпуклости функций f и g_i, ни выпуклости множества S.

Замечание. В частном случае, когда S — дискретное подмножество $\mathbf{Z}^n$ (целочисленное программирование), функция w дифференцируема не в каждой точке (мы видели, что ее график есть нижняя оболочка конечного или бесконечного семейства гиперплоскостей в $\mathbf{R}^{n+1}$, см. гл. 4, п. 3.1). Вогнутость позволяет, однако, утверждать, что локальный оптимум λ^0 для w является глобальным оптимумом. Поэтому дуальную задачу решать, вообще говоря, проще, чем исходную задачу (P), и это — один из доводов, в силу которых понятие двойственности чрезвычайно употребительно в математическом программировании.

В случае целочисленного программирования свойство

$$\max_{\lambda} w(\lambda) = w(\lambda^*) \leqslant f(x^*)$$

всегда приводит к тому, что величина $w(\lambda^*)$ (или приближение этой величины) очень часто выступает как функция оценки в рамках процедур исследования с помощью разделения и оценки (SEP — разделение и прогрессивная оценка, разветвленный поиск, метод ветвей и границ и т. д.; см. гл. 7).

2.3. Свойство 3 (теорема двойственности).

a) *Если задача* (P) *имеет седловую точку* (x^*, λ^*), *то*

$$\max(D) = w(\lambda^*) = f(x^*) = \min(P).$$

Иначе говоря, оптимальное значение исходной задачи (P) *равно оптимальному значению двойственной задачи* (D).

b) *Обратно, если существует такое решение* x^* *задачи* (P) *и такое* $\lambda^* \geqslant 0$, *что* $w(\lambda^*) = f(x^*)$, *то* (P) *имеет седловую точку, и* (x^*, λ^*) — *такая точка.*

Доказательство.

a) Так как (x^*, λ^*) — седловая точка, то

$$L(x^*, \lambda^*) = f(x^*) + \lambda^* g(x^*) = f(x^*) = \min_{x \in S} \{L(x, \lambda^*)\} = w(\lambda^*).$$

С другой стороны, вследствие свойства 1 для любого $\lambda \geqslant 0$ имеем $w(\lambda) \leqslant f(x^*)$. Отсюда следует, что

$$f(x^*) = w(\lambda^*) = \max_{\lambda \geqslant 0} \{w(\lambda)\}.$$

b) Обратно, предположим, что существует такое решение x^* задачи (P) и такое $\lambda^* \geqslant 0$, что $w(\lambda^*) = f(x^*)$.

По определению $w(\lambda^*)$ имеем

$$w(\lambda^*) \leqslant f(x) + \lambda^* g(x) \qquad \forall x \in S.$$

В частности, полагая $x = x^*$, получаем

$$w(\lambda^*) = f(x^*) \leqslant f(x^*) + \lambda^* g(x^*),$$

откуда следует, что $\lambda^* g(x^*) \geqslant 0$.

Но $\lambda^* \geqslant 0$ и $g(x^*) \leqslant 0$, поэтому имеем также и $\lambda^* g(x^*) \leqslant 0$ и, следовательно, $\lambda^* g(x^*) = 0$.

Так как $\lambda^* g(x^*)$ есть сумма отрицательных или нулевых слагаемых, то отсюда следует, что $\forall i$ имеем $\lambda_i^* g_i(x^*) = 0$ и, следовательно, (x^*, λ^*) — седловая точка.

Заметим, что свойство 3а) приложимо, в частности, к выпуклому программированию.

2.4. Пример 1. Рассмотрим задачу

$$x_1^2 + x_2^2 \to \min,$$
$$2x_1 + x_2 \leqslant -4.$$

Здесь

$$f(x_1, x_2) = x_1^2 + x_2^2, \qquad g(x_1, x_2) = 2x_1 + x_2 + 4,$$

f выпукла, g выпукла. Следовательно, седловая точка существует и

$$w(\lambda^*) = f(x^*),$$

$$L(x, \lambda) = x_1^2 + x_2^2 + 2\lambda x_1 + \lambda x_2 + 4\lambda.$$

Минимум по x функции $L(x, \lambda)$ определяется условиями

$$\partial L/\partial x_1 = 2x_1 + 2\lambda = 0 \Rightarrow x_1 = -\lambda,$$
$$\partial L/\partial x_2 = 2x_2 + \lambda = 0 \Rightarrow x_2 = -\lambda/2.$$

Отсюда получаем дуальную функцию $w(\lambda) = -5\lambda^2/4 + 4\lambda$. Она удовлетворяет условию 2 (вогнутость).

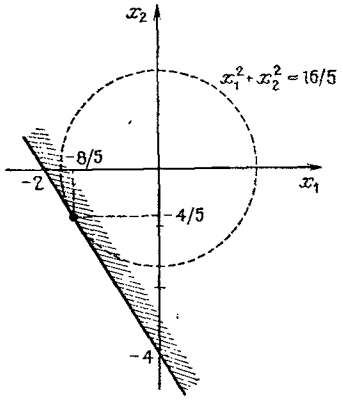


Рис. 1. Пример 1: оптимальное решение исходной задачи

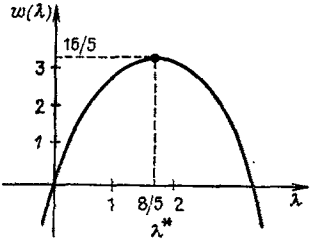


Рис. 2. Пример 1: дуальная функция и оптимум дуальной задачи

Максимум $w(\lambda)$ тогда достигается при условии $dw/d\lambda = -5\lambda/2 + 4 = 0$, откуда $\lambda^* = 8/5$.

Тогда получаем (см. рис. 1 и 2)

$$f(x^*) = w(\lambda^*) = -\frac{5}{4}\left(\frac{8}{5}\right)^2 + \frac{32}{5} = \frac{16}{5}, \qquad x_1^* = -\frac{8}{5}, \quad x_2^* = -\frac{4}{5}.$$

2.5. Пример 2. Двойственность в линейном программировании. В этом случае f и g_i выпуклы, седловая точка существует, а условия Куна — Таккера необходимы и достаточны.

Рассмотрим задачу (без ограничения на знак x)

$$cx \to \min,$$
$$b - Ax \leqslant 0, \qquad (P)$$
$$x \in \mathbf{R}^n,$$

$$c = (c_1, c_2, \ldots, c_n), \quad b = (b_1, b_2, \ldots, b_m)^T, \quad A\,(m \times n).$$

Запишем функцию Лагранжа

$$L(x, \lambda) = cx + \lambda^T (b - Ax),$$

откуда

$$w(\lambda) = \min_x \{(c - \lambda^T A)\, x + \lambda^T b\}.$$

Тогда получаем:

$w(\lambda) = \lambda^T b$ для всех таких λ, что $c - \lambda^T A = 0$,

$w(\lambda) = -\infty$ для остальных значений λ.

Двойственная задача

$$w(\lambda) \to \max,$$
$$\lambda \in \mathbf{R}^{m+}$$

оказывается тогда задачей линейного программирования:

$$\lambda^T b \to \max,$$
$$c - \lambda^T A = 0, \qquad (D)$$
$$\lambda \geqslant 0.$$

С другой стороны, условия Куна — Таккера, являющиеся необходимыми и достаточными, показывают, что в точке оптимума

$$\lambda_i^T (b_i - A_i x) = 0 \qquad \forall i = 1, \ldots, m$$

(условия дополнительности).

Для задач типа (P), не допускающих седловой точки (это как раз общий случай; в том числе в дискретном программировании, когда S — дискретное множество), получаем строгое неравенство $w(\lambda^*) < f(x^*)$, и разность $f(x^*) - w(\lambda^*)$ называется *скачком* или *разрывом двойственности*.

2.6. Пример 3.

Рассмотрим задачу

$$f(x) = 10 - 3x_1 - 2x_2 - x_3 \to \min,$$
$$2x_1 + 3x_2 + 4x_3 \leqslant 4,$$
$$0 \leqslant x_i \leqslant 1, \qquad i = 1, 2, 3.$$

Ее оптимальное решение в вещественных числах есть набор

$$x_1 = 1, \quad x_2 = 2/3, \quad x_3 = 0, \quad f(x) = 17/3.$$

Предположим теперь, что мы ищем оптимум той же задачи в целых числах — иначе говоря, при ограничении $x_i = 0$ или 1 $(i = 1, 2, 3)$.

Тогда получаем задачу

$$z = 10 - 3x_1 - 2x_2 - x_3 \to \min,$$
$$2x_1 + 3x_2 + 4x_3 \leqslant 4,$$
$$x = (x_1, x_2, x_3)^T \in S = \{0, 1\}^3.$$

Множество S конечно и содержит $2^3 = 8$ элементов, обозначаемых $y^1, y^2, \ldots, y^8$, где $y^1 = (0, 0, 0)$, $y^2 = (1, 0, 0)$ и т. д.

Свяжем с ограничением $g(x) = 2x_1 + 3x_2 + 4x_3 - 4 \leqslant 0$ двойственную переменную $\lambda \geqslant 0$.

Для каждого из восьми решений $y^1, y^2, \ldots, y^8$ функция $L(y^i, \lambda)$ определяется равенством

$$L(y^i, \lambda) = f(y^i) + \lambda g(y^i).$$

Следовательно, эта функция линейна по λ, и дуальная функция $w(\lambda)$ определяется равенством

$$w(\lambda) = \min_{y^i \in S} \{L(y^i, \lambda)\}$$

— это нижняя оболочка 8 прямых $L(y^i, \lambda)$ для $i = 1, 2, \ldots, 8$.

Эта функция представлена на рис. 3.

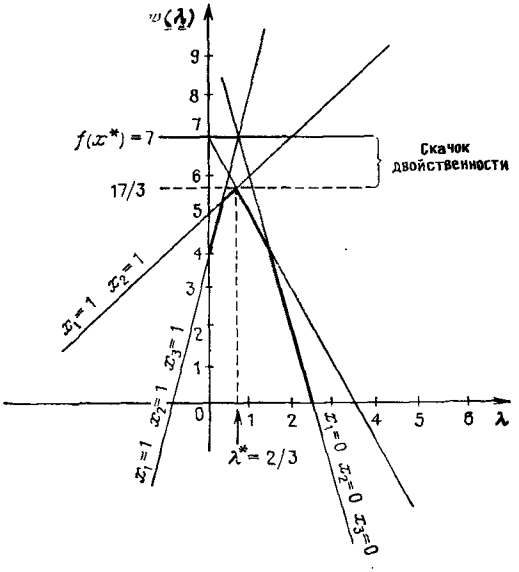


Рис. 3. Скачок двойственности в задаче целочисленного программирования

Заметим, что оптимум дуальной задачи достигается при $\lambda^* = 2/3$, и соответствующее значение равно $w(\lambda^*) = 17/3$ — оптимуму задачи для вещественных чисел.

Но оптимум целочисленной задачи есть (1, 0, 0) со значением 7: следовательно, здесь есть скачок двойственности 4/3 и седловой точки нет.

Перед тем как сформулировать фундаментальное для решения двойственной задачи свойство 4, напомним, что субградиентом функции w в точке λ называется любой вектор $\gamma=(\gamma_1, \gamma_2, \ldots, \gamma_m)^T$, удовлетворяющий условию

$$w(\lambda')\leqslant w(\lambda)+(\lambda'-\lambda)y \qquad \forall\lambda'\in \mathbf{R}^{m+}.$$

Субдифференциал функции w в точке λ (множество всех субградиентов w в точке λ) есть замкнутое выпуклое множество, обозначаемое $\partial w(\lambda)$ (см. гл. 1, п. 3.5).

Тогда выполняется

2.7. Свойство 4 (субградиент). *Для* $\bar{\lambda}\in \mathbf{R}^{m+}$ *положим*

$$Y(\bar{\lambda})=\left\{y\in S \mid f(y)+\sum_I \bar{\lambda}_i g_i(y)=w(\bar{\lambda})\right\}.$$

Тогда для любого $y\in Y(\bar{\lambda})$ *вектор* $g(y)$ *есть субградиент* w *в точке* $\bar{\lambda}$.

Доказательство. По определению w имеем для любого $\lambda\in \mathbf{R}^{m+}$ и любого $x\in S$

$$w(\lambda)\leqslant f(x)+\lambda g(x).$$

В частности, для $y\in Y(\lambda)$

$$w(\lambda)\leqslant f(y)+\lambda g(y)$$

и, по определению $Y(\bar{\lambda})$,

$$w(\bar{\lambda})=f(y)+\bar{\lambda}g(y).$$

Вычитая, получаем

$$w(\lambda)-w(\bar{\lambda})\leqslant(\lambda-\bar{\lambda})g(y),$$

что и доказывает, что $g(y)\in\partial w(\bar{\lambda})$.

Вследствие свойства 4 все векторы вида $g(y)$ с $y\in Y(\lambda)$ являются субградиентами w в точке λ. Следующая теорема показывает, что, и обратно, любой субградиент можно выразить как выпуклую комбинацию векторов $g(y)$, где y пробегает $Y(\lambda)$.

Теорема 3. *Предположим, что функции* f *и* g_i $(i\in I)$ *непрерывны на* S *и что выполняется одно из следующих двух условий:*

1) S *компактно;*

2) *существует такая окрестность* V *точки* $\bar{\lambda}$, *что для любого* $\lambda\in V$ *множество* $Y(\lambda)$ *непусто и содержится в фиксированном компакте* $K\subset S$.

Тогда субдифференциал $\partial w(\bar{\lambda})$ *совпадает с выпуклой оболочкой множества* $\Gamma=\{g(y)\mid y\in Y(\bar{\lambda})\}$.

Доказательство. Из условий теоремы следует, что w — собственная вогнутая функция в окрестности точки $\bar{\lambda}$ и, следовательно, для любого $d=(d_1, d_2, \ldots, d_m)$ существует предел (при

$\theta \to 0^+$) отношения

$$\frac{w(\bar{\lambda}+\theta d)-w(\bar{\lambda})}{\theta},$$

и он равен $\delta w(\bar{\lambda}, d)$ — производной функции w по направлению d в точке λ (см. гл. 1, п. 3.6). Кроме того, $\partial w(\bar{\lambda})$ — замкнутое выпуклое множество, и, согласно теореме 9 гл. 1, имеем

$$\delta w(\bar{\lambda}, d) = \min_{\gamma \in \partial w(\bar{\lambda})} \{d\gamma\}.$$

I. Покажем сначала, что существует такой $\bar{y} \in Y(\bar{\lambda})$, для которого

$$\delta w(\bar{\lambda}, d) \geqslant dg(\bar{y}).$$

Рассмотрим такую последовательность $\{\theta^k\}$ вещественных чисел, что $\theta^k \to 0^+$. Любому из θ^k можно сопоставить такой $y^k \in Y(\bar{\lambda}+\theta^k d)$, что

$$w(\bar{\lambda}+\theta^k d) = f(y^k)+(\bar{\lambda}+\theta^k d)g(y^k).$$

Если условие (1) выполнено, то все y^k содержатся в компактном множестве S. Если же выполнено условие 2), то можно всегда выбрать последовательность $\{\theta^k\}$ так, чтобы $\forall k$: $\bar{\lambda}+\theta^k d \in V$, тогда все y^k содержатся в компактном подмножестве $K \subset S$. Следовательно, мы всегда можем выбрать из $\{y^k\}$ такую подпоследовательность $\{y^l\}_{l \in L}$, что $\{y^l\}$ сходится к $\bar{y} \in S$. Для всех $x \in S$ можно тогда записать

$$f(x)+(\bar{\lambda}+\theta^l d)g(x) \geqslant f(y^l)+(\bar{\lambda}+\theta^l d)g(y^l) \qquad \forall l \in S.$$

Переходя к пределу при $l \to \infty$ ($l \in L$), получаем отсюда

$$f(x)+\bar{\lambda}g(x) \geqslant f(\bar{y})+\bar{\lambda}g(\bar{y}) \quad \forall x \in S,$$

что показывает, что $\bar{y} \in Y(\bar{\lambda})$ и

$$w(\bar{\lambda}) = f(\bar{y})+\bar{\lambda}g(\bar{y}).$$

Тогда $\forall l \in L$ можно записать

$$w(\bar{\lambda}+\theta^l d) = f(y^l)+(\bar{\lambda}+\theta^l d)g(y^l) =$$
$$= f(y^l)+\bar{\lambda}g(y^l)+\theta^l dg(y^l) \geqslant f(\bar{y})+\bar{\lambda}g(\bar{y})+\theta^l dg(y^l)$$

и, следовательно,

$$\frac{w(\bar{\lambda}+\theta^l d)-w(\bar{\lambda})}{\theta^l} \geqslant dg(y^l).$$

При $l \to \infty$ ($l \in L$) левая часть стремится к $\delta w(\bar{\lambda}, d)$, а правая — к $dg(\bar{y})$, так что

$$\delta w(\bar{\lambda}, d) \geqslant dg(\bar{y}), \quad \bar{y} \in Y(\bar{\lambda}).$$

II. Заметим теперь, что в условиях теоремы множество $\Gamma = \{g(y) \mid y \in Y(\bar{\lambda})\}$ компактно. Так как выпуклая оболочка компактного множества замкнута, то множество $\operatorname{conv}(\Gamma)$ замкнуто и выпукло. С другой стороны, из свойства 4 вытекает, что $\Gamma \subset \partial w(\bar{\lambda})$, и так как $\partial w(\bar{\lambda})$ есть замкнутое выпуклое множество, то $\operatorname{conv}(\Gamma) \subset \partial w(\bar{\lambda})$.

Чтобы показать, что $\partial w(\bar{\lambda}) \subset \operatorname{conv}(\Gamma)$, приведем к противоречию предположение, что существует такой $\bar{\gamma} \in \partial w(\bar{\lambda})$, что $\bar{\gamma} \notin \operatorname{conv}(\Gamma)$.

Применяя теорему 1 приложения 1, выводим, что существует такой вектор $d = (d_1, \ldots, d_m)$, $d \neq 0$, и такой скаляр α, что

$$dg(y) \geqslant \alpha \qquad \forall y \in Y(\bar{\lambda}),$$

$$d\bar{\gamma} < \alpha.$$

Используя (1), мы можем для этого вектора d найти такой $\bar{y} \in Y(\bar{\lambda})$, что $\delta w(\bar{\lambda}, d) \geqslant dg(\bar{y})$ и, следовательно, $\delta w(\bar{\lambda}, d) \geqslant \alpha$.

Но, с другой стороны,

$$\delta w(\bar{\lambda}, d) = \min_{\gamma \in \partial w(\bar{\lambda})} \{d\gamma\} \leqslant d\bar{\gamma} < \alpha,$$

откуда и получаем противоречие.

Таким образом, действительно $\operatorname{conv}(\Gamma) = \partial w(\bar{\lambda})$, и теорема доказана.

Замечание. Условие 2) теоремы 3 выполняется, например, если $f(x) \to +\infty$ при $\|x\| \to +\infty$ и если существует такое вещественное z, что

$$\forall x \in S, \quad \forall i \in I: \; g_i(x) \geqslant z.$$

2.8. Дифференцируемость дуальной функции. Если функция w дифференцируема в точке $\bar{\lambda}$, то $Y(\bar{\lambda})$ сводится к единственному элементу $\bar{y}$ и $g(\bar{y})$ есть градиент w в точке $\bar{\lambda}$. Для того чтобы этот случай имел место, необходимо и достаточно, чтобы выполнялись условия теоремы 3 в точке $\bar{\lambda}$ и чтобы функция Лагранжа $L(x, \bar{\lambda})$ имела единственный минимум в точке x для $\lambda = \bar{\lambda}$.

Заметим, что единственность минимума в точке x для функции $L(x, \bar{\lambda})$ имеет место, в частности, если функции g_i выпуклы, а f строго выпукла.

Пример. Рассмотрим пример 1 из п. 2.4.

Для любого $\lambda \geqslant 0$ (единственный) минимум функции $f(x) + \lambda g(x)$ достигается в точке $\bar{x}_1 = -\lambda$, $\bar{x}_2 = -\lambda/2$.

Тогда

$$g(\bar{x}) = 2\bar{x}_1 + \bar{x}_2 + 4 = -\frac{5\lambda}{2} + 4.$$

Можно убедиться, что эта величина совпадает с производной функции $w(\lambda) = -5\lambda^2/4 + 4\lambda$.

Покажем, наконец, как решение дуальной задачи позволяет определить седловую точку (если она существует) и получить также оптимальное решение исходной задачи.

2.9. Свойство 5. *Пусть* $\bar{\lambda} \geqslant 0$ — *оптимальное решение дуальной задачи. Тогда*:

a) *если задача* (P) *имеет седловую точку, то существует такое решение* x^* *задачи* (P), *что* $(x^*, \bar{\lambda})$ *есть седловая точка (тогда* x^* *есть оптимальное решение задачи* (P));

b) *если задача* (P) *имеет седловую точку и если* $L(x, \bar{\lambda})$ *имеет единственный минимум по* x — *пусть в точке* $\bar{x} \in S$, — *то точка* $(\bar{x}, \bar{\lambda})$ *обязательно является седловой и* $\bar{x}$ — *оптимальное решение задачи* (P);

c) *если предположить, что* w *дифференцируема в точке* $\bar{\lambda}$, *и если* $\bar{x}$ — (*единственный*) *минимум* x *в* $L(x, \bar{\lambda})$, *то* (P) *имеет седловую точку и* $(\bar{x}, \bar{\lambda})$ *есть такая седловая точка.*

Доказательство.

a) По определению $w(\bar{\lambda})$ существует такой $\bar{x} \in S$, что

$$w(\bar{\lambda}) = f(\bar{x}) + \bar{\lambda} g(\bar{x}),$$

причем

$$w(\bar{\lambda}) = f(\bar{x}) + \bar{\lambda} g(\bar{x}) \leqslant f(x) + \bar{\lambda} g(x) \qquad \forall x \in S.$$

Пусть (x^*, λ^*) — седловая точка (она существует по условию). Покажем, что $(x^*, \bar{\lambda})$ — также седловая точка. Так как x^* — оптимальное решение задачи (P), то $w(\bar{\lambda}) = f(x^*)$. Отсюда следует, что

$$w(\bar{\lambda}) = f(x^*) \leqslant f(x^*) + \bar{\lambda} g(x^*),$$

откуда $\bar{\lambda} g(x^*) \geqslant 0$.

Так как, с другой стороны, $\bar{\lambda} \geqslant 0$ и $g(x^*) \leqslant 0$, то $\bar{\lambda} g(x^*) = 0$, откуда следует, что $\forall i$: $\bar{\lambda}_i g_i(x^*) = 0$.

Наконец, соотношение

$$w(\bar{\lambda}) = f(x^*) = f(x^*) + \bar{\lambda} g(x^*) \leqslant f(x) + \bar{\lambda} g(x) \quad \forall x \in S$$

показывает, что x^* — также точка минимума по x для $L(x, \bar{\lambda})$. Согласно теореме 2 гл. 5 (п. 2.1), точка $(x^*, \bar{\lambda})$ — седловая.

b) Предположим, что $(\bar{x}, \bar{\lambda})$ — не седловая точка. Повторяя приведенное выше доказательство, получаем, что существует такое решение x^* задачи (P), что $(x^*, \bar{\lambda})$ — седловая точка. Очевидно, $x^* \neq \bar{x}$. Но, так как x^* — также точка минимума по x для $L(x, \bar{\lambda})$, то отсюда следует противоречие с тем фактом, что минимум по x для $L(x, \bar{\lambda})$ единствен.

c) Если дуальная функция w дифференцируема в точке $\bar{\lambda}$, то $L(x, \bar{\lambda})$ имеет по x единственный минимум — назовем его $\bar{x}$ — и, вследствие свойства 4, вектор $g(\bar{x})$ есть градиент функции w в точке $\bar{\lambda}$. Так как $\bar{\lambda}$ есть оптимум для w при ограничениях $\lambda \geqslant 0$,

то мы, таким образом, должны иметь

$$\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0,$$
$$\bar{\lambda}_i = 0 \Rightarrow g_i(\bar{x}) \leqslant 0.$$

Вследствие теоремы 2 гл. 5 (п. 2.1) точка $(\bar{x}, \bar{\lambda})$ — седловая.

Во многих случаях дуальную задачу (D) решить легче, чем исходную. Если седловая точка существует, то решение задачи (P) может быть, таким образом, с выгодой заменено решением задачи (D). Заметим, однако, что даже если седловой точки нет, решение задачи (P) может быть заметно упрощено, если использовать информацию, полученную при решении дуальной задачи.

Если выполняются условия предложений 5b) или 5c) и если $\bar{\lambda}$ известно, то мы *непосредственно* получаем $\bar{x}$ с помощью минимизации без ограничений, примененной к $L(x, \bar{\lambda})$. Это полностью соответствует первоначальной цели, которая состоит в преобразовании задачи в задачу без ограничений. Так как, вообще говоря, $\bar{\lambda}$ не известно заранее, то мы определяем его, максимизируя дуальную функцию итеративным методом. Если λ^k — последовательность получаемых таким образом точек $(\lambda^k \to \bar{\lambda})$ и если условия предложений 5b) или 5c) выполнены в некоторой окрестности точки $\bar{\lambda}$, то последовательность соответствующих точек x^k сходится к $\bar{x}$. Решение исходной задачи заменяется, таким образом, решением последовательности задач оптимизации без ограничений.

2.10. Скачки двойственности и недифференцируемость дуальной функции в оптимуме. Непосредственным и важным следствием свойства 5c) является следующее:

Если задача (P) не имеет седловой точки, то дуальная функция w необходимо недифференцируема в оптимуме.

Таким образом, существует много задач, для которых дуальная функция не имеет точной производной в искомой точке $\bar{\lambda}$ (в оптимуме для (D)) и для которых необходимо включать в работу алгоритмы, приспособленные к оптимизации выпуклых или вогнутых функций, которые не всюду дифференцируемы (например, алгоритмы субградиента, см. гл. 4, § 3).

2.11. Получение приближенных решений исходной задачи с помощью решения двойственной задачи. Если свойство 5b) нельзя применить в связи с тем, что (P) не допускает седловой точки, или с тем, что минимум функции $L(x, \bar{\lambda})$ по x не единствен (тогда дуальная функция недифференцируема в оптимуме $\bar{\lambda}$), то, как мы сейчас увидим, можно весьма общим образом получить хорошие приближенные решения исходной задачи (P) с помощью решения дуальной задачи. Это наблюдение является следствием свойства 6.

Свойство 6. *Для произвольного $\lambda \geqslant 0$ обозначим через $\bar{x}(\lambda) \in Y(\lambda)$ точку минимума функции $L(x, \lambda)$ по x. Тогда $\bar{x}(\lambda)$ есть*

глобальный оптимум следующей (возмущенной) задачи:

$$f(x) \to \min,$$
$$g_i(x) \leqslant g_i(\bar{x}(\lambda)), \quad i \in I, \qquad (\widehat{P})$$
$$x \in S.$$

Доказательство. Введем обозначение $\tilde{g}_i(x) = g_i(x) - g_i(\bar{x}(\lambda))$ и рассмотрим задачу

$$f(x) \to \min,$$
$$\tilde{g}_i(x) \leqslant 0, \quad i \in I, \qquad (\tilde{P})$$
$$x \in S.$$

Можно проверить, что $\bar{x}(\lambda)$ — точка минимума по x для функции $\tilde{L}(x, \lambda) = f(x) + \sum_{i \in I} \lambda_i \tilde{g}_i(x)$.

С другой стороны, условие $\tilde{g}(\bar{x}(\lambda)) = 0$ показывает (см. теорему 2 гл. 5), что $(\bar{x}(\lambda), \lambda)$ есть седловая точка задачи $(\tilde{P})$ и, вследствие этого, $\bar{x}(\lambda)$ есть глобальный оптимум для задачи $(\tilde{P})$ и для задачи $(\widehat{P})$.

Предположим тогда, что мы ищем оптимум $\bar{\lambda}$ дуальной функции w (недифференцируемой в точке $\bar{\lambda}$) с помощью алгоритма субградиента. Следовательно, строим последовательность λ^k, сходящуюся к $\bar{\lambda}$. Точки λ_k, полученные таким образом и достаточно близкие к $\bar{\lambda}$, обладают тем свойством, что модули субградиентов $g(\bar{x}(\lambda^k))$ малы, иначе говоря, что составляющие $g_i(\bar{x}(\lambda^k))$, являющиеся положительными, обязательно близки к нулю. Это показывает, что точки $\bar{x}(\lambda_k)$ доставляют хорошие аппроксимации оптимума (P), поскольку вследствие свойства 6 они являются оптимальными (глобальными) решениями задач, отличающихся от (P) только (малым) возмущением правых частей ограничений.

Эта особенность дуальных методов особенно интересна во многих практических приложениях, в которых можно удовлетвориться решениями, удовлетворяющими ограничениям «с точностью до ε».

§ 3. Классические лагранжевы методы

Эти методы состоят в решении дуальной задачи (D) с существенным использованием вогнутости дуальной функции w и свойства 4 из п. 2.7, позволяющего вывести субградиенты (и сам градиент в дифференцируемом случае) с помощью вычисления значения $w(\lambda)$. Может быть использован любой алгоритм градиента (см. гл. 4, § 2) или субградиента.

Если существует седловая точка (в частности, в выпуклом случае) и если точка минимума по x функции Лагранжа $L(x, \lambda^*)$ единственна для $\lambda = \lambda^*$ (в оптимуме дуальной задачи), то свойство 5 показывает, что решение дуальной задачи позволяет получить оптимальное решение исходной задачи. В случае, если седловая точка не существует (и есть скачок двойственности), решение

дуальной задачи доставляет, с одной стороны, приближенные решения исходной задачи (свойство 6) и, с другой стороны, оценку снизу оптимального решения исходной задачи (по свойству 1).

Основные методы, использующие лагранжеву двойственность, перечислены ниже.

3.1. Алгоритмы Удзавы (1958) и Эрроу — Гурвица (1958). Алгоритм Удзавы [58] использует классический метод градиента (в недифференцируемом случае — субградиента) для решения дуальной задачи. На каждом этапе минимизируется функция Лагранжа. В общих чертах метод выглядит следующим образом.

a) Отправиться из точки $\lambda^0 \geqslant 0$.

b) На k-м этапе итерации мы находимся в точке λ^k. Вычислить величину

$$w(\lambda^k) = \min_{x \in S} \{f(x) + \lambda^k g(x)\} = f(x^k) + \lambda^k g(x^k).$$

c) Вектор $g(x^k) = [g_i(x^k)]_{i=\overline{1,m}}$ есть субградиент функции w в точке λ^k (градиент в дифференцируемом случае). Определим λ^{k+1} равенством

$$\lambda^{k+1} = \Pr_{\mathbf{R}^{m+}} \{\lambda^k + \rho_k g(x^k)\},$$

где ρ_k — шаг перемещения на этапе k.

Если выполнен тест остановки, то СТОП. Иначе выполнить $k \leftarrow k+1$ и вернуться к b).

В этом алгоритме шаг перемещения ρ_k может быть выбран априори: тогда речь идет о методе градиента или субградиента с определенным заранее шагом, примененном к дуальной функции (более подробно о стратегиях выбора ρ_k см. гл. 4 § 3).

Эти шаги могут равным образом быть выбраны так, чтобы максимизировать $w(\lambda)$, выходя из λ^k в направлении $g(x^k)$; тогда речь идет о методе наибыстрейшего спуска, примененном к дуальной задаче (в последнем случае при одномерных минимизациях может оказаться интересным использование информации о производных, полученной при переходе к градиенту функции w).

Метод Эрроу — Гурвица [2] очень близок предыдущему. Разница состоит в том, что на каждом шаге k мы не ищем минимум по x функции Лагранжа $L(x, \lambda^k)$, а удовлетворяемся перемещением (часто на заранее определенный шаг) в направлении $-\nabla_x L(x^{k-1}, \lambda^k)$. Этот метод состоит, таким образом, в перемещениях на заранее определенные шаги поочередно в пространстве исходных переменных и в пространстве дуальных переменных.

Если дуальная функция плохо обусловлена, то можно также использовать квазиньютоновский метод для оптимизации дуальной задачи (см. п. 5.2.2).

3.2. Алгоритм Данцига (1959). Его принцип основан на следующем замечании: если в задаче (P) из § 2 множество S конечно, то дуальная задача (D) эквивалентна задаче линейного

программирования:

$$z \to \max_{\lambda, z},$$

$$f(x) + \sum_{i \in I} \lambda_i g_i(x) \geqslant z \qquad \forall x \in S, \qquad (PL)$$

$$\lambda \geqslant 0,$$

в которой каждой точке множества S соответствует ограничение.

Очевидно, если множество S бесконечно, то задача содержит бесконечно много ограничений и не может быть решена явным образом.

Однако даже в этом случае ее можно решить приближенно методом обобщенного линейного программирования («порождением ограничений») следующим образом (см. гл. 8 § 1).

а) Предположим, что на текущем этапе уже порождены K ограничений, соответствующих K точкам $x^1, x^2, \ldots, x^K$.

Решаем задачу линейного программирования

$$z \to \max_{\lambda, z},$$

$$f(x^k) + \sum_{i \in I} \lambda_i g_i(x^k) \geqslant z, \qquad k = 1, \ldots, K, \qquad PL(K)$$

$$\lambda \geqslant 0,$$

что приводит к оптимальному решению λ^{K+1} со значением z^{K+1}.

b) Вычисляем дуальную функцию

$$w(\lambda^{K+1}) = \min_{x \in S} \left\{ f(x) + \sum_I \lambda_i^{K+1} g_i(x) \right\} = f(x^{K+1}) + \sum_I \lambda_i^{K+1} g_i(x^{K+1}).$$

Тогда могут представиться два случая.

Если $w(\lambda^{K+1}) \geqslant z^{K+1}$, то все закончено: λ^{K+1} есть оптимум дуальной задачи (действительно, точка (λ^{K+1}, z^{K+1}) удовлетворяет тогда всем ограничениям задачи (PL) — даже тем, которые не выражены явно в задаче $PL(K)$).

Если же $w(\lambda^{K+1}) < z^{K+1}$, то добавим к задаче $PL(K)$ ограничение

$$f(x^{K+1}) + \sum_I \lambda_i g_i(x^{K+1}) \geqslant z,$$

соответствующее x^{K+1}, и перейдем к $(K+1)$-му шагу итерации, чтобы решить «расширенную» линейную задачу $PL(K+1)$.

Заметим, что этот алгоритм сводится к применению к дуальной функции задачи (P) метода касательной линеаризации (см. гл. 8, п. 1.6). Тогда сходимость последовательности λ^k к оптимуму λ^* дуальной задачи следует из теоремы 1 гл. 8 (п. 1.6). На каждом шаге k получаем последовательность

$$w(\lambda^k) \leqslant w(\lambda^*) \leqslant z^k,$$

что позволяет прервать вычисления, как только скачок станет меньше некоторой зафиксированной с самого начала величины $\varepsilon > 0$.

Кроме того, если задача строго выпукла (например, если g_i выпуклы, f строго выпукла и S выпукло), то последовательность полученных точек x^k сходится к (единственному) оптимальному решению x задачи (P).

Заметим, что на практике на каждом шаге решается не задача $PL(K)$, а двойственная к ней задача. Алгоритм действует тогда как порождение столбцов и отождествляется с методом барицентрической аппроксимации, описанным в гл. 5, п. 3.6.3.

3.3. Использование лагранжевых методов в невыпуклом случае. Важно заметить, что интерес к лагранжевым методам не ограничивается случаем выпуклых задач. В невыпуклом случае (где функции f и g_i не выпуклы) функция Лагранжа $L(x, \lambda)=f(x)+\lambda g(x)$ не обязана быть выпуклой по x при данном λ. Следовательно, часто оказывается затруднительным получить абсолютный минимум $L(x, \lambda)$ по x и тогда приходится удовлетвориться локальным минимумом $\bar{x}$. Однако если в некоторой окрестности $V(\bar{x}) \subset S$ точки $\bar{x}$ функции $f(x)$ и $g_i(x)$ выпуклы и если неравенство $g(x) \leqslant 0$ имеет решение в $V(\bar{x})$, то мы видим, что можно определить «локальную» седловую точку для задачи

$$f(x) \to \min,$$
$$g(x) \leqslant 0,$$
$$x \in V(\bar{x}),$$

разрешая «локальную» дуальную задачу

$$w(\lambda) \to \max,$$
$$\lambda \geqslant 0,$$

где $w(\lambda)$ определяется равенством

$$w(\lambda) = \min_{x \in V(\bar{x})} \{f(x) + \lambda g(x)\}.$$

(Свойство вогнутости $w(\lambda)$ сохраняется, даже если ограничить x подмножеством $V(\bar{x})$ множества S, что делает возможным определение абсолютного максимума функции $w(\lambda)$.)

Эта «локальная» седловая точка приводит тогда к локальному оптимуму начальной задачи

$$f(x) \to \min,$$
$$g(x) \leqslant 0, \qquad (P)$$
$$x \in S$$

(см. гл. 5, п. 2.7).

Вместе с тем существует немало случаев, в которых — хотя задача (P) и не выпукла (например, по той причине, что множество S не выпукло) — для каждого значения λ тем не менее оказывается возможным определить абсолютный минимум функции $L(x, \lambda)=f(x)+\lambda g(x)$ на S. Тогда в любом случае полученное значение $w(\lambda) = \min_{x \in S} L(x, \lambda)$ есть миноранта оптимума цены. Если мы

располагаем приближенным решением задачи, то это наблюдение позволяет оценить качество этого решения и измерить теоретический скачок от оптимальности. Это исключительно важно во многих сложных задачах, для которых известны только приближенные алгоритмы (эвристические методы).

Для целочисленных задач миноранту, получаемую решением дуальной задачи, можно также использовать как функцию оценки в ходе применения методов типа «разветвленного поиска», которые позволяют найти точное решение исходной задачи (гл. 7, § 2).

Напомним, наконец, что решение дуальной задачи часто может привести к хорошим приближенным решениям исходной задачи — вследствие свойства 6 п. 2.11. Эта особенность лагранжевых методов чрезвычайно полезна на практике.

§ 4. Обобщенные лагранжианы и седловые точки в невыпуклом программировании

4.1. Расширенные лагранжианы: введение. Мы собираемся описать здесь два типа методов, позволяющих заменить решение задачи

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \quad i = 1, \ldots, m, \qquad (P')$$
$$x \in \mathbf{R}^n,$$

решением последовательности задач оптимизации без ограничений.

1) Методы штрафов (§ 1) — например, метод внешних штрафов (см. п. 1.2), где на каждом шаге минимизируется величина

$$\varphi(x, r) = f(x) + rH(x) = f(x) + r \sum_{i=1}^{m} [\max\{g_i(x), 0\}]^2, \quad r > 0.$$

Этот метод представляет то неудобство, что для того, чтобы получить реализуемую точку, нужно заставить r стремиться к $+\infty$; тогда $\varphi(x, r)$ становится плохо обусловленной функцией, для которой методы градиента дают медленную сходимость (см. гл. 4, п. 2.12).

2) Методы, использующие лагранжеву двойственность (§ 3), где (P') заменяется последовательностью задач

$$\min_x \left\{ f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \right\}.$$

Эти методы представляют, в частности, то неудобство, что даже если скачок двойственности отсутствует и если λ^k сходятся к оптимуму λ^* дуальной задачи, то получаемые соответствующие последовательные решения x^k могут не сходиться к решению исходной задачи — за исключением случая, когда $L(x, \lambda)$ имеет в окрестности точки λ^* единственный минимум по x (см. п. 2.9; это последнее обстоятельство имеет место, например, в случае, когда $L(x, \lambda)$

строго выпукло по x). В частности, задачи линейного программирования этим способом не решаются.

В случае ограничений в виде равенств Хестенес [32] и независимо Пауэлл [49] заметили, что эти трудности могут быть частично устранены комбинированием двух подходов (штрафов + двойственности), и предложили решать последовательность задач без ограничений вида

$$\min_x L(x, \lambda, r) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + r \sum_{i=1}^{m} [g_i(x)]^2$$

(λ — число любого знака).

Обобщение функции Хестенеса — Пауэлла на случай ограничений в форме неравенств [51] совершенно очевидно. Достаточно ввести переменные скачков $s_i \geqslant 0$ и переписать (P') в виде

$$\begin{aligned} f(x) &\to \min, \\ g_i(x) + s_i &= 0, \\ s_i &\geqslant 0, \\ x &\in \mathbf{R}^n. \end{aligned}$$

Тогда на каждом шаге нужно решать задачу

$$\min_{\substack{x \in \mathbf{R}^n \\ s \geqslant 0}} \left\{ f(x) + \sum_i \lambda_i (g_i(x) + s_i) + r \sum_i [g_i(x) + s_i]^2 \right\}$$

(λ — числа любого знака).

Заметим, что в этом выражении минимизация относительно $s \geqslant 0$ при фиксированном x может быть проделана явным образом отдельно для каждого s_i. Действительно, члены, содержащие s_i, суть

$$r s_i^2 + (2 r g_i(x) + \lambda_i) s_i,$$

и если предполагать, что $r > 0$, то минимум в области $s_i \geqslant 0$ достигается

— либо при $s_i = -g_i(x) - \lambda_i/(2r)$ (если $g_i(x) \leqslant -\lambda_i/(2r)$),

— либо при $s_i = 0$ (если $g_i(x) \geqslant -\lambda_i/(2r)$).

Остается тогда задача: на каждом шаге минимизировать относительно x функцию

$$L(x, \lambda, r) = f(x) + \sum_{i=1}^{m} G(g_i(x), \lambda_i, r),$$

где

$$G(g_i(x), \lambda_i, r) = \begin{cases} -\dfrac{\lambda_i^2}{4r}, & r > 0,\ g_i(x) \leqslant -\dfrac{\lambda_i}{2r}, \\ \lambda_i g_i(x) + r [g_i(x)]^2, & r > 0,\ g_i(x) \geqslant -\dfrac{\lambda_i}{2r}, \\ \lambda_i g_i(x), & r = 0,\ \lambda_i \geqslant 0, \\ -\infty, & r = 0,\ \lambda_i < 0, \end{cases}$$

называется «расширенным лагранжианом» (Рокафеллар, [51]).

Можно предложить другие типы расширенных лагранжианов, основанные на той же идее (комбинация классического лагранжиана с *функциями штрафов*).

Например, метод множителей Накаямы и др. [47] использует функцию

$$L(x, \lambda, t) = f(x) + \sum_{i=1}^{m} \begin{cases} t\,[g_i(x)]^2 + \lambda_i g_i(x), & g_i(x) \geqslant 0 \\ \dfrac{\lambda_i^2 g_i(x)}{\lambda_i - t g_i(x)}, & g_i(x) < 0 \end{cases}$$

(определенную для $\lambda \geqslant 0$, $t > 0$), которая комбинирует одновременно и методы внутренних штрафов, и методы внешних штрафов.

Можно указать также Пьера и Лёве [48], которые используют функцию

$$L(x, \lambda, w) = f(x) + \sum_{i=1}^{m} \begin{cases} \lambda_i g_i(x) + w\,[g_i(x)]^2, & \lambda_i > 0 \\ w g_i(x) \left[\dfrac{g_i(x) + |g_i(x)|}{2}\right], & \lambda_i = 0 \end{cases}$$

и сообщают много случаев вычислений.

Использование расширенного лагранжиана может рассматриваться как улучшение методов штрафов, которое позволяет избежать использования слишком больших коэффициентов штрафов.

Это легко видеть в случае ограничений в виде равенств для функции Хестенеса — Пауэлла:

$$L(x, \lambda, r) = f(x) + \sum_i \lambda_i g_i(x) + r \sum_i [g_i(x)]^2.$$

Действительно, в точке оптимума (x^*, λ^*) градиент в точке x^* функции $L(x, \lambda^*, r)$ равен нулю для всех значений r (ибо $g_i(x^*) = 0$).

Напротив, градиент функции

$$f(x) + r \sum_i [g_i(x)]^2$$

в точке x^* равен $\nabla f(x^*)$ (поскольку $g_i(x^*) = 0$) и, таким образом, вообще говоря, отличен от нуля для любого значения r. Именно поэтому и приходится выбирать слишком большие значения r, чтобы достаточно хорошо приблизить x^*, и еще с тем неудобством, что функция штрафа становится очень плохо обусловленной.

В дальнейшем (см. п. 4.7) мы увидим, что расширенные лагранжианы еще и потому представляют исключительно большой интерес, что они позволяют распространить теорию двойственности на невыпуклые задачи.

В настоящее время общепризнано, что алгоритмы оптимизации, основанные на использовании расширенных лагранжианов, являются наиболее эффективными общими методами решения задач математического программирования с сильно нелинейными функциями стоимости и ограничениями (см. § 5 об анализе сходимости).

4.2. Лагранжевы представления задачи оптимизации с ограничениями. Рассмотрим для начала задачу вида

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \qquad i = 1, \ldots, m, \\ x \in \mathbf{R}^n. \end{gathered} \tag{P'}$$

Чтобы следующие рассуждения были более понятны, важно подчеркнуть связь, существующую между «обычной» функцией Лагранжа

$$L(x, \lambda) = \begin{cases} f(x) + \sum\limits_{i=1}^{m} \lambda_i g_i(x), & \lambda \geqslant 0, \\ -\infty, & \lambda \not\geqslant 0 \end{cases}$$

и «расширенной» функцией стоимости

$$\bar{f}(x) = \begin{cases} f(x), & \text{если } x \in \mathbf{R}^n \text{ удовлетворяет условиям } (P'), \\ +\infty & \text{в противном случае.} \end{cases}$$

Действительно, имеем

$$\max_{\lambda \in \mathbf{R}^m} L(x, \lambda) = \bar{f}(x) \qquad \forall x \in \mathbf{R}^n. \tag{8}$$

(Важное замечание: для фиксированного x область определения $L(x, \lambda)$ расширяется на все пространство $\mathbf{R}^m$ приписыванием значения $-\infty$ для всех векторов λ, не являющихся неотрицательными.)

Решение задачи (P'), эквивалентное минимизации функции $\bar{f}(x)$ на $\mathbf{R}^n$, сводится, следовательно, к решению задачи (так называемой исходной задачи)

$$\min_{x \in \mathbf{R}^n} \max_{\lambda \in \mathbf{R}^m} \{L(x, \lambda)\}.$$

Рассмотрим теперь в полной общности **задачу, поставленную** в виде

$$\begin{gathered} \bar{f}(x) \to \min, \\ x \in X \end{gathered} \tag{$\bar{P}$}$$

(X — вещественное векторное пространство; обычно $X \equiv \mathbf{R}^n$).

Определение 1. Пусть дано вещественное векторное пространство Λ и вещественная функция $\bar{L}(x, \lambda)$ на $X \times \Lambda$; функция $\bar{L}(x, \lambda)$ называется *лагранжевым представлением* задачи $(\bar{P})$ в том и только том случае, если:

1) $\bar{L}(x, \lambda)$ является замкнутой вогнутой функцией от λ для любого x;

2) $\bar{f}(x) = \max\limits_{\lambda \in \Lambda} \bar{L}(x, \lambda) \qquad \forall x \in X.$

Эта последняя формула, обобщающая (8), означает только, что $\bar{f}$ выражается как поточечный максимум семейства функций

$$\{\bar{L}(x, \lambda) \mid \lambda \in \Lambda\}.$$

Элементы семейства Λ называются *обобщенными множителями Лагранжа.*

Функция $\bar{L}$ называется *обобщенным лагранжианом* задачи $(\bar{P})$.

Для каждого фиксированного $\lambda \in \Lambda$ задача минимизации $\bar{L}(x, \lambda)$ на X может рассматриваться как некоторое (лагранжево) представление, оценивающее снизу данную задачу $(\bar{P})$. Действительно, поскольку $\forall \lambda \in \Lambda$ выполняется соотношение

$$\bar{f}(x) \geqslant \bar{L}(x, \lambda) \quad \forall x,$$

то имеем $\forall \lambda \in \Lambda$:

$$\min_{x \in X} \bar{f}(x) \geqslant \min_{x \in X} \{\bar{L}(x, \lambda)\} = \bar{w}(\lambda), \tag{9}$$

где (*обобщенная*) *дуальная функция* $\bar{w}(\lambda)$ определяется равенством

$$\bar{w}(\lambda) = \min_{x \in X} \{\bar{L}(x, \lambda)\}.$$

Заметим, что (9) есть не что иное, как обобщение минорантного свойства дуальной функции (см. свойство 1 п. 2.1).

С другой стороны, (9) выражает $\bar{w}(\lambda)$ как (поточечный) минимум семейства вогнутых функций $\{\bar{L}(x, \cdot) \mid x \in X\}$. Отсюда следует, что $\bar{w}(\lambda)$ есть вогнутая функция от λ, что обобщает свойство 2 п. 2.2.

Точка λ^* называется *седловым множителем* для $(\bar{P})$ относительно функции Лагранжа $\bar{L}$, если

$$\min_{x \in X} \bar{f}(x) = \min_{x \in X} \{\bar{L}(x, \lambda^*)\} = \bar{w}(\lambda^*).$$

Определить седловой множитель λ^* можно с помощью решения (*обобщенной*) *дуальной задачи*

$$\begin{aligned} &\bar{w}(\lambda) \to \max, \\ &\lambda \in \Lambda. \end{aligned} \tag{$\bar{D}$}$$

Действительно, если существует седловая точка (x^*, λ^*), то

$$\min (\bar{P}) = \min_{x \in X} \max_{\lambda \in \Lambda} \{\bar{L}(x, \lambda)\} = \max_{\lambda \in \Lambda} \min_{x \in X} \{\bar{L}(x, \lambda)\} = \max (\bar{D}).$$

Иначе говоря, существование седловой точки есть необходимое и достаточное условие перестановочности операторов min и max.

Если λ^* известны, то можно заменить задачу $(\bar{P})$ на (вообще говоря, более простую) задачу, состоящую в минимизации $\bar{L}(x, \lambda^*)$ по X.

Если на практике точка λ^* априори неизвестна, то мы применим к $(\bar{D})$ итеративный метод, чтобы породить последовательность $\{\lambda^k\}$, сходящуюся к λ^*. Очевидно, как и в классических лагранжевых методах, такой подход интересен только в случае, когда последовательность $\{x^k\}$, определенная условием

$$\bar{L}(x^k, \lambda^k) = \min_{x \in X} \bar{L}(x, \lambda^k) \quad \forall k,$$

сходится к оптимуму x^* задачи $(\bar{P})$, иначе говоря, если существует седловая точка для обобщенного лагранжиана $\bar{L}$.

В связи с этим мы интересуемся характеризацией седловых точек с помощью обобщения понятия функции возмущения и интерпретации седловых множителей в терминах опорных гиперплоскостей этих обобщенных функций возмущения.

Мы увидим также, как можно эффективно строить практически применимые обобщенные лагранжианы, допускающие седловую точку, даже если они применяются к невыпуклым задачам.

4.3. Представления задачи с помощью возмущений. Рассмотрим снова задачу $(\bar{P}) = \{\min \bar{f}(x),\ x \in X\}$.

Определение 2. Пусть дано вещественное векторное пространство Y; вещественная функция $\bar{F}(x, y)$ на $X \times Y$ называется *представлением задачи* (P) *с помощью возмущений* тогда и только тогда, когда:

1) для любого $x \in X$ функция $\bar{F}(x, y)$ замкнута и выпукла по y;

2) $\bar{F}(x, 0) = \bar{f}(x)\ \forall x \in X$.

Возмущенная задача, соответствующая некоторому вектору $y \in Y$, есть, по определению, задача

$$\bar{F}(x, y) \to \min, \qquad (\bar{P}_y)$$
$$x \in X.$$

Пример. Для задачи (P'): $\{\min f(x),\ g_i(x) \leqslant 0,\ x \in \mathbf{R}^n\}$ можно выбрать представление с помощью возмущений в виде

$$F(x, y) = \begin{cases} f(x), & \text{если } g_i(x) \leqslant y_i \quad \forall i, \\ +\infty & \text{в противном случае.} \end{cases}$$

В гл. 5, п. 2.4 мы видели, что это представление с помощью возмущений соответствует «обычному» лагранжеву представлению задачи (P).

Как мы сейчас увидим, существует непосредственная связь между лагранжевыми представлениями и представлениями с помощью возмущений одной и той же задачи $(\bar{P})$.

Теорема 4 [53]. *Пусть Y и Λ — вещественные векторные пространства в двойственности (скалярное произведение обозначается точкой $\cdot$). Тогда:*

1) *если $\bar{F}(x, y)$ — представление задачи $(\bar{P})$ с помощью возмущений (с функцией $\bar{F}$, замкнутой и выпуклой по y), то из него с помощью формулы*

$$\bar{L}(x, \lambda) = \min_{y \in Y} \{\bar{F}(x, y) + \lambda \cdot y\}$$

можно вывести лагранжево представление $\bar{L}(x, y)$;

2) *обратно, если $\bar{L}(x, y)$ — лагранжево представление задачи $(\bar{P})$ (с функцией $\bar{L}$, замкнутой и вогнутой по λ), то можно вывести из нее представление с помощью возмущений $\bar{F}(x, y)$ формулой*

$$\bar{F}(x, y) = \max_{\lambda \in \Lambda} \{\bar{L}(x, \lambda) - \lambda \cdot y\}.$$

Доказательство.

1) Пусть $\bar{F}(x, y)$ — представление задачи $(\bar{P})$ с помощью возмущений и пусть некоторое $x \in X$ фиксировано.

Определим $\bar{L}(x, \lambda)$ формулой

$$\bar{L}(x, \lambda) = \min_{y \in Y} \{\bar{F}(x, y) + \lambda \cdot y\} \quad \forall \lambda$$

и рассмотрим функцию φ, определенную равенством

$$\varphi(\lambda) = \max_{y \in Y} \{\lambda \cdot y - \bar{F}(x, y)\} \quad \forall \lambda.$$

Заметим, что φ — сопряженная функция к функции $\bar{F}(x, y)$, рассматриваемой как функция переменного y (см. гл. 1, п. 3.4). Следовательно, это — замкнутая выпуклая функция. Вследствие равенства

$$\bar{L}(x, \lambda) = -\varphi(-\lambda)$$

мы получаем, что $\bar{L}(x, \lambda)$ — замкнутая вогнутая функция λ.

Теперь для любого $\lambda \in \Lambda$ можно написать

$$\bar{L}(x, \lambda) \leqslant \bar{F}(x, y) + \lambda \cdot y \quad \forall y$$

и взять, в частности, $y = 0$:

$$\bar{L}(x, \lambda) \leqslant \bar{F}(x, 0) \quad \forall \lambda \in \Lambda.$$

Предположим, кроме того, что $\bar{F}(x, 0) < +\infty$, иначе говоря, что точка $y = 0$ принадлежит эффективной области $\bar{F}(x, y)$, рассматриваемой (при данном x) как функция переменного y. Так как $\bar{F}$ замкнута и выпукла, то она (ее график) имеет опорную гиперплоскость в точке $[0, \bar{F}(x, 0)]$. Пусть

$$z(y) = \bar{F}(x, 0) - \bar{\lambda} \cdot y$$

есть уравнение этой гиперплоскости. Тогда

$$\bar{F}(x, y) \geqslant \bar{F}(x, 0) - \bar{\lambda} \cdot y \quad \forall y,$$

следовательно,

$$\bar{F}(x, 0) \leqslant \bar{F}(x, y) + \bar{\lambda} \cdot y \quad \forall y,$$

откуда выводим

$$\bar{F}(x, 0) = \min_{y \in Y} \{\bar{F}(x, y) + \bar{\lambda} \cdot y\} = \bar{L}(x, \bar{\lambda}).$$

Следовательно, можно написать

$$\bar{L}(x, \bar{\lambda}) = \max_{\lambda \in \Lambda} \bar{L}(x, \lambda) = \bar{F}(x, 0) = \bar{f}(x).$$

Если же $\bar{F}(x, 0) = +\infty$, то с помощью аналогичного рассуждения можно показать, что

$$\max_{\lambda \in \Lambda} \bar{L}(x, \lambda) = \bar{F}(x, 0) = \bar{f}(x) = +\infty.$$

Отсюда следует, что $\bar{L}(x, \lambda)$ есть лагранжево представление задачи $(\bar{P})$.

2) Обратно, пусть $\overline{L}(x, \lambda)$ — лагранжево представление задачи $(\overline{P})$ и пусть некоторое $x \in X$ фиксировано.

Рассмотрим функцию

$$\psi(y) = \max_{\lambda \in \Lambda} \{\lambda \cdot y - (-\overline{L}(x, \lambda))\}.$$

Заметим, что $\psi(y)$ — функция сопряженная к $(-\overline{L}(x, \lambda))$, рассматриваемой как функция переменного λ. Следовательно это — замкнутая выпуклая функция (см. гл. 1, п. 3.4), и поскольку

$$\overline{F}(x, y) = \psi(-y),$$

то отсюда выводим, что $\overline{F}(x, y)$ — замкнутая выпуклая функция y.

Кроме того, имеем также

$$\overline{F}(x, 0) = \max_{\lambda \in \Lambda} \{\overline{L}(x, \lambda)\} = \overline{f}(x) \qquad \forall x,$$

что и доказывает, что $\overline{F}$ — представление задачи $(\overline{P})$ с помощью возмущений.

Тогда мы обобщаем понятие функции возмущений следующим образом.

Определение 3. Если дано представление $\overline{F}(x, y)$ задачи $(\overline{P})$ с помощью возмущений, то соответствующей (*обобщенной*) *функцией возмущений* называется функция $\overline{\Phi}$, определяемая формулой

$$\overline{\Phi}(y) = \min(\overline{P}_y) = \min_{x \in X} \overline{F}(x, y) \qquad \forall y \in Y.$$

(Заметим, что для $y = 0$ имеем $\overline{\Phi}(0) = \min(\overline{P})$.)

Пусть $\overline{F}(x, y)$ — представление задачи $(\overline{P})$ с помощью возмущений и пусть $\overline{L}(x, \lambda)$ — соответствующее лагранжево представление.

Рассмотрим функции

$$\overline{\Phi}(y) = \min_{x \in X} \overline{F}(x, y),$$

$$\overline{w}(\lambda) = \min_{x \in X} \overline{L}(x, \lambda).$$

Из соотношения

$$\overline{L}(x, \lambda) = \min_{y \in Y} \{\overline{F}(x, y) + \lambda \cdot y\}$$

сразу вытекает, что $\overline{\Phi}(y)$ и $\overline{w}(\lambda)$ связаны соотношением

$$\overline{w}(\lambda) = \min_{y \in Y} \{\overline{\Phi}(y) + \lambda \cdot y\}. \tag{10}$$

Следует заметить также, что $\overline{w}(\lambda)$ есть (с точностью до знака) функция, сопряженная к $\overline{\Phi}$ (см. гл. 1, п. 3.4).

Равным образом заметим, что, полагая $y = 0$ в формуле (10), мы приходим к соотношению

$$\overline{w}(\lambda) \leqslant \overline{\Phi}(0) = \min(\overline{P}) \qquad \forall \lambda.$$

Сейчас мы увидим, что соотношение (10) позволяет немедленно обобщить свойство 4 п. 2.7 (субградиент) в следующей форме.

Свойство 4′. *Пусть для данного* $\bar{\lambda} = \lambda'$ *элемент* $x^0 \in X$ *удовлетворяет условию*

$$\bar{w}(\bar{\lambda}) = \min_{x \in X} \bar{L}(x, \bar{\lambda}) = \bar{L}(x^0, \lambda)$$

и пусть y^0 *выбран так, что*

$$\bar{F}(x^0, y^0) + \bar{\lambda} \cdot y^0 = \min_{y \in Y} \{\bar{F}(x^0, y) + \bar{\lambda} \cdot y\}.$$

Тогда

$$\bar{w}(\bar{\lambda}) = \bar{\Phi}(y^0) + \bar{\lambda} \cdot y^0 = \min_{y \in Y} \{\bar{\Phi}(y) + \bar{\lambda} \cdot y\}$$

и y^0 *есть субградиент функции* $\bar{w}$ *в точке* $\bar{\lambda}$.

Доказательство. Имеем

$$\bar{F}(x^0, y^0) + \bar{\lambda} \cdot y^0 = \bar{L}(x^0, \bar{\lambda}) = \bar{w}(\bar{\lambda}),$$

с другой стороны, для произвольного x имеем

$$\bar{w}(\bar{\lambda}) = \bar{L}(x^0, \bar{\lambda}) \leqslant \bar{L}(x, \bar{\lambda}),$$

$$\bar{L}(x, \bar{\lambda}) = \min_{y \in Y} \{\bar{F}(x, y) + \bar{\lambda} \cdot y\} \leqslant \bar{F}(x, y^0) + \bar{\lambda} \cdot y^0.$$

Следовательно, можно написать

$$\bar{F}(x^0, y^0) + \bar{\lambda} \cdot y^0 \leqslant \bar{F}(x, y^0) + \bar{\lambda} \cdot y^0 \qquad \forall x \in X,$$

что показывает, что x^0 удовлетворяет условию

$$\bar{F}(x^0, y^0) = \min_{x \in X} \bar{F}(x, y^0) = \bar{\Phi}(y^0).$$

Тогда, используя (10), видим, что

$$\bar{w}(\bar{\lambda}) = \bar{\Phi}(y^0) + \bar{\lambda} \cdot y^0 = \min_{y \in Y} \{\bar{\Phi}(y) + \bar{\lambda} \cdot y\}. \tag{11}$$

Покажем, наконец, что y^0 — субградиент $\bar{w}$ в точке $\bar{\lambda}$.

Для всех $\lambda' \neq \bar{\lambda}$ вследствие (10) имеем

$$\bar{w}(\lambda') = \min_{y \in Y} \{\bar{\Phi}(y) + \lambda' \cdot y\} \leqslant \bar{\Phi}(y^0) + \lambda' \cdot y^0,$$

откуда выводим, что

$$\bar{w}(\lambda') - \bar{w}(\bar{\lambda}) \leqslant (\lambda' - \bar{\lambda}) \cdot y^0,$$

что и доказывает рассматриваемое свойство.

Для иллюстрации этого результата можно сослаться на п. 4.4, где мы показываем, как в случае «обычного» лагранжиана мы снова получаем свойство 4 п. 2.7.

Теперь можно установить следующий результат, являющийся непосредственным обобщением теоремы 4 гл. 5.

Теорема 5. *Необходимое и достаточное условие того, что точка* λ^0 *есть седловой множитель, состоит в том, что*

$$\overline{\Phi}(y) \geqslant \overline{\Phi}(0) - \lambda^0 \cdot y \qquad \forall y \in Y. \tag{12}$$

Доказательство. Условие необходимо:
если λ^0 — седловой множитель, то $\overline{w}(\lambda^0) = \overline{\Phi}(0)$.
С другой стороны, используя (10), получаем

$$\overline{w}(\lambda^0) \leqslant \overline{\Phi}(y) + \lambda^0 \cdot y \qquad \forall y \in Y,$$

и отсюда следует (12).
Условие достаточно, поскольку при этом условии

$$\overline{\Phi}(0) \leqslant \overline{\Phi}(y) + \lambda^0 \cdot y \qquad \forall y \in Y,$$

что позволяет утверждать, что в соотношении

$$\overline{w}(\lambda^0) = \min_{y \in Y} \{\overline{\Phi}(y) + \lambda^0 \cdot y\}$$

минимум достигается для $y = 0$ и

$$\overline{w}(\lambda^0) = \overline{\Phi}(0) = \min(\overline{P}).$$

4.4. Пример 1: «обычный» лагранжиан.
Рассмотрим представление с помощью возмущений:

$$F(x, y) = \begin{cases} f(x), & \text{если } g_i(x) \leqslant y_i, \quad i = 1, \ldots, m, \\ +\infty & \text{в противном случае} \end{cases}$$

для задачи

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \qquad i = 1, \ldots, m, \\ x \in \mathbf{R}^n, \end{gathered} \tag{P'}$$

где $y \in Y = \mathbf{R}^m$.
Соответствующее лагранжево представление имеет вид

$$L(x, \lambda) = \min_{y \in \mathbf{R}^m} \{F(x, y) + \lambda \cdot y\}.$$

Для любого фиксированного x минимум справа достигается:
— либо для $y_i = g_i(x)$, если $\lambda_i \geqslant 0 \quad \forall i$ (и в этом случае минимум равен $f(x) + \sum_i \lambda_i(x) g_i(x)$);
— либо для $y_i = +\infty$, если $\exists\, \lambda_i < 0$ (и в этом случае минимум равен $-\infty$).
Следовательно,

$$L(x, \lambda) = \begin{cases} f(x) + \sum_i \lambda_i g_i(x), & \text{если } \lambda \geqslant 0, \\ -\infty, & \text{если } \lambda \not\geqslant 0. \end{cases}$$

Таким образом, мы заново получили классическую функцию Лагранжа.

Столь же интересно иллюстрировать на этом примере свойство 4′. Для данного $\lambda = \bar{\lambda} \geqslant 0$ введем такой x^0, что

$$L(x^0, \bar{\lambda}) = \min_x L(x, \bar{\lambda}).$$

Тогда y^0 есть субградиент дуальной функции в точке $\bar{\lambda}$, где y^0 определяется условием

$$F(x^0, y^0) + \bar{\lambda} \cdot y^0 = \min_{y \in \mathbf{R}^m} \{F(x^0, y) + \bar{\lambda} \cdot y\}.$$

Так как $F(x, y) = +\infty$ при $g(x) \not\leqslant y$ и так как $\bar{\lambda} \geqslant 0$, то мы видим, что этот последний минимум достигается при $y^0 = g(x^0)$.

Таким образом, заново получаем свойство 4 п. 2.7.

4.5. Пример 2: «расширенный лагранжиан» Рокафеллара. Рассмотрим задачу

$$\begin{gathered} f(x) \to \min, \\ g_i(x) \leqslant 0, \qquad i = 1, \ldots, m, \qquad (P') \\ x \in \mathbf{R}^n. \end{gathered}$$

«Расширенный лагранжиан» Рокафеллара соответствует, как мы сейчас увидим, представление с помощью возмущений

$$\widehat{F}(\widehat{x}, \widehat{y}) = \begin{cases} f(x), \text{ если } g_i(x) \leqslant y_i \quad \forall i \text{ и } \sum_i y_i^2 \leqslant s, \\ +\infty \text{ в противном случае,} \end{cases}$$

где $\widehat{y} = (y_1, y_2, \ldots, y_m, s) = (y, s) \in \widehat{Y} = \mathbf{R}^{m+1}$.

Следовательно, соответствующее лагранжево представление имеет вид

$$\widehat{L}(x, \widehat{\lambda}) = \min_{\widehat{y} \in \widehat{Y}} \{\widehat{F}(x, \widehat{y}) + \widehat{\lambda}\widehat{y}\} = \min_{y, s} \{\widehat{F}(x, y, s) + \lambda y + rs\},$$

где

$$\widehat{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m, r) = (\lambda, r), \qquad r \in \mathbf{R}.$$

Вычислим $\widehat{L}(x, \widehat{\lambda})$ для фиксированного x и для фиксированных (λ, r).

Если $r < 0$ или если $r = 0$ и $\exists i$, $\lambda_i < 0$, то мы видим, что

$$\widehat{L}(x, \widehat{\lambda}) = -\infty.$$

Если $r = 0$ и $\lambda \geqslant 0$, то мы заново получаем классический лагранжиан:

$$L(x, \lambda) = \widehat{L}(x, \lambda, 0) = f(x) + \sum_i \lambda_i g_i(x).$$

Если $r > 0$ и $\lambda \geqslant 0$, то минимум необходимо достигается при $s = \sum_i y_i^2$.

Тогда получаем

$$\widehat{L}(x, \widehat{\lambda}) = f(x) + \sum_i \min_{y_i \geqslant g_i(x)} \{\lambda_i y_i + r y_i^2\}.$$

Но $\min\limits_{y_i \geqslant g_i(x)} \{\lambda_i y_i + r y_i^2\}$ достигается:

— либо для $y_i = -\lambda_i/(2r)$, если $g_i(x) \leqslant -\lambda_i/(2r)$ (и тогда этот минимум равен $-\lambda_i^2/(4r)$);

— либо для $y_i = g_i(x)$, если $g_i(x) \geqslant -\lambda_i/(2r)$ (и тогда этот минимум равен $\lambda_i g_i(x) + r[g_i(x)]^2$).

Следовательно, $\widehat{L}$ записывается в виде

$$\widehat{L}(x, \widehat{\lambda}) = \widehat{L}(x, \lambda, r) = f(x) + \sum_{i=1}^{m} G(g_i(x), \lambda_i, r),$$

где

$$G(g_i(x), \lambda_i, r) = \begin{cases} \lambda_i g_i(x) + r[g_i(x)]^2, & \text{если } g_i(x) \geqslant -\lambda_i/(2r) \text{ и } r > 0, \\ -\lambda_i^2/(4r), & \text{если } g_i(x) \leqslant -\lambda_i/(2r) \text{ и } r > 0, \\ \lambda_i g_i(x), & \text{если } r = 0 \text{ и } \lambda_i \geqslant 0, \\ -\infty \text{ в остальных случаях.} \end{cases}$$

Мы снова получили в точности выражение для расширенного лагранжиана, данное в п. 4.1.

4.6. Пример 3: методы множителей.

Расширенный лагранжиан Рокафеллара может рассматриваться как частный случай более общего класса методов, называемых методами множителей, изученных первоначально Хестенесом [32] и Пауэллом [49], а затем в [45, 47, 5].

Эти методы, примененные к задаче минимизации с ограничениями в виде равенств

$$f(x) \to \min,$$
$$g_i(x) = 0, \qquad i = 1, \ldots, m,$$

используют обобщенные лагранжианы в виде

$$\overline{L}(x, \lambda, r) = f(x) + \sum_i \lambda_i g_i(x) + r \sum_i \psi(|g_i(x)|),$$

где функция $\psi(t)$ (функция штрафа) есть такая возрастающая функция t, что $\psi(0) = 0$ и $\psi(t) \geqslant 0$ при $t \geqslant 0$.

Это лагранжево представление соответствует представлению с помощью возмущений:

$$\overline{F}(x, y, s) = \begin{cases} f(x), & \text{если } g_i(x) = y_i \text{ и } \sum\limits_i \psi(|y_i|) \leqslant s, \\ +\infty & \text{в противном случае.} \end{cases}$$

Заметим, что метод множителей Хестенеса и Пауэлла (см. п. 4.1) соответствует частному случаю, в котором функция ψ имеет вид $\psi(t) = t^2$.

4.7. Седловые точки и невыпуклое программирование. Мы сейчас покажем, что теория обобщенных лагранжианов позволяет существенно расширить класс задач, которые допускают седловую точку и, следовательно, могут быть разрешены с помощью алгоритмов, действующих с помощью решения дуальной задачи.

Рассмотрим, например, «расширенную» дуальную функцию, соответствующую «расширенному лагранжиану» Рокафеллара:

$$\widehat{w}(\widehat{\lambda}) = \widehat{w}(\lambda, r) = \min_{x \in \mathbf{R}^n} \widehat{L}(x, \lambda, r),$$

и связанную с ней «расширенную» дуальную задачу

$$\begin{gathered} \widehat{w}(\lambda, r) \to \max, \\ (\lambda, r) \in \mathbf{R}^{m+1}. \end{gathered} \tag{$\widehat{D}$}$$

Интересно интерпретировать эту дуальную функцию $\widehat{w}(\widehat{\lambda})$ в терминах «обычной» функции возмущений:

$$\Phi(y) = \min(P_y) = \min\{f(x) \mid g(x) \leqslant y\},$$

соответствующей «обычному» лагранжиану (см. п. 4.4).

«Расширенная» возмущенная функция

$$\widehat{\Phi}(\widehat{y}) = \min_x \widehat{F}(x, \widehat{y}) = \min_x \widehat{F}(x, y, s)$$

вследствие определения $\widehat{F}$ (п. 4.5) равна

$$\widehat{\Phi}(\widehat{y}) = \widehat{\Phi}(y, s) = \begin{cases} \Phi(y), & \text{если} \quad \|y\|^2 \leqslant s, \\ +\infty & \text{в противном случае.} \end{cases}$$

Тогда, используя (10), получаем

$$\widehat{w}(\widehat{\lambda}) = \widehat{w}(\lambda, r) = \min_{y,s} \{\widehat{\Phi}(y, s) + \lambda y + rs\} = \min_{\substack{y,s \\ \|y\|^2 \leqslant s}} \{\Phi(y) + \lambda y + rs\}.$$

Так как $r > 0$, то s должно быть минимальным и, следовательно, равным $\|y\|^2$. Поэтому имеем

$$\widehat{w}(\widehat{\lambda}) = \widehat{w}(\lambda, r) = \min_y \{\Phi(y) + \lambda y + r\|y\|^2\}. \tag{13}$$

В предположении, что y одномерно, мы на рис. 4 представили «обычную» функцию возмущений $z = \Phi(y)$.

Пусть y^0 — значение y, при котором в (13) достигается минимум. Имеем

$$\widehat{w}(\lambda, r) = \Phi(y^0) + \lambda y^0 + r\|y^0\|^2,$$

$$\widehat{w}(\lambda, r) \leqslant \Phi(y) + \lambda y + r\|y\|^2 \quad \forall y.$$

Следовательно,

$$\Phi(y) \geqslant \widehat{w}(\lambda, r) - \lambda y - r\|y\|^2 \quad \forall y$$

и для $y = y^0$ получаем

$$\Phi(y^0) = \widehat{w}(\lambda, r) - \lambda y^0 - r\|y^0\|^2.$$

Это показывает, что значение дуальной функции $\widehat{w}(\lambda, r)$ определяется пересечением с осью $y = 0$ параболической кривой

$$z = z^0 - \lambda y - r\|y\|^2, \qquad r > 0,$$

всюду расположенной под функцией $\Phi(y)$ и касающейся графика $\Phi(y)$ в точке y^0 (см. рис. 4).

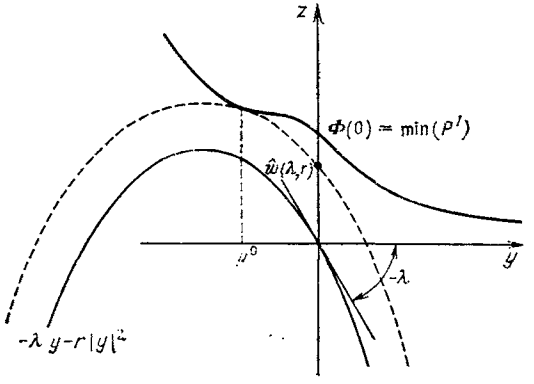


Рис. 4. Геометрическая интерпретация метода расширенного лагранжиана Рокафеллара

Тогда условие существования седловой точки

$$\widehat{w}(\lambda, r) = \Phi(0) = \min(P')$$

эквивалентно следующему условию:

«Обычная» функция возмущений допускает в точке $(y = 0, \Phi(0))$ опорную вогнутую квадратичную функцию.

Мы видим, что это условие может оказаться выполненным для намного более широкого класса функций $\Phi(y)$, чем класс выпуклых функций.

Заметим, что геометрическая интерпретация метода множителей Хестенеса и Пауэлла для задач с ограничениями в виде равенств приводит к совершенно такому же условию.

Расширенный лагранжиан (и некоторый общий вид обобщенных лагранжианов, таких как лагранжианы, описанные в п. 4.6) позволяет также распространить классические необходимые и достаточные условия оптимальности (теорема 6 гл. 5) на некоторый класс невыпуклых задач.

Однако в настоящее время мы не располагаем точной характеризацией типов задач, которые допускают, например, седловую точку расширенного лагранжиана Рокафеллара. С другой стороны, даже если бы мы располагали такой характеризацией, существовали бы определенные трудности другого рода — в частности, в том, что

касается алгоритмизации. Действительно, вычисление дуальной функции $\overline{w}(\lambda)$ влечет необходимость минимизации по переменной x обобщенного лагранжиана $\overline{L}(x, \lambda)$. Однако в случае, если функция f и функции g_i не являются выпуклыми, может оказаться затруднительным найти абсолютный минимум, вообще говоря, не выпуклой по x функции $\overline{L}(x, \lambda)$.

Если вместо абсолютного минимума мы удовлетворимся локальным минимумом, то есть опасность получить «локальную» седловую точку, удовлетворяющую лишь локальному необходимому и достаточному условию оптимальности для исходной задачи (см. п. 3.3).

§ 5. Сравнительное изучение алгоритмов. Сходимость

5.1. Геометрическая интерпретация и сравнение различных алгоритмов. Для их сравнения представляет интерес геометрическая иллюстрация значения основных методов, изученных в этой главе: методов штрафа, а также методов, использующих обычный или расширенный лагранжиан.

Чтобы упростить геометрическое представление, предположим, что задача

$$\begin{aligned} &f(x) \to \min, \\ &g(x) \leqslant 0, \\ &x \in \mathbb{R}^n, \end{aligned} \qquad (P')$$

содержит единственное ограничение ($g(x)$ — скаляр).

Мы будем представлять на рисунках график функции возмущений:

$$\Phi(y) = \min \{f(x) \mid g(x) \leqslant y\}.$$

а) Методы, использующие «обычный» лагранжиан [16].

Если λ — множитель Лагранжа, связанный с ограничением $g(x) \leqslant 0$, то «обычный» лагранжиан имеет вид

$$L(x, \lambda) = f(x) + \lambda g(x)$$

и значение дуальной функции в точке λ равно

$$w(\lambda) = \min_x \{f(x) + \lambda g(x)\} = f(\bar{x}) + \lambda g(\bar{x}).$$

Напомним, что представление с помощью возмущений, соответствующее предыдущему лагранжеву представлению, имеет вид

$$F(x, y) = \begin{cases} f(x), & \text{если } g(x) \leqslant y, \\ +\infty & \text{в противном случае,} \end{cases}$$

и, таким образом, имеем

$$\Phi(y) = \min_x F(x, y).$$

Проверим, что $w(\lambda)$ определяется значением в начале координат прямой с угловым коэффициентом $-\lambda$, касательной к графику $\Phi(y)$

в точке $\bar{y} = g(\bar{x})$, $z = f(\bar{x})$ и расположенной всюду под графиком $\Phi(y)$.

Действительно, имеем

$$w(\lambda) = \min_y \{\Phi(y) + \lambda y\} = \Phi(\bar{y}) + \lambda\bar{y} \leqslant \Phi(y) + \lambda y \quad \forall y.$$

Следовательно, имеем также

$$\Phi(y) \geqslant w(\lambda) - \lambda y \quad \forall y,$$
$$\Phi(\bar{y}) = w(\lambda) - \lambda\bar{y}.$$

Оптимальное значение $w(\lambda^*)$ дуальной задачи

$$\begin{aligned} w(\lambda) &\to \max, \\ \lambda &\in \mathbf{R}^+, \end{aligned} \tag{D}$$

определяется максимальным значением такой прямой в начале координат (рис. 5).

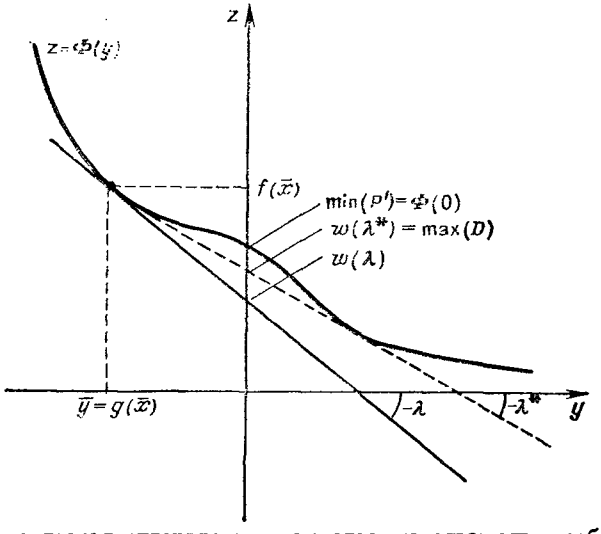


Рис. 5. Геометрическая интерпретация методов, использующих «обычный» лагранжиан

Если функция Φ невыпукла (невыпуклая задача), то, вообще говоря, наличествует скачок двойственности; иначе говоря,

$$\max(D) = w(\lambda^*) < \min(P') = \Phi(0).$$

b) Метод внешних штрафов.

Если r — коэффициент штрафа, то

$$L(x, r) = f(x) + r[\max\{0, g(x)\}]^2, \qquad r > 0,$$
$$L(x, r) = -\infty, \qquad r < 0.$$

Это лагранжево представление соответствует представлению с помощью возмущений:

$$F(x, s) = \begin{cases} f(x), & \text{если } [\max\{0, g(x)\}]^2 \leqslant s, \\ +\infty & \text{в противном случае.} \end{cases}$$

Дуальная функция имеет при $r > 0$ значение

$$w(r) = \min_x \{f(x) + r[\max\{0, g(x)\}]^2\} = f(\bar{x}) + r[\max\{0, g(\bar{x})\}]^2.$$

Полагая $\bar{y} = g(\bar{x})$, замечаем, что $\bar{x}$ необходимо удовлетворяет условию

$$f(\bar{x}) = \Phi(\bar{y}).$$

Действительно, если бы существовал такой x', что $f(x') < f(x)$ и $g(x') \leqslant \bar{y} = g(\bar{x})$, то выполнялось бы соотношение

$$f(x') + r[\max\{0, g(x')\}]^2 < f(\bar{x}) + r[\max\{0, g(\bar{x})\}]^2,$$

откуда следовало бы противоречие. Следовательно, $f(\bar{x})$ минимизирует $f(x)$ на множестве таких x, что $g(x) \leqslant \bar{y}$, и, действительно, получаем $f(\bar{x}) = \Phi(\bar{y})$.

Таким образом, можно записать

$$w(r) = \Phi(\bar{y}) + r[\max\{0, \bar{y}\}]^2 \leqslant f(x) + r[\max\{0, g(x)\}]^2 \quad \forall x. \quad (14)$$

Рассмотрим тогда некоторое y и возьмем в (14) в качестве x вектор, на котором $f(x)$ реализует минимум при ограничении $g(x) \leqslant y$ (следовательно, имеем $f(x) = \Phi(y)$). Получаем

$$w(r) \leqslant \Phi(y) + r[\max\{0, y\}]^2.$$

Поскольку это соотношение выполняется для всех y, то мы получаем:

$$\forall y: \ \Phi(y) \geqslant w(r) - r[\max\{0, y\}]^2,$$
$$y = \bar{y}: \ \Phi(\bar{y}) = w(r) - r[\max\{0, \bar{y}\}]^2.$$

Это показывает, что $w(r)$ определяется пересечением с осью (Oz) кривой, заданной уравнением

$$z = -ry^2 + w(r), \qquad y \geqslant 0,$$
$$z = w(r), \qquad y \leqslant 0,$$

которая вся расположена под графиком $\Phi(y)$ и касается этого графика в точке с координатами $(\bar{y}, \Phi(\bar{y}))$ (рис. 6).

Таким образом, мы получили геометрически тот факт, что $w(r)$ увеличивается с увеличением r и стремится к $\Phi(0)$ при $r \to +\infty$, даже если классический лагранжиан дает скачок двойственности.

С другой стороны, последовательность полученных значений $\bar{y}$ стремится к 0, что означает, что последовательность соответствующих точке $\bar{x}$ стремится к решению задачи.

c) Метод внутренних штрафов.

Если t — коэффициент штрафа ($t > 0$), то

$$L(x, t) = f(x) - \frac{t}{g(x)}$$

для $t > 0$ и $g(x) < 0$.

Для $s > 0$ рассмотрим функцию

$$F(x, s) = \begin{cases} f(x), & \text{если } g(x) \leqslant -1/s, \\ +\infty & \text{в других случаях} \end{cases}$$

(предполагается, что существует по крайней мере одна точка x с $g(x) < 0$).

Тогда получаем, что $L(x, t) = \min\limits_{s} \{F(x, s) + st\}$ для всех таких x, что $g(x) < 0$ и $t > 0$.

Заметим, однако, что $F(x, s)$ не является представлением задачи (P') с помощью возмущений, так как $F(x, 0)$ не определено и

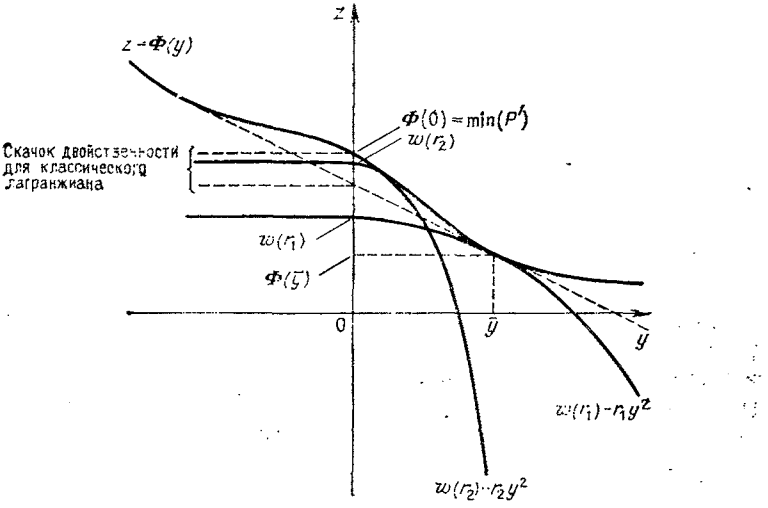


Рис. 6. Геометрическая интерпретация метода внешних штрафов

легко проверить, что $L(x, t)$ не является лагранжевым представлением.

Для данного $t > 0$ рассмотрим функцию (аналогичную дуальной функции)

$$w(t) = \min_{x} \left\{ f(x) - \frac{t}{g(x)} \right\} = f(\bar{x}) - \frac{t}{g(\bar{x})}.$$

То же рассуждение, что и в b), показывает, что необходимо выполняется соотношение

$$f(\bar{x}) = \Phi(\bar{y}) \quad \text{с} \quad \bar{y} = g(\bar{x}) < 0,$$

и можно записать $\forall x$, удовлетворяющего условию $g(x) < 0$:

$$w(t) = \Phi(\bar{y}) - \frac{t}{\bar{y}} \leqslant f(x) - \frac{t}{g(x)}. \tag{15}$$

В частности, выбирая некоторое $y<0$ и выбирая в качестве x в (15) вектор, который реализует минимум $f(x)$ при ограничении $g(x)\leqslant y$, получаем, что

$$w(t)=\Phi(\bar{y})-\frac{t}{\bar{y}}\leqslant\Phi(y)-\frac{t}{y}\qquad \forall y<0.$$

Это показывает, что $w(t)$ определяется асимптотой ветви гиперболы, определяемой уравнением

$$z=w(t)+t/y,\quad y<0,$$

расположенной целиком под графиком $\Phi(y)$ и касающейся этого графика в точке с координатами $(\bar{y}, \Phi(\bar{y}))$ (рис. 7).

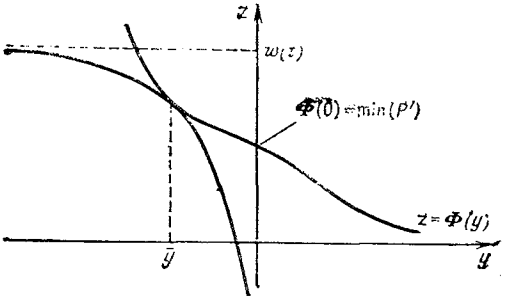


Рис. 7. Геометрическая интерпретация метода внутренних штрафов

Геометрически проверяется, что $w(t)$ убывает, когда t убывает $(t>0)$, и $w(t)$ стремится к $\Phi(0)=\min(P')$ со стороны бо́льших значений, когда t стремится к нулю по положительным значениям (но, вообще говоря, $w(t)>\Phi(0)$, что подтверждает, что $L(x, t)$ не является лагранжевым представлением).

d) Расширенный лагранжиан Рокафеллара.

Мы видели в п. 4.7, что, даже если задача не является выпуклой, может (для достаточно большого $r>0$) существовать параболическая кривая $\Phi(0)-\lambda y-r|y|^2$, касательная в точке $(0, \Phi(0))$ к графику функции $\Phi(y)$ и расположенная целиком под ним (см. рис. 8).

Таким образом, мы получаем седловую точку расширенного лагранжиана, и тогда оказывается возможным решить исходную задачу дуальным методом (отыскивая максимум расширенной дуальной функции).

Предыдущие геометрические построения хорошо иллюстрируют интерес к расширенному лагранжиану (по сравнению с классическим лагранжианом и методами штрафа); этот метод позволяет получить седловую точку и в таких случаях, в которых классический лагранжиан приводит к скачку двойственности, и при этом нет необходимости устремлять коэффициент штрафа к $+\infty$ (а это в свою очередь позволяет избежать некоторых численных трудностей, присущих методам штрафа).

5.2. Анализ сходимости. Как и в п. 3.7 предыдущей главы, мы рассмотрим здесь случай ограничений в форме равенств, что не является ограничением в анализе асимптотического поведения алгоритмов.

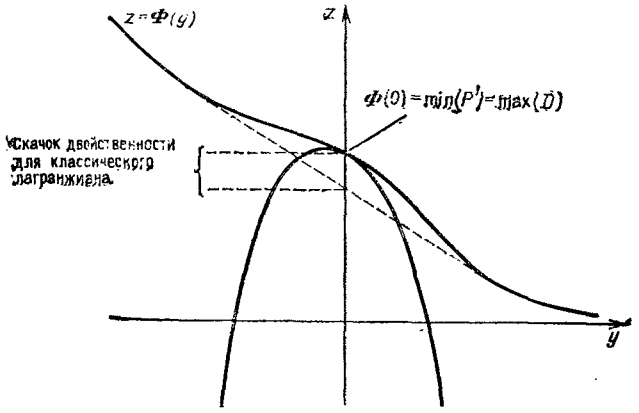


Рис. 8. Геометрическая интерпретация метода расширенного лагранжиана

5.2.1. *Методы штрафов.* Мы изучим в основном методы внешних штрафов, в которых при данном значении $r (r > 0)$ нужно минимизировать функцию

$$\varphi(x, r) = f(x) + r \sum_{i=1}^{m} [g_i(x)]^2.$$

Обозначим полученную точку через $\bar{x}(r)$.

Можно ли предсказать поведение алгоритмов, применяемых для осуществления этой минимизации?

Для метода наискорейшего спуска, например, известно, что скорость сходимости зависит от собственных значений гессиана функции $\varphi(x, r)$ в точке $\bar{x}(r)$.

Вычисление дает:

$$\nabla_x \varphi = \nabla f + 2r \sum_{i=1}^{m} g_i \nabla g_{i_x}$$

$$\nabla_x^2 \varphi = \nabla^2 f + 2r \sum_{i=1}^{m} g_i \nabla^2 g_i + 2r \sum_{i=1}^{m} \nabla g_i \nabla g_i^T$$

(производные вычисляются в точке $\bar{x}(r)$).

В § 1 мы видели, что (при некоторых предположениях) при $r \to \infty$ имеем

$$\bar{x}(r) \to x^* \quad (x^* \text{ — решение исходной задачи}),$$

$$2rg(\bar{x}(r)) \to \lambda^* \quad (\lambda^* \text{ — решение дуальной задачи}).$$

Отсюда выводим, что (при $r \to \infty$) матрица $\nabla_x^2 \varphi$ стремится к матрице, являющейся суммой:

— гессиана по x функции Лагранжа $L(x, \lambda)$ в точке (x^*, λ^*);

— матрицы ранга m вида $2r \sum_{k=1}^{m} \nabla g_k \nabla g_k^T$ с общим членом $b(i, j)$ вида

$$2r \sum_k \frac{\partial g_k}{\partial x_i} \frac{\partial g_k}{\partial x_j}.$$

Тогда можно показать (см., например, [40, 42]), что $\nabla_x^2 \varphi$ имеет m собственных значений, стремящихся к бесконечности при $r \to \infty$ (остальные $n - m$ собственных значений остаются ограниченными).

Поэтому можно предсказать, что методы наискорейшего спуска будут сходиться чрезвычайно медленно вследствие того факта, что при $r \to \infty$ (что необходимо для получения хорошей аппроксимации для x^*) гессиан функции φ становится очень плохо обусловленным.

Следовательно, единственными применимыми методами оказываются методы сопряженного градиента (например, метод Флетчера — Ривза, см. гл. 4, п. 2.7) или квазиньютоновские методы (см. гл. 4, п. 2.10 и 2.11) — вследствие их свойств сходимости (сходимость суперлинейна на n-шагах для первых и суперлинейна для вторых методов).

Относительно сходимости последовательности $\bar{x}(r)$ к x^* доказано, что для достаточно больших значений r

$$f(\bar{x}(r)) - f(x^*) \leqslant \frac{M}{2r},$$

где $M > \|\lambda^*\|^2$.

Аналогичные результаты могут быть получены и для метода внутренних штрафов.

5.2.2. *Методы, использующие «обычный» лагранжиан.* Для анализа сходимости этих методов нужно предположить, что рассматриваемая задача выпукла (по крайней мере локально). Методы типа Удзавы (п. 3.1) состоят в решении дуальной задачи

$$\begin{aligned} &\max w(\lambda), \\ &\lambda \in \mathbf{R}^m, \end{aligned} \tag{D}$$

где $w(\lambda) = \min_x \{f(x) + \lambda g(x)\}$, с помощью метода наискорейшего спуска (так как мы рассматриваем здесь только ограничения в форме равенств, то ограничения на положительность λ нет).

Следовательно, эффективность такого алгоритма зависит от гессиана функции $w(\lambda)$ в точке λ^* — оптимуме дуальной задачи.

Если $w(\lambda)$ непрерывно дифференцируема, то градиент $w(\lambda)$ в точке λ равен

$$\nabla_\lambda w(\lambda) = g(\bar{x}(\lambda)),$$

где $\bar{x}(\lambda)$ — (единственный) минимум по x функции $L(x, \lambda) = f(x) + \lambda g(x)$. Если $w(\lambda)$ дважды непрерывно дифференцируема, то можно показать, что

$$\nabla^2 w(\lambda) = -\left(\frac{\partial g}{\partial x}\right)\left[\nabla_x^2 L\right]^{-1}\left(\frac{\partial g}{\partial x}\right)^T,$$

где

$$\partial g/\partial x = [\nabla g_1, \nabla g_2, \ldots, \nabla g_m]^T \text{ — якобиан } y \text{ в } x$$

и где

$$\nabla_x^2 L = \nabla^2 f + \sum_i \lambda_i \nabla^2 g_i$$

(все производные вычисляются в точке $\bar{x}(\lambda)$).

Тогда, алгоритм типа Удзавы будет сходиться линейно, но с тем более медленной сходимостью, чем больше величина, характеризующая обусловленность матрицы

$$\left(\frac{\partial g}{\partial x}(x^*)\right)\left[\nabla_x^2 L(x^*, \lambda^*)\right]^{-1}\left(\frac{\partial g}{\partial x}(x^*)\right)^T$$

и являющаяся отношением наибольшего собственного значения этой матрицы к наименьшему.

Таким образом, для некоторых типов задач этот метод может оказаться неэффективным.

Алгоритм типа Эрроу — Гурвица (п. 3.1), которые осуществляют этапы наискорейшего спуска поочередно для дуальной функции $w(\lambda)$ и для функции Лагранжа $L(x, \lambda)$, очевидным образом подчиняются тому же типу ограничений.

Чтобы получить более быструю сходимость, нужно, таким образом, привлечь к использованию методы сопряженного градиента или квазиньютоновские методы:

— как для того, чтобы вычислить $\min\limits_x L(x, \lambda)$;

— так и для того, чтобы оптимизировать дуальную функцию $w(\lambda)$.

Предположим, что мы для данного λ ищем минимум по x функции Лагранжа $L(x, \lambda)$ с помощью квазиньютоновского метода (см. гл. 4, п. 2.10 и 2.11).

Тогда мы получаем не только $\bar{x}(\lambda)$, но еще и матрицу H, приближающую матрицу, обратную к гессиану по x функции $L(x, \lambda)$ в точке $\bar{x}(\lambda)$:

$$H \simeq \left[\nabla_x^2 L(x, \lambda)\right]^{-1}.$$

Тогда вычисление гессиана $w(\lambda)$ в точке λ немедленно осуществляется с помощью равенства

$$\nabla_\lambda^2 w(\lambda) = -\left(\frac{\partial g}{\partial x}\right) H \left(\frac{\partial g}{\partial x}\right)^T,$$

где производная $\partial g/\partial x$ вычисляется в точке $\bar{x}(\lambda)$.

Метод Ньютона, *примененный к дуальной задаче*, приводит тогда к итеративной формуле

$$\lambda' = \lambda - \left[\nabla_\lambda^2 w(\lambda)\right]^{-1} \left[\nabla_\lambda w(\lambda)\right] = \lambda + \left[\left(\frac{\partial g}{\partial x}\right) H \left(\frac{\partial g}{\partial x}\right)^T\right]^{-1} g$$

(*g и $\partial g/\partial x$ вычисляются в точке $\bar{x}(\lambda)$*).

Эта формула поправки на практике очень эффективна и приводит к суперлинейной — и даже квадратичной — сходимости в окрестности оптимума (см. [21]).

5.2.3. *Методы, использующие расширенные лагранжианы.* В отличие от предыдущих методов нет необходимости предполагать, что задача выпукла.

И здесь мы, чтобы упростить изложение, ограничимся случаем ограничений в виде равенств.

Расширенная функция Лагранжа имеет в этом случае вид

$$\hat{L}(x, \lambda, r) = f(x) + \lambda g(x) + r\|g(x)\|^2,$$

а расширенная дуальная функция —

$$\hat{w}(\lambda, r) = \min_x \hat{L}(x, \lambda, r).$$

Заметим, что для фиксированного r расширенная дуальная функция есть не что иное, как «обычная» дуальная функция задачи

$$f(x) + r\|g(x)\|^2 \to \min,$$
$$g(x) = 0.$$

Следовательно, все то, что было сказано в п. 5.2.2, приложимо к этой новой задаче — при условии, что функция f заменена функцией

$$f(x) + r\|g(x)\|^2.$$

В частности, гессиан по λ расширенной дуальной функции $\hat{w}(\lambda, r)$ равен

$$\nabla_\lambda^2 \hat{w}(\lambda, r) = -\left(\frac{\partial g}{\partial x}\right)\left[\nabla_x^2 \hat{L}\right]^{-1}\left(\frac{\partial g}{\partial x}\right)^T,$$

где $\nabla_x^2 \hat{L}$ — гессиан расширенной дуальной функции по x — равен

$$\nabla_x^2 \hat{L} = \nabla_x^2 L + 2r\left(\frac{\partial g}{\partial x}\right)^T \left(\frac{\partial g}{\partial x}\right),$$

где и первые, и вторые производные вычисляются в точке $\hat{x}(\lambda, r)$, дающей минимум по x функции $\hat{L}(x, \lambda, r)$.

Можно показать (см. [42]), что при $r \to \infty$ гессиан $\nabla_\lambda^2 \hat{w}(\lambda, r)$ стремится к единичной матрице (в очевидном предположении, что $(\partial g/\partial x)$ имеет полный ранг, равный m).

Таким образом, дуальная функция обусловлена тем лучше, чем больше r, и использование метода наискорейшего спуска для реше-

ния дуальной задачи (алгоритма типа Удзавы) может привести к достаточно быстрой (линейной) сходимости. Напротив, для большого r сама расширенная функция Лагранжа становится плохо обусловленной, и применение алгоритма наискорейшего спуска для получения $\min_x \widehat{L}(x, \lambda, r)$ следует запретить. Следовательно, это исключает и методы типа Эрроу — Гурвица (п. 3.1).

Один из первых методов наискорейшего спуска для оптимизации $\widehat{w}(\lambda, r)$ был предложен в [32]. В этом методе r выбирается достаточно большим и остается постоянным.

Каждая итерация состоит из следующих действий.

а) Для данного λ определить точку $\widehat{x}(\lambda, r)$ минимума функции $\widehat{L}(x, \lambda, r)$ по x.

b) Заменить вектор λ на вектор

$$\lambda' = \lambda + 2rg(\widehat{x}(\lambda, r)).$$

Эта процедура для достаточно больших r сходится к λ^* (оптимуму дуальной задачи) линейно, и мера сходимости тем выше, чем больше r (это сразу следует из поведения гессиана $\widehat{w}$ при $r \to \infty$). С другой стороны, заметим, что в формулу поправки для λ производные не входят.

В связи с приведенными выше соображениями минимизация (без ограничений) функции $\widehat{L}(x, \lambda, r)$ по переменной x должна осуществляться методом сопряженного градиента или квазиньютоновским методом.

Однако если для оптимизации без ограничений привлечен квазиньютоновский метод, то может оказаться выгодным использовать всю полученную информацию и, в частности, матрицу $\widehat{H}$, аппроксимирующую $\left[\nabla_x^2 \widehat{L}\right]^{-1}$, что позволяет пустить в ход метод Ньютона для максимизации $\widehat{w}(\lambda, r)$. Отсюда следует формула для поправки типа формулы

$$\lambda' = \lambda + \left[\left(\frac{\partial g}{\partial x}\right) \widehat{H} \left(\frac{\partial g}{\partial x}\right)^T\right]^{-1} g.$$

При классических предположениях (см. гл. 4, § 2) получаем суперлинейную и даже квадратичную сходимость (см. [21]).

В настоящее время изучены и другие методы, использующие дуальные переменные. В частности, в [25] показано, что может оказаться полезным исправлять каждый раз дуальные переменные — на каждом шаге итерации процесса минимизации расширенного лагранжиана (вместо того, чтобы добиваться достижения точного минимума по x). Таким образом получается метод, близкий методу Эрроу и Гурвица (см. п. 3.1).

Важный с точки зрения практики вопрос касается точности, которой необходимо достигать в минимизации без ограничений. В [4] и [30] можно найти обсуждения, углубляющиеся в этот тип задач.

СПИСОК ЛИТЕРАТУРЫ

1. Ablow C. M., Brigham G. (1955), An Analog Solution of Programming Problems, *Operations Research* 3, 4, p. 388-394.
2*. Arrow K. J., Hurwicz L., Uzawa H. (1958), *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, USA.
3. Bellmore M., Greenberg H. J., Jarvis J. J. (1970), Generalized Penalty Function Concepts in Mathematical Optimization, *Operations Research* 18, p. 229—252.
4. Bertsekas D. P. (1975), Combined primal dual and penalty methods for constrained minimization, *S.I.A.M. Journal on Control* 13, p. 521-544.
5. Bertsekas D. P. (1976), Multiplier methods: a survey, *Automatica* 12, p. 133-145.
6. Best M. J., Braüninger J., Ritter K., Robinson S. M. (1981), A globally and quadratically convergent algorithm for general nonlinear programming problems, *Computing* 26, p. 141-153.
7. Betts J. T. (1977), An Accelerated Multiplier Method for Nonlinear Programming, *Journal Optimization Theory Appl.* 21, 2, p. 137-174.
8. Buys J. D. (1972), Dual Algorithms for Constrained Optimization Problems, *Thèse, Université de Leiden, Pays Bas.*
9. Cohen G., Zhu D. L. (1983), Decomposition-coordination methods in large scale optimization problems. The nondifferentiable case and the use of augmented lagrangians, in: *Advances in large scale systems, theory and applications*, vol. 1 (J. B. Cruz Jr. ed.) JAI Press Inc., Greenwich, Conn. (USA).
10. Conn A. R. (1973), Constrained Optimization Using a Nondifferentiable Penalty Function, *S.I.A.M. J. Numerical Analysis* 10, p. 760-784.
11. Courant R. (1943), Variational methods for the solution of problems of equilibrium and vibrations, *Bull. Amer. Math. Soc.* 49, p. 1-23.
12. Courant R. (1962), *Calculus of Variations and Supplementary Notes and Exercices*, Mimeographed notes, New York University.
13. Dantzig G. B. (1959), General convex objective forms, *The Rand Corporation, Report P.* 1664.
14. Dinkel J. J. (1978), An implementation of surrogate constraint duality, *Operations Research*, vol. 26, n° 2, p. 358-364.
15. Evans J. P., Gould F. J., Tolle J. W. (1973), Exact Penalty Functions in Nonlinear Programming, *Mathematical Programming* 4, p. 72-97.
16. Everett H. (1963), Generalized Lagrange Multiplier Method for solving problems of optimum allocation of resources, *Operations Research* 11, p. 399-417.
17. Falk J. E. (1969), Lagrange Multipliers and Nonconvex Programs, *S.I.A.M. J. Control* 7, p. 534-545.
18. Fiacco A. V. (1976), Sensitivity Analysis for Nonlinear Programming using Penally Methods, *Mathematical Programming* 10, p. 287-311.
19*. Fiacco A. V., McCormick G. P. (1968), *Nonlinear programming: Sequential unconstrained minimization techniques*, John Wiley, New York.
20. Fletcher R. (1973), An Exact Penalty Function for Nonlinear Programming with Inequalities, *Mathematical Programming* 5, p. 129-150.
21. Fletcher R. (1974), Methods related to Lagrangian functions, *in*: *Numerical methods for constrained optimization*, (Gill and Murray eds.), Academic Press, London, p. 219-239.
22. Fletcher R. (1975), An ideal penalty function for constrained optimization, *Journal of the Institute of Mathematics and its Applications* 15, p. 319-342.
23. Geoffrion A. M. (1971), Duality in nonlinear programming: A simplified applications-oriented development, *S.I.A.M. Review*, vol. 13, n° 1, p. 1-37.
24*. Gill P. E., Murray W. (1974), *Numerical methods for constrained optimization*, Academic Press.
25. Glad S. T. (1979), Properties of updating methods for the multipliers in augmented lagrangians, *J. Optimization Theory and Applications*, vol. 28, n° 2, p. 135-156.

26. Glover F. (1975), Surrogate constraint duality in mathematical programming, *Operations Research*, vol. 23, n° 3, p. 434-451.
27. Gould F. J. (1969), Extension of Lagrange Multipliers in Nonlinear Programming, *S.I.A.M. J. Appl. Math.* 17, p. 1280-1297.
28. Greenberg H. J., Pierskalla W. P. (1970), Surrogate Mathematical Programming, *Operations Research* 18, p. 924-939.
29. Grinold R. C. (1970), Lagrangian subgradients, *Management Science* 17, p. 185-188.
30. Han S. P. (1975), Penalty Lagrangian methods in a quasi-Newton approach, *Report TR* 75-252, *Computer Science, Cornell University, Ithaca.*
31. Haarhoff P. C., Buys J. D. (1970), A New Method for the Optimization of a Nonlinear Function subject to Nonlinear Constraints, *Comput. J.* 13, p. 178-184.
32. Hestenes M. R. (1969), Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4, p. 303-320.
33. Huard P. (1963), Dual Programs, *in: Recent Advances in Mathematical Programming*, (R. L. Graves, P. Wolfe eds.), McGraw-Hill, New York, p. 55-62.
34. Huard P. (1971), Tour d'horizon en programmation non linéaire, *Rev. Fr. Informatique Rech. Operat.*, R 1, p. 3-48.
35. Jacques G. (1977), A necessary and sufficient condition to have bounded multipliers in nonconvex programming, *Mathematical Programming* 12, p. 136-138.
36. Kuhn H. W. (1969), Duality in Mathematical Programming, in: *Mathematical Systems Theory and Economics I, Springer-Verlag*, New York, p. 67-91.
37. Lasdon L. S. (1972), An Efficient Algorithm for Minimizing Barrier and Penalty Functions, *Mathematical Programming* 2, p. 65-106.
38. Lootsma F. A. (1972), A survey of methods for solving constrained minimization problems via unconstrained minimization, in: *Numerical Methods for Nonlinear Optimization*, (Lootsma F. A. ed.), Academic Press, New York.
39. Luenberger D. G. (1971), Convergence Rate of a Penalty Function Scheme, *Journal Optimization Theory and Appl.* 7, p. 39-51.
40. Luenberger D. G. (1973), *Introduction to linear and non-linear programming*, Addison Wesley.
41. Luenberger D. G. (1974), A Combined Penalty Function and Gradient Projection Method for Nonlinear Programming, *Journal Optimization Theory and Appl.* 14, p. 477-495.
42. Luenberger D. G. (1976), Algorithmic analysis in constrained optimization, in: *Nonlinear Programming*, (R. W. Cottle, C. E. Lemke eds.), *S.I.A.M., AMS Proceeding*, vol. IX, p. 39-52.
43. Magnanti T. L. (1974), Fenchel and Lagrange Duality are Equivalent, *Mathematical Programming* 7, p. 253-258.
44. Mangasarian O. L. (1962), Duality in Nonlinear Programming, *Quart. Appl. Math.* 20, p. 300-302.
45. Miele A., Gragg E. E., Iver R. R., Levy A. V. (1971), Use of the Augmented Penalty Function in Mathematical Programming Problems, Parts I and II, *Journal Optimization Theory and Appl.*, p. 115-130 et p. 131-153.
46. Miele A., Moseley P., Levy A. V., Coggins G. H. (1972), On the Method of Multipliers for Mathematical Programming Problems, *Journal Optimization Theory and Appl.* 10, p. 1-33.
47. Nakayama H., Sayama H., Sawaragi Y. (1975), A generalized Lagrangian function and multiplier method, *Journal Optimization Theory and Appl.*, vol. 17, n° 3/4, p. 211-227.
48. Pierre D. A., Lowe M. J. (1975), *Mathematical programming via augmented Lagrangians Addison Wesley*, Reading Mass.
49. Powell M. J. D. (1969), A Method for nonlinear constraints in minimization problems, *in: Optimization*, (R. Fletcher ed.), Academic Press, New York, p. 283-298.
50. Powell M. J. D. (1978), Algorithms for nonlinear constraints that use Lagrangian functions, *Mathematical Programming* 14, p. 224-248.

51. Rockafellar R. T. (1973a), A dual approach to solving nonlinear programming problems by unconstrained optimization, *Mathematical Programming* 5, p. 354-373.
52. Rockafellar R. T. (1973b), The Multiplier method of Hestenes and Powell applied to convex programming, *Journal of Optimization Theory and Applications* 12, 6, p. 555-562.
53. Rockafellar R. T. (1974), Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming, *S.I.A.M. Journal Control.* 12, p. 268-285.
54. Rockaffelar R. T. (1976), Lagrange multipliers in optimization, *in: Nonlinear Programming*, S.I.H.M., AMS Proceedings, vol. IX, (R. W. Cottle, C. E. Lemke eds.), New York, March 23-24, 1975.
55. Roode J. D. (1968), Generalized Lagrangian functions in Mathematical Programming, *Thèse Univ. Leiden. Pays Bas.*
56. Rupp R. D. (1975), On the combination of the multiplier method of Hestenes and Powell with Newton's method, *J. Optimization Theory and Appl.* 15, p. 167-187.
57. Ryan D. M. (1974), Penalty and Barrier functions, in: *Numerical methods for constrained optimization*, (P. E. Gill, W. Murray eds.), Academic Press, p. 174-190.
58*. Uzawa H. (1958), Iterative methods for concave programming, in: *Studies in linear and nonlinear programming*, Chap. 10. (Arrow. Hurwicz, Uzawa eds.), Stanford University Press.
59. Vajda S. (1967), Nonlinear Programming and Duality, in: *Nonlinear Programming*, (Abadie J. ed.), North Holland, Amsterdam.
60. Watanabe N., Nishimura Y., Matsubara M. (1978), Decomposition in Large System Optimization using the Method of Multipliers, *Journal Optimization Theory and Appl.* 25, n° 2, p. 181-193.

ГЛАВА 7
ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

Эта глава посвящена решению целочисленных экстремальных задач, т. е. задач, в которых переменные могут принимать лишь целые значения (или некоторые целые значения). Речь идет об одной из наиболее богатых и активно развивающихся областей математического программирования. Обилие исследований и публикаций в этой сфере после появления работ Гомори (к 1958 г.) свидетельствует о трудности предмета и важности его приложений.

Цель настоящей главы — представить синтез основных методов решения, используемых в целочисленном программировании; при этом делается попытка выделить основные идеи, поддерживающие большую часть алгоритмов, известных к настоящему времени. Тем самым будут указаны основные ключи, позволяющие в дальнейшем приступить к изучению более специальных источников.

§ 1. Введение

Рассмотрим линейную задачу

$$\begin{aligned} z = cx &\to \min, \\ Ax &= b, \\ x &\geqslant 0. \end{aligned} \tag{PL}$$

Все коэффициенты c_j $(j = 1, \ldots, n)$, a_{ij} $(i = 1, \ldots, m,\ j = 1, \ldots, n)$, b_i $(i = 1, \ldots, m)$ предполагаются целыми. С другой стороны, чтобы упростить изложение, введем условие, что политоп

$$\mathscr{P} = \{x \mid x \in \mathbf{R}^n,\ Ax = b,\ x \geqslant 0\}$$

ограничен и непуст (эти условия, как мы увидим в § 4, не являются слишком ограничительными).

Исключая весьма частные случаи (например, случай, когда полная матрица A ограничений унимодулярна), оптимальное решение задачи (PL) будет, вообще говоря, содержать дробные компоненты (поскольку общий знаменатель представляет собой абсолютное значение определителя оптимального базиса).

Однако мы будем предполагать, что переменные x_j представляют собой количество неделимых объектов (корабли, самолеты и т. д.). Например, авиакомпания будет пытаться реализовать некоторую годовую программу полетов, минимизируя общую стоимость своих самолетов. Тогда переменные будут представлять собой число аппа-

ратов каждого типа, которые требуется купить или нанять. При этом дробное решение недопустимо.

В этом случае мы должны наложить на переменные дополнительные ограничения (называемые ограничениями целости) типа

$$x_j \text{ целое} \quad \forall j = 1, \ldots, n.$$

Стало быть, задача примет вид:

$$z = cx \to \min,$$

$$Ax = b,$$

$$x \geqslant 0,\ x_j \text{ целое} \quad \forall j = 1, \ldots, n;$$

такая задача называется *целочисленной линейной задачей.*

Заметим, что (PNE) — задача того же типа, что и задача (P) из гл. 1, § 1, и что в этом случае $S = \mathbf{Z}^n$.

В противовес этому задача (PL), полученная из (PNE) освобождением ограничений от условия целости, будет называться *непрерывной линейной задачей* (подразумевается, что она соответствует целочисленной задаче (PNE)).

В некоторых случаях может оказаться, что лишь некоторые переменные принимают целочисленные значения. Тогда говорят о *смешанном линейном программировании.*

В этой главе мы не будем изучать методы решения, специально подобранные к смешанным линейным задачам: метод разбиения Бендерса [15], описываемый в гл. 8, § 4, позволяет сводить такие задачи к решению последовательности чисто целочисленных линейных задач (т. е. задач, в которых все переменные принимают целые значения).

Заметим, что целочисленные линейные задачи не покрывают всего класса целочисленных задач. В самом деле, во многих ситуациях оптимизируемая функция или ограничения задачи могут быть нелинейными. Парадоксально, что нелинейные целочисленные задачи не всегда более сложны, чем линейные. В обоих случаях непреходящая трудность одна и та же, и мы ограничимся изучением линейных задач.

Первая идея, которая возникает, когда мы сталкиваемся с целочисленной задачей, состоит в том, чтобы обратиться к методу округления, заменив, например, в непрерывном оптимальном решении каждую дробную компоненту ближайшей целочисленной.

Приводимый ниже пример ясно показывает недостаточность таких методов и позволяет лучше понять неуловимую трудность задач целочисленного программирования.

Пример. Рассмотрим задачу (типа задачи о рюкзаке) с двумя переменными и с единственным ограничением:

$$\min z = -10x_1 - 11x_2,$$

$$10x_1 + 12x_2 \leqslant 59,$$

$$x_1,\ x_2 \geqslant 0, \quad \text{целые.}$$

Легко представить себе множество решений в плоскости, непрерывных с одной стороны и целочисленных с другой (см. рис. 1). Непрерывный оптимум есть точка с координатами $x_1 = 5{,}9$, $x_2 = 0$, для которой $z = -59$.

Простой метод округления привел бы к решению $x_1 = 6$, $x_2 = 0$, которое не удовлетворяет ограничениям.

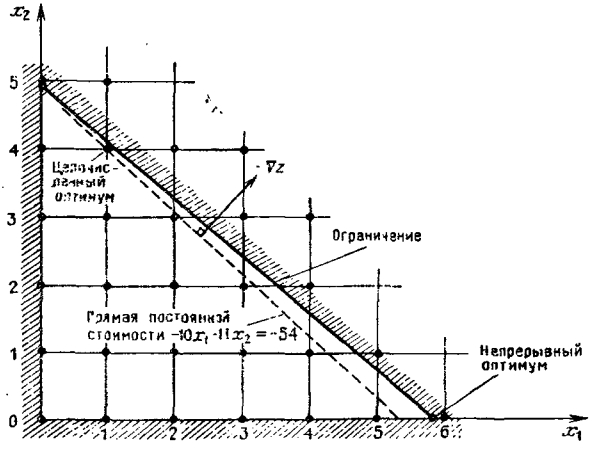


Рис. 1. Пример, иллюстрирующий различие между «целочисленным» и «непрерывным» решениями задачи линейного программирования

Изучая теперь точки с целыми координатами внутри полиэдра непрерывных решений, мы констатируем, что целочисленный оптимум есть точка с координатами $x_1 = 1$, $x_2 = 4$ и $z = -54$.

Ясно что эта точка не имеет ничего общего с непрерывным оптимумом: она отстоит от него слишком далеко.

На самом деле легко построить примеры того же типа, для которых целочисленный оптимум будет удален от непрерывного сколь угодно далеко.

Это объясняет причину того, что методы округления, вообще говоря, неэффективны (однако они могут в некоторых случаях приводить к хорошим приближенным решениям; в предыдущем примере, округляя непрерывный оптимум до ближайшего снизу целого числа, мы получили бы решение $x_1 = 5$, $x_2 = 0$ со стоимостью -50, удаленное от оптимума не более чем на 10 %).

Эту главу мы начнем с изучения двух основных семейств известных в настоящее время методов решения линейных задач программирования в целых числах: методов разветвленного поиска (или частичного перечисления) в § 2 и методов отсечения (или сечения) в § 3.

За исключением динамического программирования (детально изложенного в гл. 9), которое в некоторых задачах дает интересные алгоритмы, большая часть известных алгоритмов происходит из одного из этих двух семейств.

Наиболее широко используются методы разветвления, и они позволяют успешно разрешать некоторые трудные комбинаторные задачи, такие как задача о коммивояжере ([47, 43, 18]), задачи упорядочения ([23]), задачи локализации и классификации ([66, 3]), обобщенная задача о назначениях ([72, 60]) и т. д.

Методы сечения позволяют, со своей стороны, решать частные задачи типа «разбиения» или «покрытия» (см., например, [21]), где коэффициенты матрицы ограничений малы (0 или 1) и вследствие этого базисные миноры растут не слишком быстро.

Наконец, в § 4 мы изучим класс методов, которые используют представление целочисленных задач с помощью кратчайших путей в графе и теории конечных групп. Полученные к настоящему времени результаты представляют большой интерес в теоретической физике, поскольку скачки двойственности, вообще говоря, могут быть сведены к нулю построением подходящего графа (или конечной группы).

§ 2. Методы разветвленного поиска посредством разделения и оценки

Принцип этих методов — методов «ветвей и границ» (Branch and Bound) — в основном принадлежит Лэнду и Дойгу [56], Бертье и Рою [16], Рою, Бертье и Нгьему [73], Дакину [19], Эрве [48]. Затем они были использованы и улучшены многочисленными авторами. Можно также найти хорошие изложения одновременно теории и приложений в [59, 64, 29, 45].

Рассмотрим целочисленную задачу линейного программирования вида

$$\begin{gathered} z = cx \to \min, \\ Ax = b, \qquad (PNE) \\ x \geqslant 0,\ x_j \text{ целые}, \qquad j = 1, \ldots, n, \end{gathered}$$

где A — матрица $m \times n$, $b \in \mathbf{Z}^m$, $x \in \mathbf{Z}^n$.

Если политоп

$$\mathscr{P} = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geqslant 0\}$$

предполагается ограниченным (следовательно, это — выпуклый многогранник), то всегда возможно связать с каждой из переменных x_j границы ее изменения:

$$\alpha_j \leqslant x_j \leqslant \beta_j.$$

Чтобы определить, например, β_j, можно решить (непрерывную) задачу линейного программирования

$$\begin{gathered} x_j \to \max, \\ Ax = b, \\ x \geqslant 0. \end{gathered}$$

Итак, всегда можно ограничиться случаем, в котором каждая переменная x_j может принимать лишь конечное число значений.

2.1. Редукция к задаче с переменными, принимающими два значения (0 или 1). Мы можем продвинуться еще дальше, замечая, что каждую переменную x, могущую принимать лишь $k+1$ целых значений 0, 1, 2, ..., k, можно представить в виде линейной комбинации

$$x = y_0 + 2y_1 + 4y_2 + \ldots + 2^p y_p$$

$p+1$ переменной y_0, y_1, ..., y_p, каждая из которых связана ограничением, что она принимает не более двух значений 0 и 1 (двузначные переменные); число p — наименьшее из таких целых чисел, что $k \leqslant 2^{p+1} - 1$.

Таким образом, всегда можно ограничиться случаем, когда (*PNE*) — задача линейного программирования с двузначными переменными.

(Важное замечание: предыдущее преобразование всегда возможно, но оно не всегда дает преимущество на практике.)

Пример. Рассмотрим задачу целочисленного программирования с ограниченными переменными:

$$\begin{gathered} \min z = 3x_1 - 4x_2, \\ x_1 + 2x_2 \leqslant 6, \\ -2 \leqslant x_1 \leqslant +4, \\ 1 \leqslant x_2 \leqslant 3, \\ x_1,\ x_2 \text{ целые.} \end{gathered}$$

Используя пять двузначных переменных $y_1, \ldots, y_5$, осуществим замену переменных:

$$\begin{aligned} x_1 &= -2 + y_1 + 2y_2 + 4y_3, \\ x_2 &= 1 + y_4 + 2y_5. \end{aligned}$$

Подстановка дает

$$\begin{gathered} \min z = -10 + 3y_1 + 6y_2 + 12y_3 - 4y_4 - 8y_5, \\ y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 \leqslant 6, \\ y_1 + 2y_2 + 4y_3 \leqslant 6, \\ y_4 + 2y_5 \leqslant 2, \\ y_i = 0,\ 1, \qquad i = 1, \ldots, 5. \end{gathered}$$

На этом основании мы в настоящем параграфе ограничимся изучением задач линейного программирования с двузначными переменными в виде

$$\begin{gathered} z = cx \to \min, \\ Ax = b, \qquad (PB) \\ x_j = 0,\ 1 \qquad \forall j = 1, \ldots, n. \end{gathered}$$

Или, в другой форме, если через S обозначить множество n-векторов с компонентами 0 или 1:

$$z = cx \to \min,$$
$$Ax = b, \qquad (PB)$$
$$x \in S.$$

Так как $|S| = 2^n$, то множество решений задачи (PB) состоит из конечного числа элементов.

2.2. Перечисление? Самая простая идея, какую только можно себе представить для решения задачи (PB), состоит, следовательно, в перечислении 2^n элементов множества S и в выборе среди всех векторов $x \in S$, удовлетворяющих ограничениям $Ax = b$, вектора $\bar{x}$, приводящего к наименьшему значению $z = cx$.

Ясно, что такой алгоритм был бы конечным. Но, поскольку число переменных n может оказаться большим (больше 50, чтобы было ясно, о чем идет речь), то может оказаться, что должны пройти долгие века работы самого мощного компьютера для перечисления лишь малой доли из 2^n элементов множества S.

Следовательно, для эффективного решения на вычислительных машинах задач большого размера (с 100 или более двузначных переменных) нужно искать алгоритмические принципы, позволяющие определять оптимальное решение без необходимости эффективно (явно) перечислять элементы множества S.

Именно эта идея неявного (в противоположность явному) перечисления решений и лежит в основе методов разветвленного поиска с помощью разделения и оценки — эти методы называются так потому, что используют представление множества S в форме некоторого разветвления.

2.3. Определение разветвления: понятие разделения. Вершины разветвления соответствуют подмножествам S (иначе говоря, подмножествам множества 0—1-векторов) и следующим образом разбиваются на уровни.

Существует единственная вершина уровня 0 (называемая корнем разветвления), которой множество соответствует все целиком.

Для построения уровня 1 разветвления нужно сначала выбрать (и этот выбор произволен) некоторое переменное, например x_1. Тогда уровень 1 содержит две вершины, обозначаемые $S_{\bar{1}}$ и S_1, которые соответствуют следующим подмножествам: первое — подмножеству векторов с элементами 0—1, для которых переменное x_1 имеет значение 0, и второе — подмножеству векторов с элементами 0—1, для которых переменное x_1 имеет значение 1. Очевидно, что $S_{\bar{1}} \subset S$ и $S_1 \subset S$. Кроме того, $S_{\bar{1}}$ и S_1 образуют разбиение S.

Вершины S и $S_{\bar{1}}$ (соответственно S и S_1) связываются стрелкой (дугой), представляющей отношение включения. Тогда будем говорить, что мы «разделили» множество S относительно переменной x_1. Аналогичным образом для построения уровня 2 нужно

выбрать (произвольно) второе переменное, например x_2. Тогда мы получаем четыре (2^2) вершины уровня 2:

$S_{\bar{1}\bar{2}}$ — множество 0—1-векторов, для которых $x_1 = 0$ и $x_2 = 0$,

$S_{\bar{1}2}$ — множество 0—1-векторов, для которых $x_1 = 0$ и $x_2 = 1$,

$S_{1\bar{2}}$ — множество 0—1-векторов, для которых $x_1 = 1$ и $x_2 = 0$,

S_{12} — множество 0—1-векторов, для которых $x_1 = 1$ и $x_2 = 1$.

И снова отношения включения представляются стрелками (дугами), соединяющими вершину $S_{\bar{1}}$ с вершинами $S_{\bar{1}\bar{2}}$ и $S_{\bar{1}2}$ с одной стороны; вершину S_1 с вершинами $S_{1\bar{2}}$ и S_{12} — с другой стороны. Говорят, что мы разделили $S_{\bar{1}}$ (соответственно S_1) относительно переменной x_2.

На рис. 2 изображено разветвление, связанное с задачей типа (PB), связанное с тремя двузначными переменными x_1, x_2, x_3.

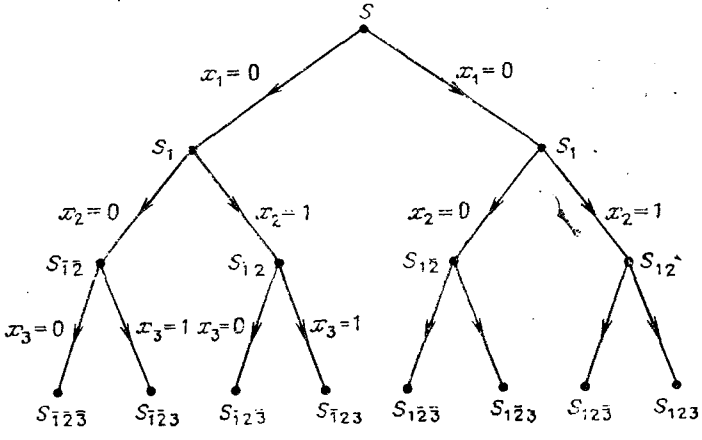


Рис. 2. Разветвление, соответствующее целочисленной задаче с тремя двузначными переменными x_1, x_2, x_3

Заметим, что восемь элементов S соответствуют вершинам, не имеющим следующих за ними вершин разветвления (такие вершины называются «висячими» или, еще, конечными).

2.4. Оценка. После того как мы ввели понятие разделения, перейдем к понятию оценки.

Предположим, что для каждой вершины S_i предыдущего разветвления можно с помощью вычисления определить оценку снизу — иначе говоря, миноранту $f(S_i)$ функции стоимости cx лучшего решения $x \in S_i$, так что

$$f(S_i) \leqslant \min_{x \in S_i} \{cx\}.$$

Тогда функция f будет называться *функцией оценки.*

Построим шаг за шагом, начиная с уровня 0 (вершина S), разветвление подмножеств S, причем разветвление осуществляется исследованием переменных в некотором (априори произвольном) порядке.

Пусть $\bar{x}$ — некоторое решение задачи (PB) со стоимостью

$$\bar{z} = c\bar{x},$$

определенное либо в ходе предыдущих этапов построения разветвления, *либо с помощью некоторого приближенного (эвристического)* метода (если ничего найти не удалось, то можно по соглашению принять $\bar{z} = +\infty$).

Предположим тогда, что для некоторой вершины S_i построенного разветвления имеем

$$f(S_i) > \bar{z}.$$

По определению функции f никакое содержащееся в S_i решение не может быть лучше, чем $\bar{z}$, и поэтому можно быть уверенным, что S_i не содержит оптимального решения.

Это признание позволяет избежать исследования (перечисления) всех прямых или косвенных вершин разветвления, следующих за S_i.

Ограничиваясь на каждом шаге перечислением вершин, следующих только за такими вершинами S_i, что $f(S_i) \leqslant \bar{z}$, можно достичь подходящей редукции числа эффективно исследуемых вершин, достаточной для исследования задач практически важных размеров. Это — общий принцип метода.

2.5. Практическая работа. Во всем изложенном остается много степеней свободы, и можно сказать, что существует столько же алгоритмов, сколько есть способов выбрать:

a) природу функции оценки;

b) вершину, подлежащую разделению, на каждом данном этапе;

c) следующую переменную, по которой будет осуществляться разделение выбранной вершины.

Исследуем последовательно эти три вопроса.

Выбор в a) не всегда очевиден. Всегда можно использовать в качестве оценки снизу «непрерывное» оптимальное решение задачи линейного программирования, ограниченной оставшимися свободными переменными. Однако такая оценка, вообще говоря, дорогостоящая по времени вычислений. Можно также предпочесть не такую хорошую функцию оценки (иначе говоря, дальше отстоящую от значения целочисленного оптимума), но допускающую гораздо более быстрое получение.

Вообще говоря, очень эффективная техника для получения хороших функций оценки связана с лагранжевым ослаблением ограничений $Ax = b$ (см. гл. 4, § 3, гл. 6, § 2 и п. 4.6 этой главы). Тогда дуальная задача разрешается с помощью алгоритма субградиента (в связи с минорантным свойством дуальной функции —

см. гл. 6, п. 2.1 — можно удовлетвориться приближенным способом решения дуальной задачи).

Целочисленные задачи в больших размерностях могут быть решены именно этим способом; таковы: задача о коммивояжере ([47, 43, 18]), задачи упорядочения ([23]), задачи локализации и классификации ([66, 3]), обобщенная задача о назначениях ([72, 60]). За обзором множества методов и их приложений можно отослать также к [28, 25, 80, 21].

Относительно выбора в b) можно предложить много стратегий, из которых, насколько нам известно, ни одна не может систематически проявлять себя как наилучшая. В методе, называемом «поиск в глубину», договариваются выбирать вершину наиболее высокого уровня среди вершин, еще не подвергнутых разделению. Если таких вершин много, то можно выбрать вершину, соответствующую наиболее низкой оценке. Этот метод имеет целью наибыстрейшим возможным образом предъявить решение задачи. В методе, называемом «поиск в ширину» («breadth first search») или методом SEP (прогрессирующее разделение и оценка) систематически выбирается вершина, имеющая наиболее низкую оценку, учитывая, что — интуитивно это ясно — именно она имеет более всего шансов содержать — среди следующих за ней вершин — оптимальное решение. С этим методом связан риск оказаться вынужденным исследовать существенную долю разветвления перед тем, как обнаружится решение. Но — в виде реванша,— вообще говоря, полученное на этом пути решение оказывается лучшего качества (иначе говоря, не с такой высокой ценой), чем решение, определенное с помощью процедуры «поиск в глубину».

В свою очередь и выбор в c) также имеет много вариантов. Очень часто порядок следования переменных фиксирован раз навсегда — либо совершенно произвольным образом, либо вследствие интуитивно проверяемых эвристических критериев (см. пример в п. 2.6). Но нужно знать, что этот выбор может быть полностью определенным (в зависимости от принятого порядка время вычисления может меняться от 1 до 10 или от 1 до 100!). Это заставляет принимать во внимание гораздо более гибкие (динамические) стратегии, в которых на каждом этапе выбирается переменная, которая оказывается наилучшей по некоторому критерию. Например, в методе, называемом *штрафным* (см., например, [44]), с каждым переменным x_i, которое может служить для разделения, связывается число p_i (штраф) — точное или приближенное значение разности между новыми оценками, получаемыми при полагании $x_i = 0$, с одной стороны, и $x_i = 1$ — с другой. Тогда если осуществить разделение, соответствующее переменной x_i, с которой связан максимальный штраф p_i, то мы получим два новых подмножества, одно из которых имеет гораздо больше шансов содержать оптимальное решение, чем другое. Речь идет в некотором роде о «наиболее информирующем» выборе. Это приводит к минимизации опасности напрасно исследовать одну ветвь разветвления, тогда как оптимальное решение содержится в другой ветви.

2.6. **Пример.** Для иллюстрации функционирования методов разветвления рассмотрим следующую задачу:

$$\min z = -20x_1 - 16x_2 - 11x_3 - 9x_4 - 7x_5 - x_6,$$
$$9x_1 + 8x_2 + 6x_3 + 5x_4 + 4x_5 + x_6 \leqslant 12,$$
$$x_j \geqslant 0,\ x_j = 0,\ 1,\ j = 1, \ldots, 6,$$

которая имеет вид

$$\min \left(\sum c_j x_j\right),$$
$$\sum a_j x_j \leqslant b,$$
$$x_j = 0,\ 1$$

(*так называемая задача о рюкзаке*).

Заметим, что переменные упорядочиваются по возрастанию отношений c_j/a_j и что непрерывное решение (получаемое заменой целочисленного ограничения $x_j = 0$ или 1 условием $0 \leqslant x_j \leqslant 1$) имеет вид

$$x_1 = 1,\ x_2 = 3/8 \text{ со значением } z = -20 - 6 = -26.$$

Это значение составляет, таким образом, оценку снизу на множестве всех возможных целочисленных решений (действительно, *добавляя ограничения*, в частности ограничения целочисленности, можно лишь увеличить значение оптимума).

Заметим теперь, что можно легко получить *приближенное* целочисленное решение этой задачи, используя (эвристическое) правило следующего типа:

a) сначала все переменные равны 0;

b) с учетом всех предыдущих выборов найти переменное с *наименьшим* отношением c_j/a_j, которому можно приписать значение 1 так, чтобы *ограничение* (неравенство) соблюдалось, и присвоить этому переменной значение 1.

Продолжать b), пока не выяснится, что больше никакой переменной, удовлетворяющей этим условиям, нет.

Это правило дает

$$x_1 = 1,$$

затем, в соответствии с этим выбором,

$$x_6 = 1.$$

Полученное таким образом целочисленное решение имеет значение -21.

В соответствии со всем предыдущим мы сейчас изучим последовательные этапы разветвленного поиска.

На первом уровне разделим S на два подмножества: множество S_1 из 0—1-векторов, для которых $x_1 = 1$, и множество $S_{\bar{1}}$ таких 0—1-векторов, что $x_1 = 0$.

(Важное замечание: выбор переменного x_1 для осуществления разделения произволен, но следует отметить, что этот выбор не

является совершенно безобидным и от него часто может зависеть эффективность метода. В дальнейшем изложении примера последовательные разделения осуществляются в соответствии с возрастанием отношений c_j/a_j, что и оказывается естественным порядком переменных. Этот порядок может быть оправдан интуитивно ясными рассуждениями; с другой стороны, можно проверить, что он приводит к намного более быстрому решению, чем принятие, например, противоположного порядка переменных.)

На этом уровне можно легко получить оценку снизу наилучшего целого решения, содержащегося в каждом из подмножеств S_1 и $S_{\bar{1}}$ соответственно.

Рассмотрим, например, подмножество S_1.

Поскольку известно, что $x_1 = 1$, то задача касается только переменных $x_2, \ldots, x_6$ (которые будут называться «свободными» переменными) и имеет вид

$$\begin{gathered} \min(-20 - 16x_2 - 11x_3 - 9x_4 - 7x_5 - x_6), \\ 8x_2 + 6x_3 + 5x_4 + 4x_5 + x_6 \leqslant 3, \qquad P_1 \\ x_i = 0,\ 1; \end{gathered}$$

она имеет непрерывное решение

$$x_2 = 3/8 \text{ со значением } z = -20 - 6 = -26.$$

Таким образом, оценка снизу наилучшего целого решения, содержащегося в S_1, есть — 26. Обозначим это решение $\hat{z}_1$.

Возьмем теперь подмножество $S_{\bar{1}}$. Поскольку известно, что $x_1 = 0$, то проблема, относящаяся к свободным переменным, имеет вид

$$\begin{gathered} \min(-16x_2 - 11x_3 - 9x_4 - 7x_5 - x_6), \\ 8x_2 + 6x_3 + 5x_4 + 4x_5 + x_6 \leqslant 12, \qquad P_{\bar{1}} \\ x_i = 0,\ 1, \end{gathered}$$

и имеет непрерывное решение

$$x_2 = 1,\ x_3 = \frac{2}{3} \text{ со значением } \hat{z}_{\bar{1}} = -16 - \frac{22}{3} = -\frac{70}{3} (\approx -23{,}3).$$

Так как наилучшее найденное к настоящему времени целое решение ($x_1 = 1$; $x_6 = 1$) имеет значение -21, то ясно, что каждое из подмножеств S_1 и $S_{\bar{1}}$ может содержать лучшее решение, в частности оптимальное решение. Следовательно, нужно разделить каждое из этих подмножеств на два (или более) подмножества, которые, в свою очередь, приведут к некоторым оценкам, и так далее вплоть до обнаружения оптимального решения.

Однако в том месте процесса разветвленного поиска, на котором мы остановились, есть важная задача: какое из множеств S_1 или $S_{\bar{1}}$ нужно разделять первым?

Как мы уже говорили, здесь можно применить много правил. Мы выберем метод исследования «поиск в ширину» («breadth first search», или метод *SEP*), основанный на следующем замечании:

если оценка снизу некоторого подмножества является хорошей аппроксимацией (иначе говоря, соответствующая разность мала) наилучшего целочисленного решения, содержащегося в этом подмножестве, то разумно искать сначала оптимальное решение в подмножестве, связанном с наименьшей из оценок. В интересующем нас примере наименьшую оценку (−26) имеет именно подмножество S_1.

Следовательно, разделим S_1 на два подмножества:

— подмножество $S_{1\bar{2}}$ — таких 0—1-векторов, что $x_1 = 1$ и $x_2 = 0$;

— подмножество S_{12} таких 0—1-векторов, что $x_1 = 1$ и $x_2 = 1$.

(Важное замечание: выбор переменного x_2 для осуществления разделения произволен, но еще раз напомним, что этот выбор не вполне безразличен и от него часто зависит эффективность метода. Однако при дальнейшем разборе примера последовательные разделения будут осуществляться в соответствии с порядком возрастания отношений c_j/a_j, иначе говоря — в естественном порядке следования переменных, и больше мы к этому возвращаться не будем.)

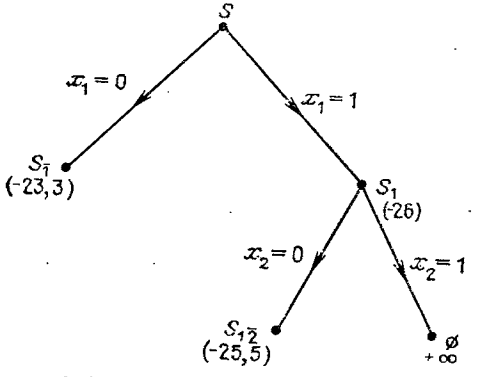


Рис. 3. Иллюстрация действия метода *SEP* на примере: начало построения разветвления

Немедленно замечаем, что множество S_{12} есть пустое множество $\varnothing$, так как $x_1 = 1$ и $x_2 = 1$ не допускают ни одного случая, когда бы ограничение выполнялось. Следовательно, мы больше никогда не будем в дальнейшем принимать это подмножество во внимание, и хорошее средство удостоверить этот факт состоит в приписывании этому подмножеству оценки $+\infty$.

Для множества $S_{1\bar{2}}$ получаем оценку снизу:

$$-20 - 11/2 = -25{,}5.$$

Множество операций, осуществленных к настоящему моменту, объединено в разветвление на рис. 3.

Точки (вершины) соответствуют различным последовательно рассматриваемым подмножествам. Полученные оценки указаны в скобках у вершины, представляющей соответствующее подмножество. Стрелки (дуги) соответствуют отклонениям включения (или родства) между подмножествами.

К настоящему моменту именно подмножество $S_{1\bar{2}}$ имеет наименьшую оценку −25,5. Если мы разделим это подмножество

(относительно переменной x_3), то мы получим два подмножества:

$S_{1\bar{2}\bar{3}}$ с оценкой $-20-27/5=-25{,}4$,

$S_{1\bar{2}3}$ с оценкой $+\infty$ $(S_{1\bar{2}3}=\varnothing)$.

Продолжая таким образом построение этой ветви разветвления, получим последовательно $S_{1\bar{2}\bar{3}\bar{4}}$ (с оценкой $-25{,}2$) и $S_{1\bar{2}\bar{3}\bar{4}\bar{5}}$ (с оценкой -23), см. рис. 4.

Так как оценка $S_{\bar{1}}$ меньше последней (она равна $-23{,}3$), то мы возвращаемся к $S_{\bar{1}}$, который разделяем на $S_{\bar{1}\bar{2}}(-21{,}7)$ и $S_{\bar{1}2}$

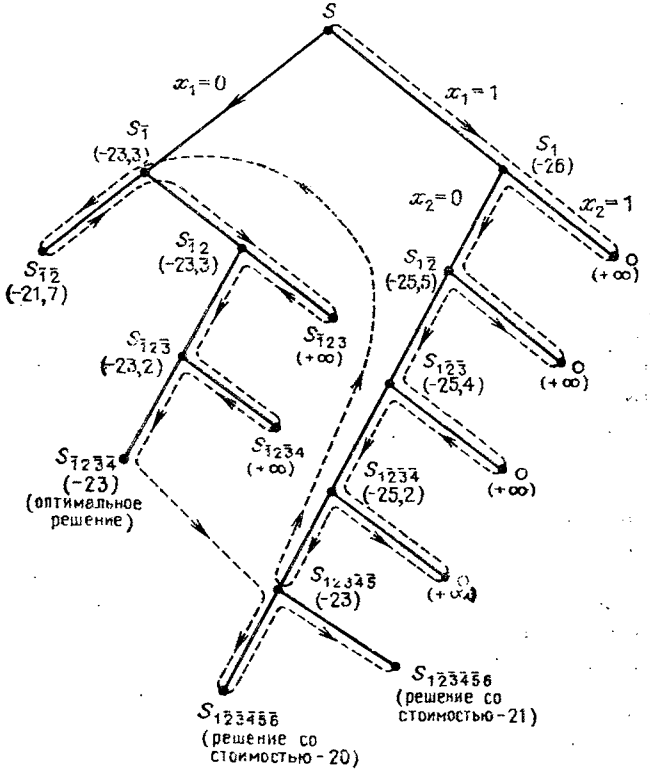


Рис. 4. Пример решения задачи с шестью переменными методом *SEP*. Пунктиром обозначен порядок, в котором исследуются вершины

$(-23{,}3)$. Тогда мы переходим к $S_{\bar{1}2}$ и разделяем его на $S_{\bar{1}2\bar{3}}(-23{,}2)$ и $S_{\bar{1}23}(+\infty)$. Затем переходим к $S_{\bar{1}2\bar{3}}$, которое разделяем на $S_{\bar{1}2\bar{3}\bar{4}}$ (-23: непрерывное решение задачи $P_{\bar{1}2\bar{3}\bar{4}}$ целочисленно: $x_2=1$, $x_5=1$) и на $S_{\bar{1}2\bar{3}4}(+\infty)$ (см. рис. 4).

На этой стадии непрерывное решение задачи $P_{\bar{1}2\bar{3}\bar{4}}$, ограниченной на свободные переменные, приводит к целочисленному реше-

нию: $x_2 = 1$, $x_5 = 1$, со значением оценки -23. Так как не осталось больше ни одного подлежащего исследованию подмножества (висячих вершин) с оценкой, меньшей числа -23, то отсюда можно заключить, что это решение — оптимальное решение задачи. Однако оно, может быть, не единственно, так как есть другое подмножество с оценкой -23, а именно $S_{1\bar{2}\bar{3}\bar{4}\bar{5}}$. Если мы хотим найти все оптимальные решения, то, следовательно, нужно продолжать и разделить это подмножество относительно переменной x_6.

Это приводит к двум подмножествам, каждое из которых содержит единственное решение:

— $S_{1\bar{2}\bar{3}\bar{4}\bar{5}\bar{6}}$, соответствующее решению $x_1 = 1$ (все остальные переменные — нули) со значением -20.

— $S_{1\bar{2}\bar{3}\bar{4}\bar{5}6}$, соответствующее решению $x_1 = 1$, $x_6 = 1$ со значением -21.

Тогда можно утверждать, что найденное решение со значением $z = -23$ есть единственное оптимальное решение, и алгоритм завершен (рис. 4).

Этот пример позволяет оценить эффективность метода. Действительно, на $2^7 - 1 = 127$ вершин, которые теоретически содержатся в полном разветвлении, мы обнаруживаем, что исследованы только 19, что составляет значительную редукцию.

2.7. Компромисс между ценой и эффективностью оценки. Вообще говоря, число подлежащих исследованию вершин тем меньше, чем лучшую аппроксимацию целочисленного оптимума дает применяемая оценка — иначе говоря, чем меньше скачок

$$\min_{\substack{x \in S_i \\ Ax = b}} \{cx\} - f(S_i).$$

Но на практике, вообще говоря, нужно находить разумный компромисс между функциями оценки, доставляющими достаточно грубую аппроксимацию, но вычисляемыми очень быстро, и более точными функциями оценки, но вычисляемыми дольше.

Нужно знать, что не существует никакого удовлетворительного общего метода, который был бы приложим к любой задаче. Этот компромисс должен каждый раз исследоваться заново — для каждого нового типа исследуемых задач,— если есть желание решать их эффективным способом.

§ 3. Методы сечений

3.1. Принцип методов сечений. Идея, на которой основаны все эти методы, состоит в следующем.

Начинаем с решения непрерывной линейной задачи (PL). Если полученное оптимальное решение есть крайняя точка с целыми координатами, то все закончено.

В противном случае легко видеть, что всегда можно усечь область решений (добавляя к задаче дополнительное ограничение) так, чтобы устранить эту (нецелую) крайнюю точку, не исключая

ни одного целого решения. Такое ограничение называется сечением (или отсечением).

Вернемся к примеру из § 1, где непрерывный оптимум представлял собой крайнюю точку: (5, 9, 0). Дополнительное ограничение

$$x_1 \leqslant 5$$

устраняет эту точку, не исключая никаких целых решений. Следовательно, это действительно сечение. Кроме того, любое дополнительное ограничение вида

$$x_1 + x_2 \leqslant \alpha$$

с $5 \leqslant \alpha < 5{,}9$ есть сечение.

(Следовательно, возможных сечений много, и, в принципе, их всегда бесконечно много.)

После того как сечение (или одновременно несколько сечений) добавлено, линейная задача с расширенным набором соответствующих ограничений подлежит новому решению (как непрерывная задача) симплексным методом.

(Важное замечание: так как старое решение останется дуально реализуемым, то выгодно использовать для этого дуальный алгоритм (см. гл. 2, п. 4.1).)

Если оптимальное решение этой новой задачи является целым, то все закончено: получено оптимальное решение целочисленной задачи. В противном случае предыдущее рассуждение можно повторить: мы будем искать новое сечение (может быть, и несколько), которое мы добавим к множеству ограничений; затем расширенная таким образом задача линейного программирования должна заново подвергнуться оптимизации, и т. д.

Если на каждом этапе сечения выбираются корректно, то начальный полиэдр $\mathscr{P}$ будет, таким образом, постепенно уменьшаться, пока не станет совпадать с выпуклой оболочкой целых решений — по крайней мере в окрестности оптимального решения. Тогда непрерывное решение расширенной задачи оказывается целым, и задача будет решена.

Однако ясно, что выбор сечений является определяющим для сходимости метода. Если в предыдущем примере на шаге k выбрать добавляемое сечение типа

$$x_1 + x_2 \leqslant \alpha_k, \quad \alpha_1 = 5{,}8, \quad \alpha_k = \alpha_{k-1} - \frac{1}{10^k},$$

то ясно, что при этих условиях алгоритм не будет сходиться к целому решению.

Напротив, если выбрать в качестве сечений два ограничения

$$x_1 + x_2 \leqslant 5, \quad x_2 \leqslant 4,$$

определяющие (с ограничениями положительности $x_1 \geqslant 0$, $x_2 \geqslant 0$) выпуклую оболочку целых решений задачи (рис. 1 § 1), то мы непосредственно получим целочисленный оптимум $x_1 = 1$, $x_2 = 4$

как непрерывное сечение расширенной линейной задачи

$$\min z = -10x_1 - 11x_2,$$

$$10x_1 + 12x_2 \leqslant 59 \text{ — начальное ограничение,}$$

$$\left.\begin{aligned} x_1 + x_2 &\leqslant 5 \\ x_2 &\leqslant 4 \end{aligned}\right\} \text{ сечения,}$$

$$x_1,\ x_2 \geqslant 0.$$

К сожалению,— и эта трудность существенна — неизвестен систематический метод порождения всех уравнений или неравенств, определяющих выпуклую оболочку множества целых точек, содержащихся в данном выпуклом полиэдре. К тому же они — уравнения и неравенства — могут оказаться чрезвычайно многочисленны, как ясно показывает результат [50, 51, 74]: даже для целочисленной задачи с двумя переменными и одним ограничением всегда можно выбрать коэффициенты так, чтобы число граней выпуклой оболочки целых точек было сколь угодно велико. Следовательно, было бы и дорого, и излишне порождать все грани выпуклой оболочки целых точек: большая часть этих ограничений не принимала бы никакого участия в формировании оптимального решения и не вносила бы никакого вклада в определяемый целочисленный оптимум.

По этим причинам один из наиболее важных результатов целочисленного программирования состоит в том, чтобы ввести явным образом (см. [31]) сечения некоторого специального вида, позволяющие, при некоторых предосторожностях, получить конечную сходимость метода.

3.2. Сечения Гомори. Рассмотрим непрерывную задачу линейного программирования (получаемую исходя из (*PNE*) снятием условий целочисленности)

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ x &\geqslant 0, \end{aligned} \qquad (PL)$$

и пусть B — оптимальная базисная матрица (регулярная квадратная подматрица матрицы A).

Поскольку коэффициенты матрицы A целочисленны, то и определитель $D = |\det(B)|$ — целое число. С другой стороны, умножая систему $Ax = b$ слева на B^{-1}, мы можем выразить любое из базисных переменных x_i ($i \in I$, где I — множество индексов базисных переменных) как функцию свободных внебазисных переменных x_j ($j \in J$, где J — множество индексов свободных переменных) с помощью формулы

$$x_i = \frac{\beta_i}{D} - \sum_{j \in J} \frac{\alpha_{ij}}{D} x_j, \qquad (1)$$

где коэффициенты β_i и α_{ij} — целые.

Напомним, что базисное решение, соответствующее базисной матрице B, имеет вид

$$x_j = 0 \quad \forall j \in J, \qquad x_i = \beta_i/D \ \forall i \in I.$$

Поскольку мы находимся в ситуации, где оптимальное решение не является целым, то одно из переменных — например x_i — дробное.

Выразим тот факт, что мы ищем решение, в котором переменное x_i является целым. Это условие вместе с уравнением (1) дает уравнение конгруентности:

$$\sum_{j \in J} \alpha_{ij} x_j = \beta_i \pmod{D}. \tag{2}$$

Заметим, что мы получили бы эквивалентное отношение конгруентности, умножая обе части равенства (2) одновременно на одно и то же число λ, взаимно простое с D:

$$\sum_{j \in J} (\lambda \alpha_{ij})\, x_j \equiv (\lambda \beta_i) \pmod{D}. \tag{3}$$

Для произвольного целого y обозначим через $|y|_D$ представитель y в отрезке $[0, D-1]$ по модулю D.

(Пример: если $D = 7$ и $y = -2$, то $|y|_D = 5$.)

Полагая тогда $f_j = |\lambda \alpha_{ij}|_D$ (для $j \in J$) и $f_0 = |\lambda \beta_i|_D$, получаем из (3), что существует такое целое s, что

$$\sum_{j \in J} f_j x_j = f_0 + sD.$$

Если s отрицательно, то $f_0 + sD$ необходимо отрицательно, что противоречит тому факту, что $f_j \geqslant 0$ и $x_j \geqslant 0$. Следовательно, s — неотрицательное целое, и для любого решения, в котором переменная x_j является целой (и потому для любого целого решения задачи (PNE)), необходимо выполняется неравенство

$$\sum_{j \in J} f_j x_j \geqslant f_0. \tag{4}$$

С другой стороны, можно заметить, что это неравенство не выполняется как раз в рассматриваемом решении, поскольку оно выделяется условием

$$x_j = 0 \qquad \forall j \in J.$$

Таким образом, неравенство (4) действительно определяет сечение.

Заметим, что некоторый произвол остается в выборе λ для определения сечения, связанного с уравнением (1). Можно предпочитать в ходе поиска такие сечения, которые имеют шанс быть наиболее эффективными среди всех сечений этого типа (которые удаляют как можно большую часть текущего полиэдра). Интересное практически правило состоит в том, чтобы выбирать λ с наибольшей возможной правой частью $f_0 = |\lambda \beta_i|_D$.

Очевидный способ получить это значение λ состоит в переборе всех чисел λ, взаимно простых с D. Но можно также заметить, что если $\delta = \text{н. о. д.}(D, \beta_i)$, то алгоритм Евклида позволяет определить такие два целых числа λ и μ, что

$$\text{н. о. д.}(\lambda, D) = 1, \quad -\lambda\beta_i + \mu D = \delta.$$

Именно это λ и позволяет получить для f_0 наибольшее значение $D - \delta$ (см. [35]).

Важно заметить, что если z — не целое, то можно действовать точно также, чтобы выразить целостность функции z (так как коэффициенты c_j целые, то и $z = cx$ должно быть целым для любого целого решения), что приводит к сечению, связанному с уравнением

$$z = z_B + \sum_{j \in J} \bar{c}_j x_j$$

(где $\bar{c}_j$, $j \in J$,— приведенные цены свободных переменных). На самом деле $z_B = c_B B^{-1} b$ и числа $\bar{c}_j$ являются дробными со знаменателем D.

Выражая таким образом условие целости дробных базисных переменных (включая целевую функцию), мы можем вывести все семейство сечений непосредственно из коэффициентов канонической формы текущей задачи линейного программирования (исходной или расширенной).

Посмотрим теперь, как можно использовать сечения этого вида для реализации алгоритма, если и не эффективного, то по крайней мере сходящегося конечным образом, и в чем состоят предосторожности, которые для этого нужно соблюдать.

3.3. Дуальный алгоритм Гомори. Принцип и варианты. Первая идея реализации алгоритма состоит в том, чтобы полностью реоптимизировать новую задачу линейного программирования каждый раз, когда мы добавляем сечение (или несколько сечений). Для этого используется двойственный симплексный метод (см. гл. 2, § 4) и может потребоваться много итераций (поворотов).

Если всегда оказывается возможным найти сечение вида (4), гарантирующее строгое убывание функции z, то тогда легко представить себе конечный алгоритм. Действительно, пока z не становится целым, всегда можно присоединить сечение, получаемое из формулы

$$z = z_B + \sum_{j \in J} \bar{c}_j x_j.$$

Для такого сечения заметим, что разрешающий элемент обязательно соответствует такому переменному j, что $\bar{c}_j \neq 0$; отсюда следует, что вырождения быть не может, что на каждом шаге z строго убывает и что за конечное число шагов функция z может быть снова сделана целой.

Когда функция z становится целой, мы, добавляя сечение (выражающее целость одного из дробных базисных переменных), застав-

ляем z *строго уменьшаться и получаем заново некоторое дробное* значение z, и процесс продолжается.

Число замен базиса между двумя последовательными целыми значениями функции z конечно, и поэтому ясно, что алгоритм сходится конечным образом (оптимум имеет конечное значение вследствие гипотез § 1).

Трудность состоит именно в том, что не всегда можно найти сечение вида (4), которое, будучи добавлено к задаче, приводит к строгому убыванию функции z.

Если такое происходит для целого значения $z = z_0$, то мы оказываемся в случае вырождения: для некоторого сечения разрешающий элемент соответствует переменному x_j с приведенной ценой $\bar{c}_j$, равной нулю.

Обозначим через $j' \subset J$ *множество индексов базисных* переменных с нулевыми приведенными ценами $\bar{c}_j$.

Тогда необходимо знать, есть ли у многогранника

$$\mathscr{P}' = \{x' \in \mathbf{R}^{n'} \mid A'x' = b,\ x' \geqslant 0\}$$

(где A' — подматрица A, столбцы которой соответствуют $n' = |J'| + |I|$ переменным с пулевой приведенной ценой) целочисленная крайняя точка, или такой точки нет; действительно, если она есть, то эта целочисленная вершина соответствует оптимальному решению задачи (очевидно, с ценой z_0).

В противном случае можно быть уверенным, что никакая комбинация, составленная только из переменных x_j, $j \in J'$, не может обеспечить условия целости для всех базисных переменных, и тогда можно добавить ограничение (сечение)

$$\sum_{j \in J - J'} x_j \geqslant 1.$$

Поворот (разрешение) над этим ограничением необходимо приводит тогда к строгому убыванию функции z, поскольку ее ненулевые коэффициенты соответствуют переменным с ненулевой приведенной ценой $\bar{c}_j$. Тогда все сводится к предыдущей задаче.

Во многих алгоритмах сечений, описанных в литературе ([31, 32, 33, 86, 87, 30]), перечисление вершин многогранника $\mathscr{P}'$ осуществляется перечислением подбазисов, выделенных из A' в соответствии с лексикографическим упорядочением,— классическая техника для исследования случаев вырожденности в линейном программировании (см. гл. 2, п. 2.3).

Однако этот метод, вообще говоря, гораздо дороже по времени вычислений, поскольку он приводит к большому количеству операций разрешения (поворота), которое, разумеется, является конечным, но может оказаться значительным.

Как заметил Гондран [35], именно в этом — без всякого сомнения — и состоит одна из главных причин практической неэффективности большого числа методов сечений; он предпочитает помогать этому методу прямым перечислением целых точек в $\mathscr{P}'$ (мето-

дом типа разветвленного поиска), позволяющим избежать дорогостоящих операций разрешения (см. алгоритм в п. 3.4 ниже).

Поскольку многогранник $\mathscr{P}'$ ограничен, то он содержит не более чем конечное число целых точек, поэтому перечисление возможно, и за конечное число шагов получаем либо оптимальное решение, либо строгое уменьшение функции z. Отсюда и следует конечная сходимость алгоритма.

3.4. Метод убывающих конгруентностей. Вместо того чтобы подвергать задачу полной реоптимизации каждый раз, когда добавляется новое сечение, лучше использовать другую идею, состоящую в осуществлении одной-единственной итерации двойственного симплексного алгоритма.

Предположим, что мы добавили сечение вида (4):

$$\sum_{j\in J} \frac{f_j}{D} x_j \geqslant \frac{f_0}{D},$$

происходящее из уравнения

$$x_i = \frac{\beta_i}{D} - \sum_{j\in J} \frac{\alpha_{ij}}{D} x_j. \qquad (1)$$

Первая итерация двойственного симплексного алгоритма приводит к повороту (разрешению) вокруг элемента $f_{j_0}/D \neq 0$ ($j_0 \in J$ — индекс переменного, возвращающегося в базисные).

Тогда определитель D' новой базисной матрицы равен

$$D' = D \times \frac{f_{j_0}}{D} = f_{j_0} \leqslant D - 1$$

(новый определитель = старый определитель × разрешающий элемент).

Если после осуществления этого поворота возникает решение, не полностью целочисленное, то, записывая условие целочисленности соответствующего базисного переменного, мы получаем уравнение конгруентности того же типа, что и (2), но с группой порядка D', строго меньшего D.

Следовательно, мы видим, что, повторяя операции, состоящие в

— добавлении сечения, выражающего целость базисного переменного,

— осуществлении шага дуального алгоритма,

мы обязательно получим за конечное число шагов дуально реализуемое целое решение.

Это — так называемый метод *убывающих конгруентностей* [35]; таким образом, он позволяет, исходя из любого дуально реализуемого нецелого решения, получить за конечное число шагов дуально реализуемое целое решение. Кроме того, этот метод очень эффективен, так как экспериментально установлено, что число необходимых шагов в среднем порядка $\log_2 D$, если выбирать λ в (3) так, чтобы f_0 принимало максимальное значение.

Применение метода убывающих конгруентностей в ряду алгоритмов сечений может осуществляться следующим образом.

Алгоритм сечений методом убывающих конгруентностей.

Шаг 1. Решение задачи (PL) как непрерывной задачи симплексным алгоритмом. Если полученное решение целое, то КОНЕЦ. Иначе:

Шаг 2. (Редукция задачи.)

Отыскать (с помощью эвристической процедуры) хорошее целочисленное решение. Если его не удается найти — перейти к шагу 3. Иначе: пусть $\hat{z}$ — стоимость этого решения. Целевая функция записывается в канонической форме относительно оптимального базиса

$$z = z_0 + \sum_{j \in J} \bar{c}_j x_j$$

(где z_0 — значение непрерывного оптимума, а J — множество индексов базисных переменных). Следовательно, $\forall j \in J$: $\bar{c}_j \geqslant 0$ (все приведенные цены неотрицательны). Тогда каждая из переменных x_j, удовлетворяющих условию

$$\bar{c}_j > \hat{z} - z_0,$$

должна обращаться в нуль в оптимальном решении и, следовательно, может быть исключена из задачи.

Шаг 3. Пусть $J' = \{j \in J | \bar{c}_j = 0\}$.

Если значение z_0 является целым и если $J' \neq \varnothing$ перейти к шагу 5. Иначе — добавить сечения методом убывающих конгруентностей, пока расширенная задача не будет иметь целое (дуально реализуемое) решение.

Если это решение также и исходно реализуемо (иначе говоря, если $x_j \geqslant 0$ для всех переменных j), то это — оптимальное решение задачи. КОНЕЦ.

Иначе:

Шаг 4. Решение расширенной задачи «как непрерывной» двойственным симплексным алгоритмом.

Если полученный оптимум целый, то это — оптимальное решение задачи. КОНЕЦ.

Иначе — перейти к шагу 3 (после возможного устранения некоторого количества неработающих ограничений — сечений).

Шаг 5. Найти с помощью (неявного) перечисления целое решение задачи $Ax = b$ с $x_j = 0$ для $j \in J - J'$.

Если такое решение существует, то это — оптимальное решение задачи. КОНЕЦ.

Иначе — добавить сечение

$$\sum_{j \in J - J'} x_j \geqslant 1.$$

Осуществить первую итерацию двойственного алгоритма, разрешая относительно этого ограничения, и вернуться к шагу 3.

Конечная сходимость этого алгоритма вытекает из различных сделанных ранее замечаний.

Интересно отметить аналогию между этим алгоритмом и алгоритмом «полностью целочисленно двойственных» сечений Гомори [33]: в обоих случаях получение целого решения рассматривается как приоритетная задача по отношению к получению оптимального решения. Однако они существенно отличаются в работе. В частности, устранено лексикографическое перечисление базисов в случае вырожденности — в соответствии с замечанием, сделанным в п. 3.3.

Для некоторых классов задач — особенно для таких, в которых матрица A имеет не слишком большие миноры (для примера, меньше 100),— этот алгоритм приводит к очень интересным результатам. Например, это относится к задачам покрытия [21], где число сечений, порождаемых алгоритмом, особенно мало (часто меньше десяти) даже для задач большого размера (типичный размер — 2000 переменных, 120 ограничений).

3.5. Пример. В заключение интересно проиллюстрировать работу предыдущего алгоритма на небольшом примере.

Рассмотрим снова пример из § 1 с двумя переменными и единственным ограничением:

$$\min z = -10x_1 - 11x_2,$$
$$10x_1 + 12x_2 \leqslant 59,$$
$$x_1,\ x_2 \text{ целые} \geqslant 0.$$

Так как $-10/10 < -11/12$, то непрерывное решение этой задачи достигается для $x_1 = 5{,}9$, $x_2 = 0$ и соответствующее значение z равно -59.

Симплекс-таблица в канонической форме относительно оптимального базиса $\{x_1\}$ имеет вид

	x_1	x_2	x_3	
минимизируемая функция →	0	+1	+1	+59
ограничение →	1	$\frac{12}{10}$	$\frac{1}{10}$	$\frac{59}{10}$

x_3 — переменное скачка ($x_3 \geqslant 0$), позволяющее представить таблицу в стандартной форме (все ограничения имеют форму равенств).

Поскольку это решение не является целым, то мы применим метод убывающих конгруентностей для получения целого решения. Условие целости для переменного x_1 выводится из уравнения

$$x_1 = \frac{59}{10} - \frac{12}{10}x_2 - \frac{1}{10}x_3$$

и записывается в виде

$$12x_2 + x_3 \equiv 59 \pmod{10}.$$

или

$$2x_2 + x_3 \equiv 9 \pmod{10},$$

откуда получаем сечение

$$2x_2 + x_3 \geqslant 9$$

(следует выбирать для сечения как можно более удаленный f_0; следовательно, здесь берем $\lambda = 1$).

Добавляя переменное скачка $x_4 \geqslant 0$, записываем это сечение в виде

$$-2x_2 - x_3 + x_4 = -9.$$

Добавляя это ограничение к предыдущей таблице, получаем каноническую форму расширенной задачи относительно базиса $\{x_1, x_4\}$:

	x_1	x_2	x_3	x_4	
	0	+1	+1	0	+59
	1	$\frac{12}{10}$	$\frac{1}{10}$	0	$\frac{59}{10}$
разрешающая строка →	0	−2	−1	+1	−9

↑ разрешающий столбец

Этот базис дуально реализуем (все приведенные цены неотрицательны), но не является исходно реализуемым ($x_4 = -9$ не удовлетворяет условиям положительности).

Осуществим один шаг двойственного симплексного алгоритма (см. гл. 2, п. 4.1). Разрешение с помощью второго ограничения показывает, что именно переменное x_2 возвращается в базисные переменные, поскольку

$$\min\left\{-\frac{+1}{-2}, -\frac{+1}{-1}\right\} = \frac{1}{2}.$$

После разрешения получаем новую таблицу (каноническую форму относительно нового базиса (x_1, x_2)):

x_1	x_2	x_3	x_4	
0	0	$+\frac{1}{2}$	$+\frac{1}{2}$	+54,5
1	0	$-\frac{1}{2}$	$\frac{12}{20}$	$\frac{1}{2}$
0	1	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{9}{2}$

которая дает решение: $x_1 = 1/2$, $x_2 = 9/2$.

Это решение не полностью целочисленно (здесь оказывается, что это решение оптимально для расширенной задачи, так как оно одновременно исходно реализуемо и дуально реализуемо; однако, вообще говоря, требуется много итераций дуального алгоритма, чтобы получить новое оптимальное решение расширенной задачи).

Продолжая применять метод убывающих конгруентностей, выразим целость одного из базисных переменных, например x_2.

Имеем

$$x_2 = \frac{9}{2} - \frac{1}{2}x_3 + \frac{1}{2}x_4,$$

что дает уравнение конгруентности

$$x_3 + x_4 \equiv 1 \pmod 2,$$

откуда выводим сечение

$$x_3 + x_4 \geqslant 1.$$

Добавляя это сечение к предыдущей таблице и вводя переменное скачка $x_5 \geqslant 0$, получаем

x_1	x_2	x_3	x_4	x_5	
0	0	$+\frac{1}{2}$	$+\frac{1}{2}$	0	$+54{,}5$
1	0	$-\frac{1}{2}$	$\frac{12}{20}$	0	$\frac{1}{2}$
0	1	$\frac{1}{2}$	$-\frac{1}{2}$	0	$\frac{9}{2}$
0	0	-1	-1	$+1$	-1

Разрешение относительно последней строки приводит к выбору x_3 или x_4 — безразлично — в качестве переменного, являющегося кандидатом на возвращение в базисные переменные. Выбирая, например, x_3, получаем:

x_1	x_2	x_3	x_4	x_5	
0	0	0	0	$+\frac{1}{2}$	$+54$
1	0	0	$\frac{22}{20}$	$-\frac{1}{2}$	1
0	1	0	-1	$\frac{1}{2}$	4
0	0	1	1	-1	1

Полученное на этот раз решение

$$x_1 = 1, \quad x_2 = 4, \quad x_3 = 1$$

является целым.

Кроме того, оказывается, что это решение является также и исходно реализуемым, так как оно положительно (вообще говоря, как мы уже указывали, эта ситуация должна появляться только

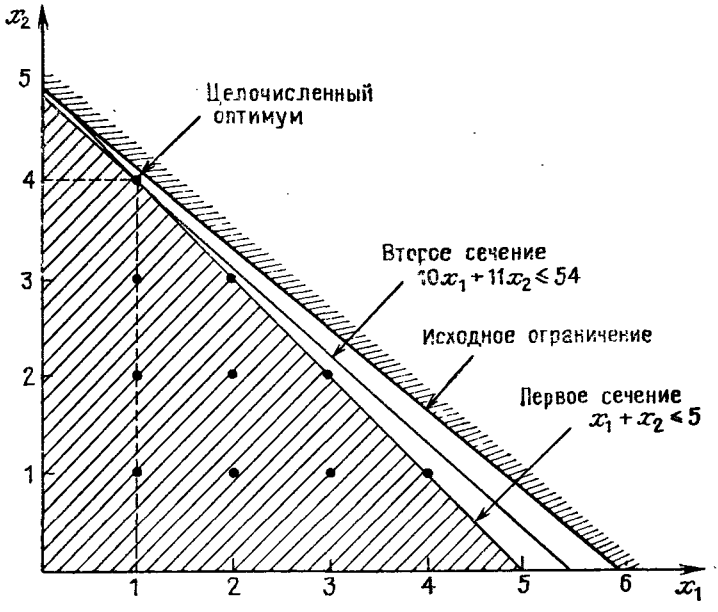


Рис. 5. Два сечения, порожденные алгоритмом, и уменьшенный многоугольник (заштрихован)

после многих итераций дуального алгоритма); следовательно, это — оптимальное решение целочисленной задачи.

Интересно нанести на график на рис. 5 два ограничения, соответствующие сечениям, порожденным таким образом нашим алгоритмом. Первое сечение имеет вид

$$2x_2 + x_3 \geqslant 9.$$

Устраняя x_3, определенное формулой

$$x_3 = 59 - 10x_1 - 12x_2,$$

мы выражаем ограничение в виде функции только от x_1 и x_2, что дает

$$-10x_1 - 10x_2 \geqslant -50.$$

Иначе говоря,

$$x_1 + x_2 \leqslant 5.$$

Второе сечение имеет вид

$$x_3 + x_4 \geqslant 1.$$

Осуществляя подстановки

$$x_4 = -9 + 2x_2 + x_3,$$
$$x_3 = 59 - 10x_1 - 12x_2,$$

мы можем выразить $x_3 + x_4$ как функцию только от x_1 и x_2:

$$x_3 + x_4 = 50 - 10x_1 - 10x_2 + x_3 = 109 - 20x_1 - 22x_2,$$

откуда получаем ограничение

$$10x_1 + 11x_2 \leqslant 54.$$

Эти два ограничения пересекаются в точке $(1, 4)$, как это показано на рис. 5.

Одно из этих ограничений (первое) есть грань выпуклой оболочки целых решений.

§ 4. Целочисленные задачи программирования и кратчайшие пути. Представление с помощью конечных групп

Рассмотрим целочисленную линейную задачу программирования вида

$$z = cx \to \min,$$
$$Ax = b, \qquad (5) \qquad (PNE)$$
$$x \geqslant 0, \quad x \in \mathbf{Z}^n,$$

в которой — мы напомним —

1) политоп $\mathscr{P} = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geqslant 0\}$ предполагается ограниченным и непустым;

2) все коэффициенты c_j, a_{ij}, b_j предполагаются целыми.

Первая гипотеза совершенно не является ограничением. В случае, если политоп $\mathscr{P}$ не ограничен, задача (PNE) тогда и только тогда имеет оптимум на конечном расстоянии, когда не существует целого решения $\bar{x}$, удовлетворяющего задаче

$$A\bar{x} = 0,$$
$$\bar{x} \geqslant 0, \qquad (I)$$
$$c\bar{x} < 0.$$

Мы видим, что для этого достаточно, чтобы задача (I) не имела решений в вещественных числах, что равносильно тому, что задача (PL) без ограничений целости имеет оптимум на конечном расстоянии.

Именно это и будет всегда предполагаться в дальнейшем. Если при этих условиях $\mathscr{P}$ не ограничен, то в случае, когда все коэффициенты c_j положительны, знание мажоранты $\tilde{z}$ оптимума z^* задачи (PNE) позволяет свести дело к ограниченному случаю, добавляя связь типа $z = cx \leqslant \tilde{z}$.

На практике $\bar{z}$ может быть ценой известного априори хорошего решения или может быть получено приближенным методом.

Вторая гипотеза тем более не является ограничением: действительно, с теоретической точки зрения в [38] показано, что в случае произвольных вещественных коэффициентов — при условии, что $\mathscr{P}$ есть ограниченный полиэдр,— всегда можно найти представление с целыми коэффициентами, имеющее то же множество целых решений.

С практической же точки зрения мы всегда используем на компьютере рациональное представление, которое позволяет ограничиться целыми значениями (с помощью умножения на общий знаменатель).

4.1. Эквивалентность с задачей о кратчайшем пути. Покажем сначала, что задача (*PNE*) может быть сведена к задаче о кратчайшем пути в некотором графе $\mathscr{H}' = [H', U']$, содержащем, в принципе, бесконечное число вершин.

Каждой вершине $s \in H'$ соответствует вектор-столбец $s = \begin{bmatrix} s_1 \\ s_2 \\ \cdots \\ s_m \end{bmatrix}$ вида

$$s = \sum_{j=1}^{n} A^j y_j,$$

где y_j — неотрицательные целые и где A^j представляет j-й столбец матрицы A. В частности, H' содержит вершину $0 = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$, получаемую для $y_j = 0$ $(\forall j = 1, \ldots, n)$.

Две вершины $s = Ay$ и $s' = Ay'$ соединены дугой типа k — дугой $(s, s') \in U'$ — тогда и только тогда, когда векторы y и y' отличаются только в компоненте k, причем $y'_k = 1 + y_k$.

Длина, приписываемая этой дуге (s, s'), есть тогда c_k — единичная цена переменного x_k.

(Важное замечание: между двумя данными вершинами s и s' может существовать много дуг различных типов и потому различных длин.)

Если в $\mathscr{H}'$ не существует вершины, соответствующей столбцу $b = \begin{bmatrix} b_1 \\ \cdots \\ b_m \end{bmatrix}$ (правой части (5)), то тогда ясно, что (*PNE*) не имеет решения.

В противном случае любой путь, соединяющий в $\mathscr{H}'$ вершину 0 с вершиной b, соответствует некоторому решению $(y_1, y_2, \ldots, y_n) \geqslant \geqslant 0$, где $\forall k$ число y_k есть число дуг типа k в данном пути. В частности, кратчайший путь между 0 и b соответствует решению $(y_1, \ldots, y_n)$ с минимальной ценой $\sum_{k=1}^{n} c_k y_k$ — следовательно, оптимальному решению задачи (*PNE*).

(Заметим, что $\mathcal{H}'$ не может содержать никакого цикла с отрицательной ценой, поскольку, по предположению, задача (*PNE*) имеет оптимум с конечным значением.)

Проиллюстрируем эту конструкцию с помощью следующего примера:

$$\min(2x_1 + x_2 + 5x_3),$$
$$3x_1 + x_2 = 11,$$
$$-12x_1 + 7x_2 + x_3 = 11,$$
$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_1, \quad x_2, \quad x_3 \in \mathbf{Z}.$$

На рис. 6 сплошными линиями представлена часть соответствующего графа $\mathcal{H}'$.

Если в предыдущем построении перестать принимать во внимание ограничения положительности на переменные y_j, то мы получим новый граф $\mathcal{H} = [H, U]$, имеющий еще больше вершин (действительно, каждая вершина соответствует теперь подмодулю в $\mathbf{Z}^m$,

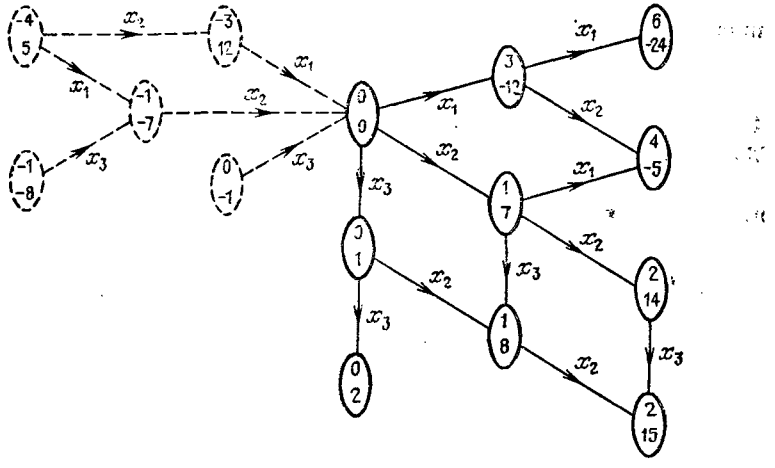


Рис. 6. Пример построения графа $\mathcal{H}'$ (сплошные линии) и графа $\mathcal{H}$ (нужно присоединить пунктирные вершины и дуги)

порожденному столбцами A) и еще больше дуг, но не допускающий никаких новых путей между 0 и b.

Действительно, новый путь между 0 и b должен был бы необходимо проходить через одну из новых вершин — пусть вершину $s = Ay$, где у вектора y есть составляющая $y_j < 0$. Но такая вершина не может быть получена, если исходить из 0, с помощью последовательности вершин

$$s^1 = Ay^1, \quad s^2 = Ay^2, \ldots, s^p = Ay^p$$

с $0 \leqslant y^1 \leqslant y^2 \leqslant \ldots \leqslant y^p$.

(Действительно, напомним, что по построению существование дуги между s^i и s^{i+1} влечет $y^i \leqslant y^{i+1}$.) В предыдущем примере это можно легко проверить, построив некоторые вершины и некоторые дуги из $\mathscr{H}$, не содержащиеся в $\mathscr{H}'$ (пунктирные линии на рис. 6).

Следовательно, после устранения ограничений положительности задача (*PNE*) становится эквивалентной задаче поиска кратчайшего пути (между 0 и b), но на сей раз в расширенном графе $\mathscr{H}$.

К сожалению, на практике эта эквивалентность очень малополезна, так как даже для маленьких задач и даже при ограничении области изменений (ограниченных) переменных число вершин графа $\mathscr{H}$ может оказаться значительным и очень быстро превосходит возможности наиболее эффективных алгоритмов поиска кратчайших путей (не более нескольких тысяч вершин).

Поэтому мы займемся поиском преобразований, позволяющих уменьшить размер этого графа и, если это возможно, ограничиться случаем конечных графов.

4.2. Гомоморфные образы задачи. Мы собираемся изучить тип преобразования, основанный на следующем замечании.

Множество H вершин графа $\mathscr{H}$ изоморфно подмодулю $\mathbf{Z}^m$ (подмодулю, порожденному столбцами матрицы A). Множество H образует группу относительно операции сложения $+$ в $\mathbf{Z}^m$ (в частности, если $s \in H$, $s' \in H$, то $s - s' \in H$). Конечно, речь идет о бесконечной группе.

Пусть G — другая группа (априори произвольная), снабженная операцией сложения (снова обозначенной $+$), и пусть φ — гомоморфизм $H \to G$; иначе говоря — отображение $H \to G$, удовлетворяющее для всех s, s' условию

$$\varphi(s + s') = \varphi(s) + \varphi(s'),$$

где сложение слева есть сложение в H, а справа — в G.

Построим граф $\mathscr{G} = [G, V]$, являющийся образом при гомоморфизме φ графа $\mathscr{H} = [H, U]$ следующим способом:

— *вершины* $\mathscr{G}$ *суть элементы группы* G, *являющиеся образами* под действием φ элементов группы H (так как φ сюръективно, то каждая вершина $\mathscr{G}$ соответствует по крайней мере одной вершине $\mathscr{H}$);

— дуга (типа k) между двумя вершинами t и t' графа $\mathscr{G}$ существует тогда и только тогда, когда между двумя вершинами s и s' в $\mathscr{H}$, такими что $t = \varphi(s)$ и $t' = \varphi(s')$, существует дуга (типа k). Длина этой дуги есть c_k.

Заметим, что несколько дуг типа k с различными концами, входящих в $\mathscr{H}$, могут индуцировать несколько дуг типа k между данными двумя вершинами в $\mathscr{G}$.

Задача *PCCH* ($\mathscr{H}$) определения кратчайшего пути между 0 и b в $\mathscr{H}$ индуцирует тогда задачу об определении кратчайшего пути между $\varphi(0) = 0$ и $\varphi(b)$ в $\mathscr{G}$, или задачу *PCCH* ($\mathscr{G}$).

Заметим, что задача $PCCH(\mathcal{G})$ соответствует задаче

$$\min\left(\sum_{j=1}^{n} c_j x_j\right),$$

$$\sum_{j=1}^{n} \varphi\left(A^j\right) x_j = \varphi(b), \qquad P(G)$$

$$x \geqslant 0, \quad x \in \mathbf{Z}^n,$$

в которой равенства и сложения происходят в группе G.

Это построение представляет интерес, если G — конечная группа (и, если возможно, не слишком большого порядка), так как тогда $PCCH(\mathcal{G})$ есть задача об определении кратчайшего пути в конечном графе, которая может быть решена эффективно с помощью алгоритма кратчайшего пути.

Однако заметим сразу же, что, вообще говоря, между начальной задачей (PNE) и ее образом под действием гомоморфизма — задачей $P(G)$ — эквивалентности нет. Действительно, по построению, любое решение (PNE) есть решение $P(G)$, но обратное, вообще говоря, неверно: $P(G)$ может иметь много решений, не являющихся решениями (PNE).

Задача $P(G)$ называется *ослаблением*, или *релаксацией*, задачи (PNE).

Очевидно, что одна и та же задача (PNE) может допустить много различных ослаблений, в соответствии со способом выбора гомоморфизмов φ.

Для иллюстрации изложенного рассмотрим следующий пример:

$$\begin{gathered} \min:\ 2x_1 + x_2 + 5x_3, \\ 3x_1 + x_2 = 11, \qquad (PNE_1) \\ -12x_1 + 7x_2 + x_3 = 11, \\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_1, \quad x_2, x_3 \in \mathbf{Z}. \end{gathered}$$

Гомоморфизм φ определяется здесь взятием первого условия по модулю 4, а второго условия — по модулю 2; получается новая задача с конечной группой

$$G_1 = \mathbf{Z}_4 \oplus \mathbf{Z}_2$$

($\mathbf{Z}_4$ — группа целых чисел по модулю 4, а $\mathbf{Z}_4 \oplus \mathbf{Z}_2$ обозначает прямую сумму групп $\mathbf{Z}_4$ и $\mathbf{Z}_2$):

$$\begin{gathered} \min(2x_1 + x_2 + 5x_3), \\ 3x_1 + x_2 \equiv 3 \pmod 4, \\ x_2 + x_3 \equiv 1 \pmod 2, \qquad P(G_1) \\ x_1, \quad x_2, \quad x_3 \geqslant 0, \quad x_1, \quad x_2, \quad z_3 \in \mathbf{Z}. \end{gathered}$$

Сама же задача $P(G_1)$ эквивалентна поиску кратчайшего пути между $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ и $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ в графе $\mathcal{G}_1$ на рис. 7.

Этот граф содержит восемь вершин, соответствующих восьми элементам группы G_1:

$$\binom{0}{0}, \binom{1}{0}, \binom{2}{0}, \binom{3}{0}, \binom{0}{1}, \binom{1}{1}, \binom{2}{1}, \binom{3}{1}.$$

Поиск кратчайшего пути между вершинами $\binom{0}{0} = \varphi(0)$ и $\binom{3}{1} = \varphi(b)$ приводит тогда к пути $\binom{0}{0} - \binom{1}{1} - \binom{2}{0} - \binom{3}{1}$ с ценой 3 и соответствующему решению $\bar{x}$:

$$\bar{x}_1 = 0, \quad \bar{x}_2 = 3, \quad \bar{x}_3 = 0.$$

Заметим, что это решение не является решением задачи (PNE_1), так как

$$A\bar{x} = \begin{pmatrix} 3 \\ 21 \end{pmatrix} \neq b = \begin{pmatrix} 11 \\ 11 \end{pmatrix}.$$

Несмотря на отсутствие эквивалентности с исходной задачей, задача $P(G)$ обладает интересными свойствами. С одной стороны, ее, вообще говоря, много проще решить, чем исходную задачу.

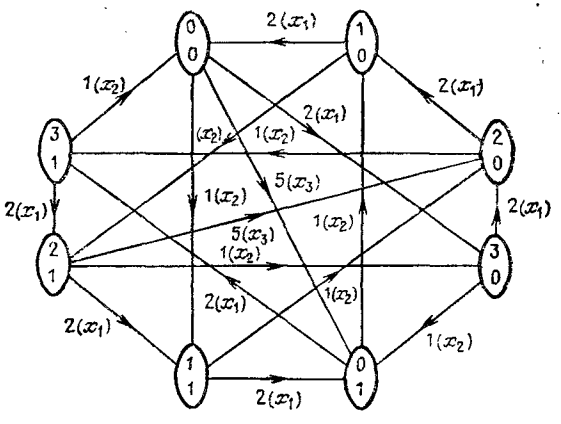


Рис. 7. Граф $\mathcal{G}_1$ с 8 вершинами, связанный с задачей $P(G_1)$. На каждой из дуг указана ее цена и (в скобках) переменное, меняющееся на единицу при переходе по дуге из начала в конец

С другой стороны, притом, что множество решений (PNE) содержится в множестве решений $P(G)$, оказывается, что если оптимальное решение $\bar{x}$ задачи $P(G)$ удовлетворяет ограничениям задачи (PNE), т. е. если $A\bar{x} = b$, то тогда можно быть уверенным, что $\bar{x}$ есть также оптимальное решение задачи (PNE). Отсюда возникает идея поиска оптимального решения (PNE) с помощью попытки построения такого ослабления $P(G)$, чтобы задачи (PNE) и $P(G)$ имели общие оптимальные решения.

Например, рассмотрим другое ослабление задачи (PNE_1), получаемое взятием первого условия по модулю 4, а второго условия — по модулю 6. Получаем задачу в группе $G_2 = \mathbf{Z}_4 \oplus \mathbf{Z}_6$:

$$\begin{gathered} \min(2x_1 + x_2 + 5x_3), \\ 3x_1 + x_2 \equiv 3 \pmod 4, \qquad P(G_2) \\ x_2 + x_3 \equiv 5 \pmod 6, \\ x_1,\ x_2,\ x_3 \geqslant 0, \quad x_1,\ x_2,\ x_3 \in \mathbf{Z}. \end{gathered}$$

Эта задача эквивалентна задаче о кратчайшем пути в графе $\mathcal{G}_2$ с 24 вершинами (это — порядок группы G_2).

Решение этой задачи приводит к решению

$$x_1^* = 2, \quad x_2^* = 5, \quad x_3^* = 0$$

с ценой 9.

Кроме того, убеждаемся, что x^* удовлетворяет ограничениям (PNE_1). Действительно:

$$Ax^* = b = \begin{pmatrix} 11 \\ 11 \end{pmatrix}.$$

Следовательно, это — оптимальное решение исходной задачи (PNE_1).

4.3. Эквивалентная формулировка задачи $P(G)$. Заметим, что гомоморфизм φ группы $\mathbf{Z}^m$ полностью определен своим ядром $\mathrm{Ker}(\varphi) = \{s \in H \mid \varphi(s) = 0\}$, которое является подмодулем $\mathbf{Z}^m$.

Этот подмодуль может, в свою очередь, рассматриваться как подмодуль $\mathbf{Z}^m$, порожденный столбцами некоторой матрицы $R(m \times r)$ максимального ранга r. Иначе говоря, существует такая матрица $R(m \times r)$, что

$$s \in \mathrm{Ker}(\varphi) \Leftrightarrow s = Ry, \qquad y \in \mathbf{Z}^r.$$

Тогда для целого x имеем

$$\begin{gathered} \sum_{j=1}^{n} \varphi(A^j)\, x_j = \varphi(b), \\ \Updownarrow \\ \varphi\left(\sum_{j=1}^{n} A^j x_j - b\right) = 0, \\ \Updownarrow \\ (Ax - b) \in \mathrm{Ker}(\varphi), \\ \Updownarrow \\ \exists y \in \mathbf{Z}^r\colon\ Ax = b + Ry. \end{gathered}$$

Отсюда следует, что задача $P(G)$ равносильна задаче

$$\begin{gathered} \min cx \\ Ax = b + Ry, \\ x \geqslant 0, \quad x \in \mathbf{Z}^n, \qquad P(R) \\ y \in \mathbf{Z}^r, \end{gathered}$$

в которой группа G есть факторгруппа $G = \Pi/\mathrm{Ker}(\varphi)$, причем

$$\mathrm{Ker}(\varphi) = \{s \in \mathbf{Z}^m \mid \exists y \in \mathbf{Z}^r:\ s = Ry\}.$$

Обратно, если задано ослабление целочисленной задачи вида $P(R)$, то всегда можно, используя нормальную форму Смита матрицы R (см. приложение 3), свести задачу к случаю, когда ограничения состоят в системе конгруентностей, иначе говоря, случаем, когда у нас есть уравнение в группе вида $\mathbf{Z}_{\alpha_1} \oplus \ldots \oplus \mathbf{Z}_{\alpha_r}$.

Сейчас мы дадим иллюстрацию этого в частном случае.

4.4. Важный частный случай: группа Гомори. Гомори [34] первым предложил представлять целочисленные задачи линейного программирования в конечной группе.

Идея построения состоит в следующем.

Задача (*PNE*) без условий целости есть просто задача линейного программирования:

$$\begin{aligned} &\min z = cx, \\ &Ax = b, \\ &x \geqslant 0. \end{aligned} \qquad (PL)$$

Ее решение с помощью симплексного алгоритма приводит к оптимальному базису B. Пусть (см. гл. 2)

$$\begin{aligned} A &= [B,\ N], \\ x &= [x_B,\ x_N]^T, \\ c &= [c_B,\ c_N]. \end{aligned}$$

Соответствующее оптимальное решение определяется равенствами

$$x_B = B^{-1}b, \qquad x_N = 0.$$

По определению, нормальная форма Смита матрицы B (с целыми элементами) есть диагональная матрица с неотрицательными элементами по диагонали

$$S = \begin{bmatrix} \varepsilon_1 & & 0 \\ & \varepsilon_2 & \\ & & \ddots \\ 0 & & \varepsilon_m \end{bmatrix}$$

(в которой $\forall i = 1, \ldots, m-1$ число ε_i делит ε_{i+1}), арифметически эквивалентная*) матрице B, иначе говоря, такая, что существуют две квадратные унимодулярные матрицы P и Q ($\det(P) = \pm 1$, $\det(Q) = \pm 1$), удовлетворяющие условию

$$PBQ = S = \begin{bmatrix} \varepsilon_1 & & 0 \\ & \varepsilon_2 & \\ & & \ddots \\ 0 & & \varepsilon_m \end{bmatrix}.$$

*) Две матрицы с целыми элементами называются *арифметически эквивалентными* тогда и только тогда, когда они имеют общие НОД всех порядков (НОД некоторого порядка q есть НОД всех квадратных миноров данной матрицы порядка q). Можно сослаться на приложение 3.

Заметим, что $|\det(B)| = |\det(S)| = \left|\prod_i \varepsilon_i\right|$.

В приложении 3 мы докажем, что нормальная форма Смита единственна (но матрицы P и Q, вообще говоря, не единственны).

Кроме того, существуют алгоритмы, позволяющие строить матрицы P и Q (см. приложение 3).

При этих условиях равенство

$$Ax = Bx_B + Nx_N = b$$

можно переписать в виде

$$PBQQ^{-1}x_B + PNx_N = Pb.$$

Пусть

$$Sy_B + \tilde{N}x_N = \tilde{b},$$

где

$$y_B = Q^{-1}x_B, \quad \tilde{N} = PN, \quad \tilde{b} = Pb.$$

Так как матрица Q унимодулярна, то целость x_B равносильна целости y_B, откуда следует система конгруентностей

$$\tilde{N}x_N \equiv \tilde{b} \mod \begin{bmatrix} \varepsilon_1 \\ \cdots \\ \varepsilon_m \end{bmatrix}.$$

Мы получаем уравнение в конечной группе

$$G = \mathbf{Z}_{\varepsilon_1} \oplus \mathbf{Z}_{\varepsilon_2} \oplus \ldots \oplus \mathbf{Z}_{\varepsilon_m}.$$

Порядок этой группы равен

$$\left|\prod_i \varepsilon_i\right| = |\det(S)| = |\det(B)|.$$

Асимптотическая задача Гомори есть ослабление типа $P(G)$, записываемое в виде

$$\begin{gathered} \min \bar{c}_N x_N = \left(c_N - c_B B^{-1} N\right) x_N, \\ \tilde{N}x_N \equiv \tilde{b} \bmod \begin{bmatrix} \varepsilon_1 \\ \cdots \\ \varepsilon_m \end{bmatrix}, \\ x_N \geqslant 0, \quad x_N \in \mathbf{Z}. \end{gathered} \tag{PA}$$

Заметим, что в этой задаче $\bar{c}_N \geqslant 0$, так как эти величины суть приведенные стоимости в небазисных переменных в оптимуме непрерывной задачи.

Легко видеть, что задача (PA) соответствует образу задачи (PNE) при гомоморфизме φ, ядро которого $\mathrm{Ker}(\varphi)$ есть подмодуль $\mathbf{Z}^m$, порожденный столбцами базиса B (действительно, $\varphi(A^j) = 0$ тогда и только тогда, когда A^j — столбец базиса).

Иначе говоря, задача (PA) эквивалентна задаче

$$\begin{gathered} \min(\bar{c}_N x_N), \\ Ax = b + By, \\ x = [x_B, x_N] \in \mathbf{Z}^n, \qquad x_N \geqslant 0, \\ y \in \mathbf{Z}^m, \end{gathered} \tag{P(B)}$$

т. е. задаче типа $P(R)$ с $R = B$ (видно, что ограничения положительности для базисных переменных устранены).

Следует заметить также, что цены c_N внебазисных переменных заменены приведенными ценами в непрерывном оптимуме: $\bar{c}_N = c_N - c_B B^{-1} N$. Иначе говоря, мы удаляем из цен c_N линейную комбинацию коэффициентов ограничений рассматриваемой задачи. Заметим, что это позволяет получить в задаче (PA) неотрицательные

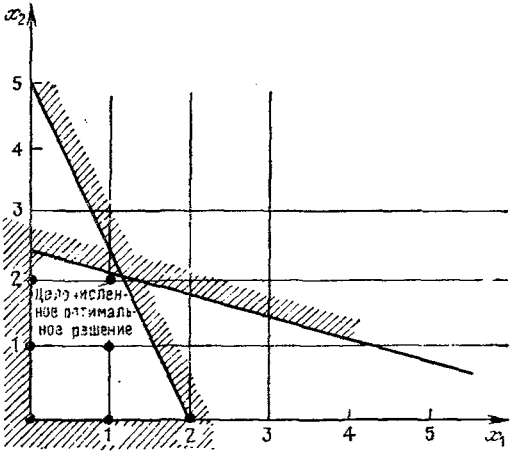


Рис. 8. Множество решений задачи (PNE_2)

цены $\bar{c}_N$, не изменяя цен решений x, удовлетворяющих ограничениям $Ax = b$. Речь идет об основополагающей идее, которая будет обобщена в следующем параграфе.

4.5. Эквивалентные образы одной и той же задачи. Образ $P(G)$ задачи (PNE) не всегда дает столь же интересные сведения, как в примере п. 4.2.

Рассмотрим, например, задачу (см. рис. 8)

$$\begin{aligned} &\max(x_1 + x_2), \\ &5x_1 + 2x_2 \leqslant 10, \\ &3x_1 + 10x_2 \leqslant 24, \\ &x_1, \quad x_2 \geqslant 0, \quad x_1, \quad x_2 \in \mathbf{Z}. \end{aligned}$$

Если представить ее в стандартной форме с переменными скачков x_3 и $x_4 \geqslant 0$, то она примет вид

$$\begin{aligned} &\min(-x_1 - x_2 + 0x_3 + 0x_4), \\ &5x_1 + 2x_2 + x_3 = 10, \\ &3x_1 + 10x_2 + x_4 = 24, \\ &x_1, x_2, x_3, x_4 \geqslant 0, \quad x_1, x_2, x_3, x_4 \in \mathbf{Z}. \end{aligned} \qquad (PNE_2)$$

Если взять первое ограничение по модулю 3, а второе — по модулю 5, то мы получим ослабление $P(G_2)$, где $G_2 = \mathbf{Z}_3 \oplus \mathbf{Z}_5$:

$$\begin{aligned} &\min(-x_1 - x_2 + 0x_3 + 0x_4), \\ &2x_1 + 2x_2 + x_3 \equiv 1 \pmod 3, \qquad P(G_2) \\ &3x_1 + x_4 \equiv 4 \pmod 5, \\ &x_1, x_2, x_3, x_4 \geqslant 0, \quad x_1, x_2, x_3, x_4 \in \mathbf{Z}. \end{aligned}$$

Легко видеть, что задача $P(G_2)$ не имеет оптимального решения на конечном расстоянии. Например, если взять

$$\begin{aligned} x_1 &= 5K, \\ x_2 &= K, \\ x_3 &= 1, \\ x_4 &= 4, \end{aligned}$$

то можно, выбирая целое K достаточно большим, придать функции стоимости сколь угодно малое значение (это означает, что на графе $\mathcal{G}_2$, представляющем $P(G_2)$, есть замкнутые пути с отрицательной стоимостью).

И любое ослабление задачи (PNE_2) приводит к тому же выводу.

Эту трудность можно уменьшить с помощью следующего замечания.

Любое решение задачи (PNE) (как целое, так и нецелое) удовлетворяет ограничениям

$$Ax = b. \tag{5}$$

Рассмотрим вектор $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ множителей, связанных с m ограничениями (5), и пусть P_π — новая задача, получаемая из (PNE) возмущением цен:

$$\begin{aligned} &\min c'x = (c - \pi A)x, \\ &Ax = b, \quad (5) \qquad (P_\pi) \\ &x_j \geqslant 0, \quad x_j \in \mathbf{Z}. \end{aligned}$$

Ясно, что эти задачи (PNE) и (P_π) имеют одни и те же оптимальные решения. Однако всегда можно выбрать π так, что выполняется неравенство $c' = c - \pi A \geqslant 0$.

Действительно, если политоп $\mathcal{P} = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geqslant 0\}$ непуст и ограничен, то «непрерывная» линейная задача

$$\begin{aligned} &\min z = cx, \\ &Ax = b, \\ &x \in \mathbf{R}^n, \quad x \geqslant 0, \end{aligned}$$

допускает оптимум с конечным значением, и, следовательно, ее дуальная задача

$$\begin{aligned} &\max \pi b, \\ &\pi A \leqslant c, \\ &\pi \gtrless 0, \end{aligned}$$

имеет непустое множество решений (теорема двойственности в линейном программировании, см. гл. 2, § 3).

Следовательно, существует такое π, что $c' = c - \pi A \geqslant 0$.

Тогда для решения задачи (PNE) мы выбираем именно этот вектор π и решаем эквивалентную задачу P_π. Ослабление задачи P_π будет тогда иметь вид

$$\min c'x,$$

$$\sum_{j=1}^{n} \varphi\left(A^j\right) x_j = \varphi(b), \qquad P_\pi(G)$$

$$x \geqslant 0, \quad x \in \mathbf{Z}^n.$$

Так как $c' \geqslant 0$, то эта последняя задача имеет оптимальное решение с конечным значением.

4.6. Лагранжево ослабление и определение дуальной задачи. Об ослаблениях задачи (PNE) вида $P_\pi(G)$ можно задать вопрос, не позволяют ли некоторые значения π приблизить оптимум задачи (PNE) лучше, чем другие его значения.

С этой целью начнем с напоминания, что задача (PNE) эквивалентна задаче

$$\begin{aligned} &\min cx, \\ &Ax = b, \\ &x \in X(G), \end{aligned} \tag{5}$$

где $X(G)$ — множество неотрицательных целых решений задачи $P(G)$:

$$X(G) = \left\{ x \in \mathbf{Z}^n \,\middle|\, \sum_{j=1}^{n} \varphi\left(A^j\right) x_j = \varphi(b),\ x \geqslant 0 \right\}.$$

Связывая с ограничениями (5) вектор множителей Лагранжа (дуальные переменные), определим функцию Лагранжа $L(x, \pi)$ формулой

$$L(x, \pi) = cx + \pi(b - Ax)$$

и затем — дуальную функцию

$$w(\pi) = \min_{x \in X(G)} L(x, \pi) = \min_{x \in X(G)} \{(c - \pi A)x + \pi b\} =$$

$$= \pi b + \min_{x \in X(G)} \{(c - \pi A)x\}. \tag{6}$$

Функция $w(\pi)$ может быть легко вычислена (с помощью решения задачи о кратчайшем пути в графе $\mathcal{G}$, связанном с группой G) для каждого π, удовлетворяющего условию $c' = c - \pi A \geqslant 0$.

Свойства функции w хорошо известны (см. гл. 6, § 2):

1) она вогнута, не всюду дифференцируема;

2) она представляет собой миноранту величины $z = cx$ для произвольного целого решения задачи (PNE);

3) если $\bar{x}$ — такой элемент $X(G)$, для которого минимум в (6) достигается, иначе говоря, если

$$(c - \pi A)\,\bar{x} = \min_{x \in X(G)} \{(c - \pi A)\,x\},$$

то $b - A\bar{x}$ — субградиент функции w в точке π.

Эти свойства позволяют решить дуальную задачу $D(G)$, которая, по определению, имеет вид

$$\max w(\pi), \quad \pi \in \mathbf{R}^m, \qquad D(G)$$

с помощью методов субградиента (см. гл. 4, § 4).

Именно оптимальное значение $w(\pi^*)$ задачи $D(G)$ и позволяет наилучшим образом приблизить (снизу) оптимум задачи (PNE) при ослаблении в данной группе G. Заметим, что это свойство позволяет использовать дуальные функции в качестве функций оценки в ходе процедур разветвления с помощью разделения и оценки (см. § 2 этой главы). Многочисленные задачи целочисленного программирования могут быть решены именно этим способом (см. [47, 27, 24]).

В связи с этим могут быть сделаны два особенно важных наблюдения:

— если $\bar{x}$ — такой элемент $X(G)$, для которого минимум в (6) достигается, то $\bar{x}$ — оптимальное целочисленное решение (возмущенной) задачи

$$\min cx, \quad Ax = A\bar{x}, \quad x \geqslant 0, \quad x \in \mathbf{Z}^n.$$

Это может оказаться интересным для приближенного решения задач (когда правые части ограничений в (PNE) не являются предписанными значениями) (см. гл. 6, п. 2.11);

— если, кроме того, $\bar{x}$ удовлетворяет условию $A\bar{x} = b$, то $\bar{x}$ есть (целочисленное) оптимальное решение (P). Именно эту ситуацию мы и пытаемся получить, когда хотим найти точное решение задачи (PNE).

Сейчас мы опишем основные методы, которые могут быть предложены для решения задач целочисленного программирования с использованием ослаблений в конечных группах — ослаблений типа $P(G)$.

4.7. Алгоритм 1. Решение асимптотической задачи [34]. Гомори [34] первым исследовал применение конечных групп для решения задач целочисленного программирования. Он предложил решение асимптотической задачи (см. п. 4.4)

$$\min \tilde{c}_N x_N, \quad \tilde{N} x_N \equiv \tilde{b} \mod \begin{bmatrix} \varepsilon_1 \\ \cdots \\ \varepsilon_m \end{bmatrix}, \quad x_N \geqslant 0,$$

иначе говоря, задачи о кратчайшем пути в графе $\mathcal{G}$, связанном с группой Гомори G.

Если оптимальное решение x_N задачи (PA) удовлетворяет условию

$$x_B = B^{-1}b - B^{-1}Nx_N \geqslant 0,$$

то $x = [x_B, x_N]$ есть оптимальное решение задачи (PNE), и все закончено.

В противном случае мы отыскиваем другой кратчайший путь, затем, возможно, третий и т. д. Первое из полученных решений, удовлетворяющее ограничениям положительности для x_B, есть оптимальное решение задачи (PNE).

Хотя и существуют алгоритмы, позволяющие последовательно определить k первых кратчайших путей, но рассматриваемый метод применим только к графам $\mathcal{G}$ малых размеров (не более чем нескольких десятков вершин).

В связи с этим предположим, что оптимальный базис B имеет очень маленький детерминант. Однако в этом случае задачи равным образом хорошо решаются и другими методами (например, методами сечений, см. § 3).

Следуя Гомори, отметим, что поиски решений особенно тесно связаны с решением асимптотической задачи

— для получения хороших оценок в методах разветвления с помощью разделения и оценки (см. § 2);

— для построения новых сечений, позволяющих увеличить эффективность методов сечений (см. § 3).

Однако недавно у исследователей возник большой интерес в связи с появлением методов, использующих более широкий класс ослаблений и позволяющих — по крайней мере теоретически — построить ослабление с нулевым скачком двойственности. Эти алгоритмы мы сейчас опишем и приведем некоторые характерные примеры.

4.8. Алгоритм 2 [12]. Этот метод не использует лагранжева ослабления и не решает дуальную задачу.

Предположим, что $c \geqslant 0$ (мы видели, что к этому случаю всегда можно свести задачу), иначе говоря, что любая задача $P(G)$ с

$$G = \mathbf{Z}_{\alpha_1} \oplus \mathbf{Z}_{\alpha_2} \oplus \ldots \oplus \mathbf{Z}_{\alpha_p}$$

имеет оптимум с конечным значением.

На некотором этапе задача $P(G)$ решается с

$$G = \mathbf{Z}_{\alpha_1} \oplus \ldots \oplus \mathbf{Z}_{\alpha_p}.$$

Если оптимальное решение $\bar{x}$ задачи $P(G)$ удовлетворяет ограничениям $A\bar{x} = b$, то все конечно: $\bar{x}$ есть оптимальное решение (PNE).

В противном случае обозначим через $u = (u_1, \ldots, u_m)$ такой целочисленый вектор (строку) и через α_{p+1} такое целое число, что

$$u(A\bar{x} - b) \not\equiv 0 \pmod{\alpha_{p+1}}.$$

Ясно, что конгруентность

$$uAx \equiv ub \pmod{\alpha_{p+1}}$$

выполняется для любого целого решения (PNE), но не для решения $\bar{x}$.

Добавляя эту конгруентность, мы получим новое ослабление $P(G')$ в группе $G' = G \oplus \mathbf{Z}_{\alpha_{p+1}}$ бо́льшего порядка, и притом такой, что $X(G') \not\subseteq X(G)$. Алгоритм продолжается тогда разрешением задачи $P(G')$, и т. д.

Конечную сходимость метода легко доказать. Так как $\mathcal{P}$ ограничен, то каждая переменная x_j ограничена, следовательно, $X(G)$ имеет конечное число элементов. Так как на каждой итерации добавление новой конгруентности удаляет по крайней мере одно решение, то алгоритм должен прийти к завершению за конечное число итераций.

Практический способ выбрать вектор u и число α_{p+1} состоит в следующем (см. [12]): нужно взять

α_{p+1} — наименьшее целое, не являющееся делителем всех компонент $A\bar{x} - b$;

$u_i = 0 \ \forall i$ кроме $u_{i_0} = 1$ для одной из компонент i_0 вектора $A\bar{x} - b$, которая не делится на α_{p+1}.

Заметим в заключение, что эффективность этого алгоритма может быть увеличена применением лагранжевой двойственности.

Действительно, решая дуальную задачу $D(G)$, соответствующую ослаблению $P(G)$ (см. п. 4.6), получаем векторы $A\bar{x} - b$, компоненты которых сколь угодно малы, что позволяет выбрать α_{p+1} сколь угодно малым. Таким образом, существенно уменьшается порядок групп, в которых приходится работать. Кроме того, нет необходимости получать точный оптимум дуальной задачи и можно удовлетвориться хорошим приближенным решением. Поэтому особенно рекомендуется использование алгоритма субградиента для решения дуальной задачи (см. гл. 4, § 3).

4.9. Алгоритм 3 [11]. Этот алгоритм использует ослабление (PNE) вида

$$\begin{gathered} \min cx, \\ Ax = b + Ry, \\ x \geqslant 0, \qquad x \in \mathbf{Z}^n, \\ y \in \mathbf{Z}^m, \end{gathered} \qquad P(R)$$

где матрица R равна $B\Delta$, причем B — матрица оптимального базиса непрерывной линейной задачи, а Δ — диагональная матрица вида

$$\Delta = \begin{bmatrix} \delta_1 & & 0 \\ & \delta_2 & \\ & & \ddots \\ 0 & & \delta_m \end{bmatrix}$$

(δ_i — неотрицательные целые).

Если нормальная форма Смита матрицы $B\Delta$ равна (см. приложение 3)

$$P(B\Delta)Q = \begin{bmatrix} \varepsilon_1 & & 0 \\ & \varepsilon_2 & \\ & & \ddots \\ 0 & & & \varepsilon_m \end{bmatrix}$$

(P и Q унимодулярны), то задача $P(R)$ эквивалентна задаче $P(G)$ в группе $\mathbf{Z}_{\varepsilon_1} \oplus \mathbf{Z}_{\varepsilon_2} \oplus \ldots \oplus \mathbf{Z}_{\varepsilon_m}$, порядок которой равен произведению $\det(B) \times \det(\Delta)$.

Заметим, что для $\Delta = I$ группа G есть группа Гомори (см. п. 4.4).

Белл [11] показал, что если множество решений задачи (PNE) ограничено, то (благодаря выбору достаточно больших δ_i) всегда существует такая матрица

$$\Delta^0 = \begin{bmatrix} \delta_1^0 & & 0 \\ & \ddots & \\ 0 & & \delta_m^0 \end{bmatrix},$$

что решение дуальной задачи $D(G)$ либо приводит к целому оптимальному решению (PNE), либо позволяет доказать, что (PNE) не имеет решения (кроме того, это свойство остается справедливым для любой матрицы $\Delta > \Delta^0$).

С другой стороны, у Белла [11] можно найти несколько простых правил, позволяющих изменить числа δ_i, если решение задачи $D(G)$ не привело к целочисленному оптимальному решению (PNE) (эти правила не требуют необходимо точного решения двойственной задачи).

Алгоритм можно инициализировать, беря $\Delta = I$, т. е. используя группу Гомори.

4.10. Алгоритм 4 [14]. Предположим для простоты, что все переменные задачи (PNE) двузначны ($x_j = 0$ или 1, $\forall j$). Известно, что если область решений ограничена, то всегда можно свести задачу к этому случаю (см. п. 2.1).

На некотором шаге рассмотрим ослабление $P(G)$ задачи (PNE), где G — группа типа $\mathbf{Z}_{\alpha_1} \oplus \ldots \oplus \mathbf{Z}_{\alpha_p}$ (вначале можно взять $G = \mathbf{Z}_1$ или — если ее легко найти — группу Гомори). Обозначим через $X(G)$ множество решений $P(G)$. Так как $X(G) \subset \{0, 1\}^n$, то это — конечное множество, которое можно занумеровать конечным индексным множеством T:

$$X(G) = \{x^t \mid t \in T\}.$$

Решение дуальной задачи $D(G)$ эквивалентно тогда задаче линейного программирования

$$w(\pi^*) = \max v,$$
$$\pi(b - Ax^t) + cx^t \geqslant v \qquad \forall t \in T,$$
$$\pi \in \mathbf{R}^m, \qquad \pi \geqslant 0.$$

Связывая с каждым ограничением $t \in T$ из (L) дуальное переменное $\lambda_t \geqslant 0$, мы можем записать дуальную задачу в виде

$$\min \sum_{t \in T} (cx^t)\lambda_t,$$

$$\sum_{t \in T} (Ax^t)\lambda_t = b,$$

$$\sum_{t \in T} \lambda_t = 1,$$

$$\lambda_t \geqslant 0 \qquad \forall t \in T.$$

Замечание 1. Если ввести в задаче (DL) условие целости на переменные λ_t ($\forall t \in T$: $\lambda_t = 0$ или 1), то мы получим задачу, эквивалентную первоначально поставленной целочисленной задаче (PNE).

Замечание 2. (DL) имеет, вообще говоря, значительное число переменных, но тем не менее может быть решена средствами обобщенного линейного программирования (порождение столбцов) (см. гл. 8, § 1).

Если (L) не имеет оптимума на конечном расстоянии, то (DL) не имеет решения, а вследствие этого, согласно замечанию 1, и задача (PNE) не имеет решения, и работа алгоритма завершена.

В противном случае обозначим через (π^*, v^*) оптимальное решение задачи (L), и пусть λ^* — соответствующее оптимальное дуальное решение.

Предположим сначала, что λ^* оказалось целым, иначе говоря, что все его компоненты — нули, кроме компоненты $k \in T$, для которой $\lambda_k = 1$. Тогда, согласно замечанию 1, элемент $x^k \in X(G)$ есть оптимальное решение (PNE), и алгоритм завершен.

Предположим, следовательно, что λ^* имеет дробные компоненты, и пусть $T^* \subset T$ — множество индексов базисных переменных задачи (DL).

Так как соответствующие ограничения в (L) выполнены, то $\forall t \in T^*$ имеем

$$w(\pi^*) = \pi^*(b - Ax^t) + cx^t.$$

Всегда можно предполагать, что

$$Ax^t \neq b \qquad \forall t \in T^*.$$

Действительно, если бы существовал такой $k \in T^*$, что $Ax^k = b$, то выполнялось бы $w(\pi^*) = cx^k$ и, следовательно, x^k был бы оптимальным целочисленным решением задачи (PNE).

Можно представить себе два случая.

Случай 1. Значение $w(\pi^*) = v^*$ дробное. Тогда мы добавляем к $P(G)$ конгруентность

$$(\pi^* A)x \equiv \pi^* b \pmod 1.$$

Так как π^* имеет дробные компоненты, то можно ввести q — такое наименьшее натуральное число, что $q\pi^*$ целое.

Тогда предыдущая конгруентность эквивалентна конгруентности

$$(q\pi^* A)x \equiv (q\pi^*)b \pmod{q}.$$

Эта последняя конгруентность при добавлении к $P(G)$ позволяет исключить по крайней мере все решения x^t $(t \in T^*)$.

Действительно, так как $w(\pi^*)$ дробное (кратное $1/q$) и так как cx^t целое, то

$$|\pi^*(b - Ax^t)| = p/q$$

(p натуральное, не кратное q).

Отсюда следует, что $\forall t \in T^*$

$$\pi^*(Ax^t - b) \not\equiv 0 \pmod{1}.$$

Рассматривая тогда новую группу

$$G' = G \oplus \mathbf{Z}_q,$$

образуем новое ослабление $P(G')$ с

$$|X(G')| < |X(G)|,$$

и алгоритм продолжается.

Случай 2. Значение $w(\pi^*)$ целое. Тогда мы выразим условия целости переменных λ_i задачи (DL) в построении группы Гомори H, связанной с оптимальным базисом (DL).

Как мы уже видели в п. 4.4, это сводится к рассмотрению гомоморфизма h группы $\mathbf{Z}^{m+1}$, определенного формулой

$$h\begin{pmatrix} Ax^t \\ 1 \end{pmatrix} \equiv 0$$

тогда и только тогда, когда λ_i принадлежит оптимальному базису задачи (DL).

Добавляя тогда к $P(G)$ уравнение

$$h\begin{pmatrix} Ax \\ 1 \end{pmatrix} = h\begin{pmatrix} b \\ 1 \end{pmatrix} \quad \text{(равенство в группе } H)$$

(которое, как мы видели в п. 4.4, сводится к системе конгруентностей), исключаем по крайней мере все решения x^t для $t \in T^*$.

Тогда мы построим также новое ослабление $P(G')$ с $G' = G \oplus H$, удовлетворяющее условию $|X(G')| < |X(G)|$, и алгоритм продолжается.

Конечная сходимость описанной выше процедуры к целочисленному оптимальному решению (PNE) вытекает из того факта, что мощность множества решений последовательных ослабленных задач убывает строго. С другой стороны, можно показать (см. [14]), что эта процедура позволяет построить аппроксимацию выпуклой оболочки целочисленных решений задачи (PNE) в окрестности оптимального решения.

Неудобство этого метода состоит в том, что (в отличие от алгоритма 2 п. 4.8) он требует точного решения задач $D(G)$ и (DL).

С другой стороны, порядок групп может расти очень быстро, что может привести к численным трудностям в приложении к задачам большого размера.

СПИСОК ЛИТЕРАТУРЫ

1. Abadie J. (1969), Une méthode arborescente pour les programmes non linéares partiellement discrets, *Revue Fr. Informatique, Recherche Opérationnelle* 3, p. 24-50.
2. Abadie J. (1970), *Integer and Nonlinear Programming*, North Holland, Amsterdam.
3. Arditti D., Minoux M. (1981), Un algorithme de détermination de partition utilisant la dualité lagrangienne, *Actes regroupés des journées de classification ae Toulouse* (mai 1980) *et de Nancy* (juin 1981). I. C. Lerman ed.
4. Bachem A., Schrader R. (1980), Minimal inequalities and subadditive duality, *S.I.A.M. Journal on Control and Optimization*, vol. 18, n° 4, p. 437-443.
5. Balas E. (1965), An additive algorithm for solving linear programs with zero-one variables, *Operations Research* 13, 4, p. 517-546.
6. Balas E. (1971), A duality theorem and an algorithm for mixed integer nonlinear programming, *Linear Algebra and Appl.* 4, p. 341-352.
7. Balas E., Ho A. (1980), Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study, *Mathematical Programming* 12, p. 37-60.
8. Balas E., Martin C. H. (1980), Pivot and Complement: A heuristic for 0-1 programming, *Management Science*, vol. 26, n° 1, p. 86-96.
9. Balas E., Zemel E. (1980), An algorithm for large zero-one knapsack problems, *Operations Research*, vol. 28, n° 5, p. 1130-1154.
10. Balinsky M. L. (1965), Integer Programming: Methods, Uses, Computation, *Management Science* 12, p. 253-313.
11. Bell D. E. (1976), Constructive group relaxations for integer programs, *S.I.A.M. Journal Appl. Math.*, vol. 30, n° 4, p. 708-719.
12. Bell D. E. (1977), A Simple algorithm for integer programs using group constraints, *Operations Research Quarterly*, vol. 28, n° 2, p. 453-458.
13. Bell D. E. (1979), Efficient group cuts for integer programs, *Mathematical Programming* 17, n° 2, p. 176-183.
14. Bell D. E., Shapiro J. F. (1977), A Convergent duality theory for integer programming, *Operations Research*, vol. 25, n° 3, p. 419-434.
15. Benders J. F. (1962), Partitioning Procedures for Solving Mixed Variables Programming Problems, *Numerische Mathematik* 4, p. 238-252.
16. Bertier P., Roy B. (1964), Une procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire, *Cahiers du Centre de'Études de Recherche Opérationnelle*, 6, p. 202-208.
17. Bradley G. H. (1971), Transformation of integer programs to knapsack problems, *Discrete Mathematics* 1, p. 29-45.
18. Crowder H., Padberg M. W. (1980), Solving large-scale symmetric travelling salesman problems to optimality, *Management Science* 26, p. 495-509.
19. Dakin R. J. (1965), A tree search algorithm for mixed integer programming problems, *Computer journal*, 8, 3, p. 250-255.
20. Dantzig G. B. (1959), Note on solving Linear Programs in Integers, *Nav. Res. Log. Quart.* 6, p. 75-76.
21. Delorme J. (1974), Contribution à la résolution du problème de recouvrement: méthodes de troncatures, *Thèse Doct. Ing., Université Paris VI.*
22. Fayard D., Plateau G. (1972), Résolution du problème du Knapsack en variables bivalentes. Comparaison des méthodes, *Bulletin Dir. Et. Rech. EDF*, série C, n° 1, p. 5-42, paru dans *Mathematical Programming* 8, p. 272-307, (1975).

23. Fisher M. L. (1976), A dual algorithm for the one-machine scheduling problem, *Mathematical Programming* 11, p. 229-251.
24. Fisher M. L. (1981), The Lagrangian relaxation method for solving integer programming problems, *Management Science*, vol. 27, n° 1, p. 1-18.
25. Fisher M. L., Northup W. D., Shapiro J. F. (1975), Using duality to solve discrete optimization problems: Theory and computational experience, *Mathematical Programming Study* 3, p. 56-94.
26. Fisher M. L., Shapiro J. F. (1974), Constructive duality in integer programming, *S.I.A.M. J. Appl. Math.* 27, p. 31-52.
27. Garfinkel R. S., Nemhauser G. L. (1972), *Integer programming*, Wiley and Sons ed., 390 p.
28. Geoffrion A. M. (1974), Lagrangian relaxation and its uses in integer programming, *Mathematical Programming Study* 2, p. 82-114.
29. Geoffrion A. M., Marsten R. E. (1972), Integer programming algorithms: a framework and state-of-the-art survey, *Management Science* 18, p. 465-491.
30. Glover F. (1965), A Multiphase-Dual Algorithm for the Zero one Integer Programming Problem, *Operations Research* 13, p. 879 919.
31. Gomory R. E. (1958), Outline of an algorithm for integer solutions to linear programs, *Bull. Am. Math. Soc.* 64, p. 275-278.
32. Gomory R. E. (1963*a*), An algorithm for integer solutions to linear programs, in: *Recent advances in mathematical programming* (Graves and Wolfe eds.), p. 269-302.
33. Gomory R. E. (1963*b*), All-Integer Programming Algorithm, in: *Industrial Scheduling* (J. F. Muth, G. L. Thompson eds.), Prentice Hall, p. 193-206.
34. Gomory R. E. (1965), On the relation between integer and non-integer solutions to linear programs, *Proc. Nat. Acad. Sci.*, 53, p. 260-265.
35. Gondran M. (1973), Un outil pour la programmation en nombres entiers. La méthode des congruences décroissantes, *Rev. Fr. Automatique, Informatique, Rech. Opérationnelle* 3, p. 35-54.
36. Gondran M. (1974), Problèmes combinatoires et programmation en nombres entiers, *Thèse Doct. ès Sc.*, Université Paris VI.
37. Gondran M., Minoux M. (1979), *Graphes et algorithmes*, Collection de la Direction des Études et Recherches EDF, Eyrolles, Paris.
38. Gould F. J., Rubin D. S. (1973), Rationalizing discrete programs, *Operations Research* 21, p. 343-345.
39. Greenberg H. (1971), *Integer Programming*, Academic Press, New York.
40. Hammer P. L. (1975), Boolean elements in combinatorial optimization, in: *Combinatorial Programming: methods and applications*, (B. Roy ed.), Reidel, p. 67-92.
41. Hammer P. L., Johnson E. L., Korte B. H. (1979), *Discrete Optimization*, vols 1 et 2. North Holland, Amsterdam.
42. Hammer P. L., Johnson E. L., Korte B. H., Nemhauser G. L. (1977), *Studies in Integer Programming*, North Holland.
43. Hansen K. H., Krarup J. (1974), Improvements of the Held-Karp Algorithm for the Symmetric Travelling Salesman Problem, *Mathematical Programming* 7, p. 87-96.
44. Hansen P. (1971), Pénalités additives pour les programmes en variables zéro-un, *C. R. Académie Sc.*, Paris, 273, p. 175-177.
45. Hansen P. (1975), Les procédures d'exploration et d'optimisation par séparation et évaluation, in: *Combinatorial Programming: methods and applications*, (B. Roy ed.), Reidel, p. 29-65.
46. Hansen P. (1977), Methods of nonlinear 0-1 programming, in: *Discrete optimization II*, (P. L. Hammer, E. L. Johnson, B. H. Korte eds.), North Holland, 1979, p. 53-70.
47. Held M., Karp R. M. (1971), The traveling salesman problem and minimum spanning trees: part II, *Mathematical Programming* 1, p. 6-25.
48. Herve P. (1967), Résolution des programmes linéaires à variables mixtes par la procédure SEP, *METRA*, vol. 6, n° 1.
49*. Hu T. C. (1969), *Integer programming and network flows*, Addison Wesley Publish. Corp.

50. Jeroslow R. G. (1969), On the unlimited number of faces in integer hulls of linear problems with two constraints, *Techn. Rep.* n° 67, Dept of Operations Research, Cornell University.
51. Jeroslow R. G. (1971), Comments on Integer Hulls of Two linear constraints, *Operations Research* 19, p. 1061-1069.
52. Jeroslow R. G. (1974), Trivial integer programs unsolvable by Branch and Bound, *Mathematical Programming* 6, p. 105-109.
53. Jeroslow R. G. (1977), An introduction to the theory of cutting planes, in: *Discrete optimization II*, (P. L. Hammer, E. L. Johnson, B. H. Korte eds.), North Holland, 1979, p. 71-96.
54. Jeroslow R. G. (1978), Cutting-plane theory: Algebraic methods, *Discrete Mathematics* 23, p. 121-150.
55. Johnson E. L. (1973), Cyclic groups, cutting planes and shortest paths, in: *Mathematical Programming*, (T. C. Hu, S. Robinson eds.), Academic Press.
56. Land Al. H., Doig A. G. (1960), An automatic method for solving discrete programming problems, *Econometrica* 28, p. 497-520.
57. Lauriere M. (1978), An algorithm for the 0/1 Knapsack problem, *Mathematical Programming* 14, p. 1-10.
58. Lawler E. L. (1976), *Combinatorial Optimization: Networks and Matroïds*, Holt, Rinehart and Winston.
59. Lawler E. L., Wood D. E. (1966), Branch and bound methods: a survey, *Operations Research* 14, p. 699-719.
60. Legendre J. P., Minoux M. (1977), Une application de la notion de dualité en programmation en nombres entiers: sélection et affectation optimales d'une flotte d'avions, *R.A.I.R.O.*, vol. 11, n° 2, p. 201-222.
61. Maculan N. (1978), *Programaçao linear inteira*, Publication COPPE, PDD 17/78, Université Fédérale de Rio de Janeiro, Brésil, 212 p.
62. Minoux M. (1975), Résolution des problémes de multiflots en nombres entiers dans les grands réseaux, *RAIRO*, vol. 3, p. 21-40.
63. Minoux M. (1975), Plus court chemin avec contraintes, algorithmes et applications, *Annales des Télécommunications* 30, n° 11-12, p. 383-394.
64. Mitten L. G. (1970), Branch and bound methods: general formulation and properties, *Operations Research* 18, p. 24-34.
65. Morin T. L., Marsten R. E. (1976), Branch and Bound Strategies for Dynamic Programming, *Operations Research* 24, p. 611-627.
66. Mulvey J. M., Growder H. P. (1979), Cluster analysis: an application of Lagrangian relaxation, *Management Science*, vol. 25, n° 4, p. 329-340.
67. Murty K. G. (1976), *Linear and Combinatorial Programming*, Wiley, New York.
68. Padberg M. W., Rao M. R. (1979), The Russian method for linear inequalities and linear optimization, *GBA*, *New-York University*, (Version révisée: juin 1980).
69. Padberg M. W., Rao M. R. (1980), The Russian method and integer programming, *GBA*, *New-York University*.
70. Padberg M. W., Rao M. R. (1981), The Russian method for linear inequalities III: bounded integer programming, *Rapport de Recherche* n° 78, INRIA (France).
71. Plane D. R., McMillan C. Jr. (1971), *Discrete Optimization*, Prentice Hall.
72. Ross G. T., Soland R. M. (1975), A Branch and Bound Algarithm for the Generalized Assignmet Problem, *Mathematical Programming* 8, p. 91-103.
73. Roy B., Bertier P., Nghiem P. T. (1965), Programmes linéaires en nombres entiers et procédure SEP, *Metra*, vol. 4, n° 3
74. Rubin D. S. (1970), On the unlimited number of faces in integer hulls of linear programs with a single constraint, *Operations Research* 18, p. 940-946.
75. Saaty T. L. (1970), *Optimization in Integers and Related Extremal Problems*, McGraw-Hill, New York.
76. Salkin H. M. (1975), *Integer Programming*, Addison Wesley Publish. Company, Reading, Mass.

77. Serreault J. Y., Minoux M (1980), Le programme d'admissibilité avec contraintes de sécurité et coûts de mutation: application aux réseaux de transmission mixtes, *Annales des Télécommunications*, tome 35, n° 1-2, p. 23-38.
78. Shapiro J. F. (1968), Dynamic Programming Algorithm for the Integer Programming Problem-I: The Integer Programming Problem viewed as a Knapsack-type Problem, *Operations Research* 16, p. 103-121.
79. Shapiro J. F. (1971), Generalized Lagrange Multipliers in Integer Programming, *Operations Research* 19, p. 68-76.
80. Shapiro J. F. (1979), A survey of Lagrangian tecnıques for discrete optimization, *Annals Discrete Math.*, vol. 5, p. 113-138.
81. Shapiro J. F. (1979), *Mathematical programming: structures and algorithms*, Wiley, New York.
82. Taha H. A. (1975), *Integer Programming, Theory, Applications, Computations*, Academic Press, New York.
83. Tomlin J. (1970), Branch and bound methods for integer and non-convex programming, in: *Integer and nonlinear programming*, (Abadie ed.), North Holland.
84. Wilson R. (1970), Integer programming via modular representations, *Management Science*, vol. 16, n° 5, p. 289-294.
85. Wolsey L. A. (1974), A view of shortest route methods in integer programming, *Cahiers du Centre d'Études Rech. Operat.* 16, p. 317-335.
86. Young R. D. (1965), A Primal (All Integer) Integer Programming Algorithm, *J. Res. Nat. Bur. Standards* 69 B, p. 213-250.
87. Young R. D. (1968), A simplified Primal (All Integer) Integer Programming Algorithm, *Operations Research* 16, p. 750-782.
88. Zionts S. (1974), *Linear and integer programming*, Prentice Hall Inc., New Jersey, 1974.

ГЛАВА 8

РЕШЕНИЕ ЗАДАЧ БОЛЬШИХ РАЗМЕРНОСТЕЙ: ОБОБЩЕННОЕ ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ И ТЕХНИКА РАЗЛОЖЕНИЯ

В этой главе мы рассматриваем решение задач математического программирования, содержащих очень большое число переменных, и/или ограничений, и изучаем основные методы последовательности задач меньших размерностей: разложение Данцига — Вольфе (§ 2), разложение по действию правых частей (§ 3), разложение Бендерса (§ 4). Хотя для ясности изложения мы и предпочли ограничиться изложением приемов разложения только для задач *линейного программирования*, но все описанные здесь методы могут быть обобщены на случай разложения нелинейных задач программирования, обладающих подходящей структурой. С другой стороны, некоторые из этих технических приемов равным образом очень полезны при решении задач целочисленного программирования: это так в случае метода Данцига — Вольфе (который встречается в контексте лагранжевого ослабления) и, очевидным образом, в методе расчленения Бендерса, который, исторически, с самого начала развивался для решения смешанно-целочисленных задач программирования (см. [5, 14]).

Несколько практических применений методов разложения к задачам оптимизации больших сетей приведены в § 5.

Мы начнем с введения в § 1 техники, называемой *обобщенным линейным программированием*, которая составляет основное орудие для большинства методов, описанных ниже в этой главе.

§ 1. Обобщенное линейное программирование (порождение столбцов)

1.1. Постановка задачи. Рассмотрим задачу линейного программирования в стандартной форме

$$\begin{aligned} z = cx &\to \min, \\ Ax &= b, \qquad (P) \\ x &\geqslant 0. \end{aligned}$$

Введем обозначения:
n — число переменных,
m — число ограничений,
A — матрица $m \times n$ коэффициентов (a_{ij}),
$c = (c_1, \ldots, c_n)$ — вектор-строка цен,
$b = (b_1, \ldots, b_m)^T$ — вектор-столбец правых частей,
z — функция стоимости.

В этом параграфе мы изучим задачи линейного программирования типа (P), в которых число n переменных сильно превосходит число ограничений $(n \gg m)$ и столь велико, что матрица A не может быть полностью выражена (на практике это означает, что эту матрицу нельзя поместить полностью в память калькулятора). Предположим просто, что матрица A известна неявно, и ее столбцы соответствуют, вообще говоря, математическим объектам, которые полностью определены, но число которых очень велико. Так, в примере 1 ниже столбцы матрицы A соответствуют крайним точкам некоторого выпуклого многогранника; в примере 2 они соответствуют путям в графе.

Во всех случаях будем предполагать, что выполнено следующее условие (H).

Условие (H): для любого вектора-строки $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ существует алгоритм (называемый *порождающим алгоритмом*) для эффективного определения (т. е. иначе, чем с помощью перечисления столбцов A) такого столбца A_s, что

$$c_s - \pi A_s = \min_{j=1,\ldots,n} \{c_j - \pi A_j\}. \tag{1}$$

1.2. Пример 1 (разложение Данцига — Вольфе). В этом примере столбцы A соответствуют крайним точкам $(A_1, A_2, \ldots, A_n)$ выпуклого многогранника (Q), определенного системой

$$Fy = f,$$
$$y \geqslant 0,$$

где F — матрица $p \times m$, $y \in \mathbf{R}^m$, $f \in \mathbf{R}^p$.

Мы будем предполагать, что (Q) непуст (и, для простоты, ограничен). С другой стороны, цена c_j некоторого столбца A_j предполагается равной $c_j = \gamma A_j$ — скалярному произведению A_j на фиксированный вектор $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_m)$. В этом случае условие (H) действительно выполняется. Действительно, если π — некоторый вектор в $\mathbf{R}^m$, то определение столбца A_s в (1) равносильно решению задачи линейного программирования

$$\min (\gamma - \pi) y,$$
$$Fy = f,$$
$$y \geqslant 0.$$

Если матрица F имеет разумный размер (иначе говоря, можно полностью ввести ее в память калькулятора), то «обычный» алгоритм линейного программирования можно использовать для решения этой задачи, и полученное решение было бы крайней точкой A_j многогранника (Q) — такой, что скалярное произведение $(\gamma - \pi) A_j$ минимально.

В этом примере, к которому мы вернемся в методе разложения Данцига — Вольфе в п. 2.6, «порождающий алгоритм» состоит, следовательно, в «обычном» симплексном алгоритме (см. гл. 2, п. 2.1).

1.3. Пример 2 (поток с минимальной стоимостью в графе). Рассмотрим граф $G = [X, U]$, где X ($|X| = N$) есть множество вершин, а U ($|U| = M$) есть множество дуг. Разыскивается поток между двумя вершинами s и t с заранее заданным значением d, согласованный с емкостями b_u ($u \in U$) на дугах и имеющий минимальную стоимость (для $u \in U$ обозначим через γ_u стоимость единицы потока по дуге u).

Предположим, что граф не содержит замкнутых путей μ, стоимость которых

$$\gamma(\mu) = \sum_{u \in \mu} \gamma_u$$

отрицательна.

(Для введения в задачи потоков на графах см., например, [18], гл. 5.)

Граф содержит конечное число путей, содержащих вершину s с вершиной t, которые могут быть занумерованы индексом $j = 1, \ldots, p$. Сопоставим пути j ($1 \leqslant j \leqslant p$):

— вектор P_j с такими составляющими $P_j(1), \ldots, P_j(M)$, что

$$P_j(u) = \begin{cases} 1, & \text{если дуга } u \text{ принадлежит пути } j. \\ 0 & \text{в противном случае;} \end{cases}$$

— переменное $x_j \geqslant 0$ — количество потока, протекающее по пути j;

— стоимость c_j пути j, иначе говоря:

$$c_j = \sum_{u \in U} \gamma_u P_j(u) = \gamma P_j,$$

т. е. скалярное произведение вектора $\gamma = (\gamma_1, \ldots, \gamma_M)$ на вектор P_j.

Задача о потоке, согласованном со значением d и имеющем минимальную стоимость, может быть переписана тогда в виде

$$\sum_{j=1}^{p} c_j x_j \to \min,$$

$$\sum_{j=1}^{p} P_j x_j \leqslant b,$$

$$\sum_{j=1}^{p} x_j = d,$$

$$x \geqslant 0.$$

Если граф G имеет большой размер (много десятков вершин), то число p путей между двумя вершинами, вообще говоря, громадно и матрица ограничений не может быть полностью выражена явным образом. Однако легко видеть, что условие (Π) действительно выполнено, поскольку для данного π определение столбца P_s, минимизирующего $c_s = \pi P_s$ на множестве столбцов (путей), сводится к решению задачи о кратчайшем пути между вершинами s и t того же графа, дуги $u \in U$ которого снабжены длинами

$\gamma_u - \pi_u$. Для этой цели существует много эффективных методов (см., например, [18], гл. 2).

Таким образом, в этом примере порождающий алгоритм состоит в алгоритме кратчайшего пути.

Замечание. Задача о потоке на графе с минимальной стоимостью обычно решается не использованием предыдущей постановки задачи, а с помощью алгоритма Форда и Фалкерсона. Однако предложенная постановка лежит в основе решения задач о мультипотоках (см. [18], гл. 6, и § 5 настоящей главы).

Другие примеры приложений будут даны в пп. 1.6 и 1.7.

1.4. Решение задач с помощью обобщенного линейного программирования: алгоритм порождения столбцов. Вернемся к общей задаче (P) из п. 1.1.

Мы сейчас увидим, что если условие (H) п. 1.1 выполнено, то можно решить задачу так, что при этом приходится явным образом использовать относительно небольшое число столбцов. Это можно объяснить тем, что в точке оптимума число ненулевых переменных никогда не превосходит числа m ограничений задачи — хотя бы число переменных и оказалось гораздо большим.

Как и *при решении «обычных» задач линейного программирования*, всегда можно ограничиться (возможно, введением переменных скачка или искусственных переменных, см. гл. 2, п. 2.6) случаем, когда матрица A содержит единичную матрицу, которую можно взять в качестве исходного реализуемого базиса.

Тогда алгоритм выглядит следующим образом.

a) B^0 — исходный реализуемый базис. Шаг итерации $k = 0$.

b) $k \leftarrow k + 1$.

c) На текущей итерации пусть B — текущий базис (матрица размера $m \times m$), а c_B — m-вектор-строка стоимостей базисных переменных. Вычислить:

$\bar{b} = B^{-1}b$ (текущее базисное решение есть $x_B = \bar{b}$, $x_N = 0$).

$\pi = c_B B^{-1}$ (вектор симплексных множителей).

d) Определить, используя порождающий алгоритм, такой столбец A_s матрицы A, что

$$\bar{c}_s = c_s - \pi A_s = \min_{j=1,\ldots,n} \{c_j - \pi A_j\}.$$

e) Если $\bar{c}_s \geqslant 0$, то закончить: текущее решение $x_B = \bar{b}$, $x_N = 0$ оптимально.

Если $\bar{c}_s < 0$, то:

f) Вычислить $\bar{A}_s = B^{-1}A_s$.

Если $\bar{A}_s \leqslant 0$, то закончить: оптимум не ограничен $(-\infty)$. В противном случае вычислить

$$\hat{x}_s = \frac{\bar{b}_r}{\bar{A}_{rs}} = \min_{i|\bar{A}_{is}>0} \left\{\frac{\bar{b}_i}{\bar{A}_{is}}\right\}.$$

g) Пусть x_t — переменное, соответствующее r-й строке базиса, иначе говоря, такое, что $B^{-1}A_t = e_r$ (m-вектор, все компоненты которого равны нулю, кроме r-й компоненты, равной 1).

Определить новый базис B, заменяя столбец t столбцом s. Вычислить $\widehat{B}^{-1}$ и вернуться в b).

Мы можем убедиться, что этот алгоритм отличается от «обычного» алгоритма линейного программирования только методом выделения переменного и столбца, включаемого в базис в d). Так как (вследствие огромного количества переменных) невозможно действовать перечислением столбцов матрицы A (вычисляя для каждого столбца приведенную цену $\bar{c}_j = c_j - \pi A_j$), то мы используем порождающий алгоритм для определения включаемых переменных и столбцов.

1.5. Работа алгоритма и распределение памяти. Заметим, что загрузка памяти в этом алгоритме может быть относительно ограниченной и, во всяком случае, не сравнимой с тем, что необходимо для полной записи всей матрицы A.

Действительно, в каждый данный момент нужно сохранять просто обратную матрицу к текущей матрице (причем сама базисная матрица не нужна).

При каждой замене базиса матрица $\widehat{B}^{-1}$, обратная к новой базисной матрице, выводится непосредственно из обратной B^{-1} к старой базисной матрице умножением спереди на матрицу замены базиса (см. гл. 2, п. 2.5). Заметим, с другой стороны, что нет и необходимости явным образом вычислять матрицу, обратную к текущей базисной матрице: достаточно сохранять в памяти последовательность матриц перехода в компактной форме. (Это — форма произведения обратных матриц, уже описанная в гл. 2, п. 2.5.)

Различные матричные вычисления, необходимые при осуществлении алгоритма, а именно

$$\bar{b} = B^{-1}b,$$

$$\pi = c_B B^{-1}$$

и

$$\bar{A}_s = B^{-1}A_s,$$

осуществляются — при переходе к элементарным матрицам замены базиса — очень просто; кроме того, этот метод оказывается более эффективным и более устойчивым численно, поскольку он приводит к уменьшению числа вычислительных операций и распространения ошибок округлений. Однако базисная матрица должна периодически подвергаться обращению заново, если число итераций очень велико по сравнению с m (числом ограничений), а также если устанавливается наличие ухудшения точности вычислений (эту точность можно легко контролировать, например, вычисляя приведенную стоимость базисного переменного — тогда должен получаться нуль — или вычисляя произведение $B^{-1}A_i$, где A_i есть базисный столбец — тогда должен получаться столбец единичной матрицы — см., например, [43]).

1.6. Приложение к оптимизации выпуклых или вогнутых функций методом секущих плоскостей. Особенно важное приложение обобщенного линейного программирования состоит в оптимизации

выпуклых (или вогнутых) функций. Этот метод применим как к *дифференцируемым функциям* (*иначе говоря, имеющим градиент* в каждой точке), так и к не всюду дифференцируемым функциям (ср. гл. 4, § 3).

Пусть, например, нужно минимизировать выпуклую функцию

$$f(x)=f(x_1, \ldots, x_n) \text{ на } \mathbf{R}^n.$$

(Случай максимизации вогнутой функции исследуется аналогично.)

Предположим, что в любой точке $x \in \mathbf{R}^n$ мы можем вычислить субградиент γ функции f в точке x, т. е. некоторый элемент субдифференциала $\partial f(x)$ (если f дифференцируема, то $\partial f(x)$ содержит единственный элемент, а именно градиент f в x).

Известно (см. гл. 1), что если γ — субградиент f в точке $\bar{x}$, то гиперплоскость в $\mathbf{R}^{n+1}$, заданная уравнением

$$z=f(\bar{x})+\gamma^T(x-\bar{x})$$

есть опорная гиперплоскость к надграфику функции f:

$$\text{épi}(f)=\{(\mu, x) \mid f(x) \leqslant \mu, x \in \mathbf{R}^n, \mu<+\infty\}.$$

Иначе говоря, épi(f) может рассматриваться как верхняя оболочка (априори бесконечного) семейства гиперплоскостей, заданных уравнениями

$$z=f(\bar{x})+\gamma^T(x-\bar{x}),$$

где $\bar{x}$ пробегает $\mathbf{R}^n$, а γ пробегает $\partial f(\bar{x})$.

Тогда идея метода состоит в приближении функции f функцией $\widehat{f}$, надграфик которой есть линейное приближение кусками épi(f), которое строится лишь по конечному числу гиперплоскостей

$$\begin{aligned}
z &= f(x^1)+(\gamma^1)^T(x-x^1),\\
z &= f(x^2)+(\gamma^2)^T(x-x^2),\\
&\cdots\cdots\cdots\cdots\cdots\\
z &= f(x^p)+(\gamma^p)^T(x-x^p),
\end{aligned}$$

где

$$\begin{aligned}
&\gamma^1 \text{ — субградиент } f \text{ в } x^1,\\
&\gamma^2 \text{ — субградиент } f \text{ в } x^2,\\
&\cdots\cdots\cdots\cdots\cdots\\
&\gamma^p \text{ — субградиент } f \text{ в } x^p.
\end{aligned}$$

Заметим, что по построению $\widehat{f}(x) \leqslant f(x)$ $\forall x \in \mathbf{R}^n$.

Затем приближение итеративно улучшается следующим образом.

Ищется минимум функции $\widehat{f}$, что равносильно решению задачи линейного программирования

$$\begin{aligned}
&\min_{(z,x)} z,\\
&z-(\gamma^1)^T(x-x^1) \geqslant f(x^1), \qquad (PL)\\
&\cdots\cdots\cdots\cdots\cdots\\
&z-(\gamma^p)^T(x-x^p) \geqslant f(x^p).
\end{aligned}$$

Пусть (z^{p+1}, x^{p+1}) — оптимальное решение этой задачи линейного программирования.

Вычисляем $f(x^{p+1})$: необходимо имеем

$$z^{p+1} = \hat{f}(x^{p+1}) \leqslant f(x^{p+1}).$$

Тогда могут встретиться два случая.

Случай 1. Если $z^{p+1} < f(x^{p+1})$, то мы определяем субградиент γ^{p+1} функции f в точке x^{p+1} и добавляем к задаче (PL) ограничение

$$z - (\gamma^{p+1})^T (x - x^{p+1}) \geqslant f(x^{p+1})$$

(которое не удовлетворяется текущим решением (z^{p+1}, x^{p+1})), и теперь подлежит решению новая «расширенная» таким образом задача (PL).

Случай 2. Если $z^{p+1} = f(x^{p+1})$, то алгоритм завершает работу.

Действительно, поскольку x^{p+1} есть оптимум для $\hat{f}$, то

$$f(x^{p+1}) = \hat{f}(x^{p+1}) \leqslant \hat{f}(x) \leqslant f(x) \qquad \forall x \in \mathbf{R}^n,$$

следовательно, x^{p+1} есть также оптимум для f.

Заметим, что этот метод может рассматриваться как специальный случай метода секущих плоскостей Келли [23] (см. гл. 5, п. 3.6) для задачи

$$\min y,$$
$$f(x) - y \leqslant 0,$$
$$x \in \mathbf{R}^n, \qquad y \in \mathbf{R}.$$

Нужно также заметить, что на практике решается не задача (PL) (с ее порождением препятствий), а ее двойственная задача, которая решается порождением столбцов в соответствии с принципом обобщенного линейного программирования.

В многочисленных приложениях надграфик функции f есть оболочка конечного семейства гиперплоскостей (среди прочих задач этому типу принадлежит двойственная функция задачи дискретного программирования, см. гл. 4 п. 3.1). Тогда наш метод очевидным образом сходится за конечное число итераций.

Даже если надграфик функции f есть оболочка бесконечного семейства гиперплоскостей, приводимая ниже теорема 1 показывает, что при малоограничительных предположениях последовательность получаемых точек x^p имеет по крайней мере одну точку накопления, которая является оптимумом для функции f. При этих условиях критерий остановки (случай 2), вообще говоря, не может быть удовлетворен в результате конечного числа итераций. Однако, как всегда, z^p является минорантой, а $f(x^p)$ — мажорантой оптимума, и можно прервать вычисления, как только оказывается, что $|f(x^p) - z^p| \leqslant \varepsilon$, где допустимая величина ε зафиксирована с самого начала.

Теорема 1. *Пусть* $\{x^p\}_{(p\in\mathbf{N})}$ — *последовательность точек, порождаемая методом секущих плоскостей. Если функция выпукла, непрерывно дифференцируема на* $\mathbf{R}^n$ *и удовлетворяет условию*

$$f(x) \to +\infty, \quad \|x\| \to +\infty,$$

то из последовательности $\{x^p\}$ *можно выделить подпоследовательность* $\{x^l\}_{l\in L}$ $(L \subset \mathbf{N})$, *сходящуюся к* (*глобальному*) *оптимуму* x^* *функции* f.

Доказательство. Оно вытекает из сходимости метода секущих плоскостей Келли, установленной в гл. 5 (теорема 10 п. 3.6.2).

Заметим, что эта теорема может быть обобщена на случай, когда значения функции f и/или ее субградиентов известны не точно, а лишь ограничены некоторой точностью ε. Тогда можно доказать (см. [33]) сходимость к приближенному с точностью до η значению оптимального значения (где η зависит от ε).

Отметим, наконец, что метод секущих плоскостей, примененный к дуальной функции задачи нелинейного выпуклого программирования, есть не что иное, как алгоритм Данцига, тщательно изученный в гл. 6, п. 3.2.

1.7. Другие приложения обобщенного линейного программирования. Приложения обобщенного линейного программирования чрезвычайно многочисленны.

a) Прежде всего можно обратить внимание на метод секущих плоскостей Келли [23] для задач нелинейного выпуклого программирования с ограничениями (см. гл. 5, п. 3.6). Алгоритм, описанный в п. 1.6 выше, может рассматриваться как частное приложение этого метода.

b) Другой важный класс приложений, как мы увидим в следующих разделах, состоит в разложении больших задач линейного программирования. Так, в алгоритме разложения Данцига — Вольфе (см. § 2 ниже), основная задача решается порождением столбцов. В алгоритме разложения с использованием правых частей (§ 3 ниже) и в алгоритме разложения Бендерса (§ 4 ниже) равным образом используется обобщенное линейное программирование.

c) Наконец, обобщенное линейное программирование связано с многочисленными прямыми приложениями: к задачам о потоках в графах [17], к задачам о промышленном раскрое [15, 16], к задачам о мультипотоках [32, 33].

§ 2. Лагранжево ослабление и разложение по ценам (Данциг — Вольфе)

2.1. Общие соображения о структуре больших задач линейного программирования. В приложениях чрезвычайно редко встречаются задачи линейного программирования очень большой размерности (тысячи или десятки тысяч переменных, много тысяч ограничений), которые бы не имели специальной структуры.

Прежде всего, матрицы ограничений этих задач линейного программирования обычно «очень полые», т. е. содержат большую долю нулевых членов. Это соответствует, кроме всего прочего, необходимости практического порядка: если этот случай не имеет места, то данные задачи просто физически невозможно «ухватить» («полная» матрица $5000 \times 20\,000$ соответствует ста миллионам членов)!

Кроме того, не только доля нулевых элементов должна быть достаточно высокой, но и сами эти нулевые члены, вообще говоря, не должны быть распределены как попало. Рисунки 1—3 показывают структуру наиболее типичных матриц ограничений, встречающихся в большинстве приложений (ненулевые элементы содержатся только в заштрихованных частях):

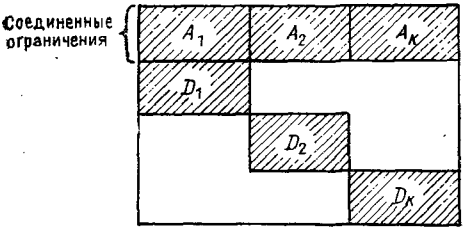


Рис. 1. Блочно-диагональная структура с соединенными ограничениями

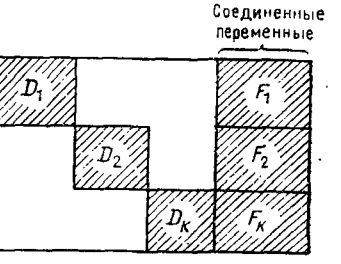


Рис. 2. Блочно-диагональная структура с соединенными переменными

— рис. 1 соответствует случаю K блоков, связанных с K подзадачами, которые были бы независимы, не будь некоторого количества дополнительных ограничений (так называемых соединенных ограничений) (задачи типа 1);

— рис. 2 соответствует случаю K блоков, которые были бы независимы, не будь некоторого количества дополнительных переменных (так называемых соединенных переменных) (задачи типа 2);

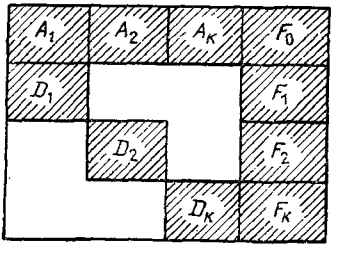


Рис. 3. Блочно-диагональная структура с соединенными ограничениями и соединенными переменными

— рис. 3 соответствует наиболее сложному случаю, в котором есть одновременно и переменные, и ограничения, связывающие подзадачи между собой (задачи типа 3).

2.2. Постановка задачи. В этом параграфе мы будем заниматься только разложимыми задачами линейного программирования типа 1, иначе говоря, только с соединенными ограничениями.

Таким образом, эти задачи имеют вид

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ Dx &= d, \\ x &\geqslant 0, \end{aligned} \qquad (P_1)$$

где матрица D блочно-диагональна:

$$D = \begin{bmatrix} D_1 & & 0 \\ & D_2 & \\ & \ddots & \\ 0 & & D_K \end{bmatrix}.$$

Различным блокам $D_1, \ldots, D_k$ соответствуют разбиения на блоки:

— вектора c на подвекторы $(c_1, \ldots, c_k)$;
— вектора x на подвекторы $(x_1, \ldots, x_k)$;
— матрицы A на подматрицы $(A_1, \ldots, A_k)$;
— вектора d на подвекторы $(d_1, \ldots, d_k)$.

Тогда задачу можно переписать в матричных обозначениях:

$$\begin{aligned} \min z = cx &= \sum_{k=1}^{K} c_k x_k, \\ \sum_{k=1}^{K} A_k x_k &= b, \\ D_k x_k &= d_k, \qquad k = 1, \ldots, K, \\ x_k &\geqslant 0, \qquad k = 1, \ldots, K, \end{aligned}$$

или еще, вводя выпуклые политопы

$$X_k = \{x_k | D_k x_k = d_k; \quad x_k \geqslant 0\},$$

в форме

$$\begin{aligned} \min z = cx &= \sum_{k=1}^{K} c_k x_k, \\ \sum_{k=1}^{K} A_k x_k &= b, \\ x_k &\in X_k, \ k = 1, \ldots, K. \end{aligned}$$

Если мы обозначим через X выпуклый политоп

$$X = \{x | x = (x_1, \ldots, x_K); \ x_1 \in X_1, \ldots, x_K \in X_k\},$$

то можно, наконец, записать задачу в виде

$$\begin{aligned} \min z &= cx, \\ Ax &= b, \\ x &\in X. \end{aligned}$$

Для упрощения изложения предположим, что политопы X_k ($k = 1, \ldots, K$) ограничены и непусты. Очевидно, что с помощью ряда предосторожностей можно без труда применять наши методы к случаю неограниченных политопов.

Мы опишем ниже алгоритм разложения Данцига — Вольфе. Однако вместо того, чтобы выбрать классическое изложение [9], мы сочли более предпочтительным выявить глубокие связи, существующие между этим алгоритмом и техникой лагранжевых ослаблений (см. гл. 6, § 3), которая в настоящее время используется все шире (в сочетании с методами субградиента) для решения задач больших размерностей (см. [26, 27, 29]).

2.3. Лагранжево ослабление соединенных ограничений. Вычисление дуальной функции с помощью разложения. Свяжем с каждым соединенным ограничением i задачи (P_1) переменное λ_i (произвольного знака), называемое *множителем Лагранжа* или *двойственным переменным*. Обозначим через λ вектор (строку) множителей Лагранжа.

Для произвольного λ определим функцию Лагранжа $L(x, \lambda)$ формулой

$$L(x, \lambda) = cx - \lambda(Ax - b) = \lambda b + \sum_{k=1}^{K} (c_k - \lambda A_k) x_k,$$

а затем дуальную функцию $w(\lambda)$ формулой

$$w(\lambda) = \min_{x \in X} \{L(x, \lambda)\} = \lambda b + \min_{x \in X} \{(c - \lambda A) x\} =$$

$$= \lambda b + \sum_{k=1}^{K} \min_{x_k \in X_k} \{(c_k - \lambda A_k) x_k\}.$$

Мы сразу видим, что вычисление дуальной функции разлагается в решение K независимых подзадач вида

$$\begin{aligned} \tilde{c}_k x_k &\to \min, \\ D_k x_k &= d_k, \\ x_k &\geqslant 0, \end{aligned} \qquad (S_k)$$

где положено $\tilde{c}_k = c_k - \lambda A_k$ («модифицированная цена»).

Теперь каждая из подзадач является задачей линейного программирования с много меньшим числом переменных, чем исходная задача; следовательно, она может быть эффективно разрешена с помощью симплексного алгоритма.

(Важное замечание: в алгоритме разложения Данцига — Вольфе (см. п. 2.6) подзадачи S_k называются *вспомогательными задачами* или *служебными задачами*.)

Вычисление дуальной функции сводится, таким образом, к решению K независимых задач линейного программирования с меньшим числом переменных, и каждая из этих задач соответствует некоторому блоку в глобальной задаче линейного программирования.

Как мы увидим, именно это свойство дуальной функции и позволит сейчас решить общую, глобальную задачу с помощью разложения.

Определим дуальную задачу (D_1) задачи (P_1) как задачу оптимизации (без ограничений);

$$\max_{\lambda \text{ любого знака.}} w(\lambda), \qquad (D_1)$$

Тогда теория лагранжевой двойственности (см. гл. 6 § 2) позволяет установить следующие свойства дуальной функции.

Свойство 1. *Функция w вогнута и не всюду дифференцируема по λ.*

Свойство 2. *Если (P_1) имеет оптимальное решение x^* с конечным значением, то существует седловая точка, и для любого λ и любого решения x задачи (P_1) имеем*

$$w(\lambda) \leqslant w(\lambda^*) = cx^* \leqslant cx$$

(где λ^* — оптимальное решение двойственной задачи).

Свойство 3. *Пусть $\tilde{x}_k$ $(k = 1, \ldots, K)$ — оптимальное решение подзадачи (S_k) (где $\tilde{c}_k = c_k - \lambda A_k$), тогда вектор*

$$\gamma = b - \sum_{k=1}^{K} A_k \tilde{x}_k = b - A\tilde{x}$$

есть вектор субградиента функции w в точке λ.

2.4. Решение дуальной задачи с помощью алгоритма субградиента. Только что приведенное свойство 3 показывает, что в качестве промежуточного результата вычисления значения $w(\lambda)$ в точке λ мы сразу получаем ее субградиент в λ, т. е. элемент субдифференциала $\partial w(\lambda)$.

Следовательно, для решения двойственной задачи остается найти максимум вогнутой (не всюду дифференцируемой) функции, для которой в каждой точке известен субградиент. Тогда можно применить для решения этой дуальной задачи алгоритм субградиента (см. гл. 4, § 3).

Этот подход представляет интерес для быстрого получения хорошей аппроксимации оптимальных двойственных переменных λ^* и хороших аппроксимаций снизу стоимости оптимального решения задачи (P_1). Эти аппроксимации снизу могут быть использованы затем для контроля качества решений исходной задачи, полученных другими методами, и для выработки критериев остановки.

Этот метод был с успехом применен в многочисленных задачах, где число переменных велико. Можно указать, в частности,

— задачи о мультипотоках в графах ([20, 18], гл. 6);

— задачи об оптимальном синтезе сетей, пробегаемых не одновременными потоками (см. [32, 33]);

— задачи об обобщенном действии в больших размерностях [28]).

Как мы сейчас увидим, другой способ решения двойственной задачи состоит в применении обобщенного линейного программи-

рования. Это приведет нас в п. 2.6 к алгоритму разложения Данцига — Вольфе.

2.5. Формулировка дуальной задачи как задачи линейного программирования. Напомним определение дуальной функции:

$$w(\lambda) = \lambda b + \sum_{k=1}^{K} \min_{x_k \in X_k} \{(c_k - \lambda A_k) x_k\}. \qquad (2)$$

Так как оптимум подзадачи, подобной задаче (S_k), всегда достигается в крайней точке многогранника X_k, то значение $w(\lambda)$ не изменится, если в (2) брать минимум по (конечному) множеству Y_k крайних точек X_k. Мы получаем равносильное выражение

$$w(\lambda) = \lambda b + \sum_{k=1}^{K} \min_{y_k \in Y_k} \{(c_k - \lambda A_k) y_k\}.$$

Обозначим через Y (также конечное) множество крайних точек множества X, т. е.

$$Y = \{y \mid y = (y_1, \ldots, y_K);\ y_1 \in Y_1,\ y_K \in Y_K\}.$$

Следовательно, можно написать

$$w(\lambda) = \lambda b + \min_{y \in Y} \{(c - \lambda A) y\}.$$

Предположим теперь, что мы мысленно пронумеровали крайние точки множества Y:

$$Y = \{y^1, \ldots, y^q\}.$$

Мы видим, что значение $w(\lambda)$ удовлетворяет системе неравенств

$$\begin{aligned} w(\lambda) &\leqslant \lambda b + (c - \lambda A) y^1, \\ w(\lambda) &\leqslant \lambda b + (c - \lambda A) y^2, \\ &\cdots\cdots\cdots \\ w(\lambda) &\leqslant \lambda b + (c - \lambda A) y^q. \end{aligned}$$

Кроме того, для каждого λ существует по крайней мере одно неравенство, обращающееся в равенство.

Отсюда следует, что максимум $w(\lambda^*)$ функции $w(\lambda)$ есть не что иное, как оптимальное решение задачи линейного программирования

$$\begin{gathered} \max_{(v,\lambda)} v, \\ v \leqslant \lambda b + (c - \lambda A) y^1, \\ \cdots\cdots\cdots \\ v \leqslant \lambda b + (c - \lambda A) y^q, \end{gathered}$$

которую, вводя обозначения

$$z^i = cy^i, \quad b - Ay^i = \gamma^i$$

(для $i = 1, \ldots, q$), можно переписать в виде

$$\begin{aligned} &\max_{(v,\lambda)} v, \\ &v - \lambda\gamma^1 \leqslant z^1, \\ &v - \lambda\gamma^2 \leqslant z^2, \\ &\cdots\cdots\cdots \\ &v - \lambda\gamma^q \leqslant z^q. \end{aligned} \qquad (I)$$

Очевидно, что для задач больших размерностей число q элементов множества Y (множества крайних точек X) огромно, и нет возможности явным способом найти все ограничения в (I).

Мы оказываемся здесь в типичной ситуации, в которой для решения задачи (I) (на самом деле — ее двойственной задачи) может быть применено обобщенное линейное программирование.

2.6. Алгоритм разложения Данцига — Вольфе [9]. Этот алгоритм состоит в применении обобщенного линейного программирования (см. § 1) к задаче, двойственной к задаче (I).

Свяжем с каждым ограничением i $(i = 1, \ldots, q)$ в (I) двойственное переменное $u_i \geqslant 0$.

Двойственная к (I) задача имеет вид

$$\begin{aligned} &\sum_{i=1}^{q} z^i u_i \to \min, \\ &-\sum_{i=1}^{q} \gamma^i u_i = 0, \quad (3) \\ &\sum_{i=1}^{q} u_i = 1, \quad (4) \\ &u_i \geqslant 0, \qquad i = 1, \ldots, q. \end{aligned} \qquad (II)$$

Эта задача содержит умеренное число ограничений ($m + 1$, если m — число соединенных ограничений задачи (P_1)), но зато огромное количество переменных. Поэтому она не может быть решена непосредственно, как «обычная» задача линейного программирования.

Поэтому мы удовлетворимся решением задачи линейного программирования, ограниченной малым числом переменных (или столбцов), выделенных из задачи (II), которая будет называться *главной задачей*.

Чтобы иметь возможность применить к задаче (II) обобщенное линейное программирование, достаточно удостовериться, что гипотеза (H) из п. 1.1 действительно выполнена.

Иначе говоря, пусть λ — некоторый вектор дуальных переменных (симплексных множителей), связанных с ограничениями (3) и v — дуальное переменное (симплексный множитель), связанный с ограничением (4) в (II); нужно знать, существует ли эффективный метод определения (без помощи перечисления переменных)

такого переменного u_s, чтобы

$$z^s + \lambda\gamma^s - v = \min_{i=1,\ldots,q} \{z^i + \lambda\gamma^i - v\}.$$

Для этого достаточно найти такой индекс s, что

$$z^s + \lambda\gamma^s = \min_{i=1,\ldots,q} \{z^i + \lambda\gamma^i\}.$$

Возвращаясь к определениям величин z^i и γ^i:

$$z^i = cy^i, \quad \gamma^i = b - Ay^i,$$

видим, что это равносильно тому, чтобы найти

$$\lambda b + (c - \lambda A)\, y^s = \min_{i=1,\ldots,q} \{\lambda b + (c - \lambda A)\, y^i\} = \min_{x \in X} \{\lambda b + (c - \lambda A)\, x\},$$

что в точности эквивалентно вычислению дуальной функции w в точке λ.

В этом случае порождающий алгоритм есть не что иное, как алгоритм вычисления $w(\lambda)$, и он состоит в решении — для текущего значения λ (оптимальных дуальных переменных текущей главной задачи) — K вспомогательных задач, определенных в п. 2.3:

$$\begin{gathered} \tilde{c}_k x_k \to \min, \\ D_k x_k = d_k, \\ x_k \geqslant 0 \end{gathered} \qquad (S_k)$$

(с $\tilde{c}_k = c_k - \lambda A_k$) (см. пример 1 п. 1.2).

Эти задачи решаются симплексным алгоритмом независимо, одна за другой.

Полученные решения $\tilde{x}_1, \ldots, \tilde{x}_k$ суть крайние точки множеств $X_1, \ldots, X_k$ соответственно, и $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_k)$ есть крайняя точка X, следовательно, это — некоторый элемент y^s множества Y.

Таким образом, порождающий алгоритм приводит на каждой итерации к дополнительному столбцу в главной задаче, определяемому формулами

$$\gamma^s = b - A\tilde{x} = b - Ay^s,$$

$$z^s = c\tilde{x} = cy^s.$$

Получается «расширенная» главная задача, которая решается заново (оптимальное решение предыдущей главной задачи используется в качестве исходного решения), что приводит к новым дуальным переменным λ. Они включаются в порождающий алгоритм, который дает новый столбец, и т. д.

Итерации продолжаются, пока новое переменное порождает строго отрицательную приведенную цену, т. е.

$$z^s + \lambda\gamma^s - v < 0.$$

Если же $z^s + \lambda\gamma^s - v \geqslant 0$, то нужно остановиться, так как в этом случае все переменные в (II) имеют неотрицательную приведенную стоимость: это означает, что текущее решение главной задачи есть оптимальное решение задачи (II).

Тогда v есть значение этого оптимального решения, а оптимальные переменные u_i^* текущей главной задачи позволяют определить оптимальное решение задачи (P_1) по формуле

$$x^* = \sum_{i|u_i^*>0} y^i u_i^*$$

(по определению $Dx^* = d$, и можно проверить, что действительно $Ax^* = b$).

Замечание. На практике сходимость алгоритма Данцига — Вольфе часто оказывается довольно-таки медленной, особенно в окрестности оптимума. С другой стороны, ошибки округления (неизбежные с того момента, когда мы начали исследовать большие задачи), вообще говоря, не позволяют дать для описанного выше удовлетворительный критерий остановки.

Тогда может оказаться необходимым подготовить другой критерий остановки. Но это легко сделать, если заметить, что на каждой итерации

— определение входного переменного u_s с помощью порождающего алгоритма приводит одновременно к значению дуальной функции $w(\lambda) = \lambda b + (c - \lambda A) y^s$, которое является минорантой оптимума cx^*;

— оптимальное значение текущей главной задачи составляет мажоранту оптимума cx^* (так как главная задача получается из (II) отменой некоторого числа переменных).

Следовательно, на каждом шаге итерации мы располагаем двусторонней оценкой искомого оптимального значения cx^*, и можно прервать вычисления, как только мы получим величину зазора, меньшую фиксированной с самого начала точности ε.

2.7. **Экономическая интерпретация.** В заключение приведем интересную экономическую интерпретацию принципа разложения Данцига — Вольфе.

Можно представить себе экономическую систему (большую компанию), содержащую некоторое количество K подсистем (филиалов).

Если каждая подсистема (каждый филиал) k действует независимо от остальных, то предположим, что нужно так согласовать их уровни активов (переменные x_k), чтобы минимизировать величину $c_k x_k$ (экономическую функцию подсистемы k) при ограничениях $x_k \in X_k$ (что выражает различные возможности использования их собственных средств).

Однако подсистемы (филиалы) не независимы, но связаны между собой ограничениями использования ресурсов, разделяемых на глобальном уровне (финансовый пакет, рабочая сила и т. д.). Правые части $b = (b_1, \ldots, b_m)$ суть полные количества ресурсов, подлежащие разделению между различными подсистемами (филиалами).

Метод Данцига — Вольфе состоит в приписывании каждой подсистеме (филиалу) требования уплатить единичную цену — λ_i (множитель Лагранжа) за каждый распределяемый ресурс b_i.

Тогда каждая подсистема k реагирует тем, что включает цену распределяемых ресурсов в свою собственную экономическую функцию. Иначе говоря, отыскиваются новые уровни активов x_k, минимизирующие величину

$$\tilde{c}_k x_k = c_k x_k - \lambda A_k x_k$$

при ограничениях $x_k \in X_k$.

Цены λ подбираются тогда так, чтобы ограничения ресурсов сохранялись на глобальном уровне. Это выражается в названии *разложения по ценам*, которое относится также к методу Данцига — Вольфе, в противоположность технике *разложения по ресурсам*, которая будет содержанием следующего раздела.

§ 3. Разложение по действию правых частей (разложение по ресурсам)

Этот параграф посвящен другому важному классу методов решения разложимых больших задач типа 1 (т. е. только с соединенными ограничениями): *методам разложения по действию правых частей* (по-английски: right-hand-side allocation), называемым также *методами разложения по ресурсам* [11, 42].

3.1. Принцип метода. Как и в § 2, рассмотрим задачу

$$\min z = cx = \sum_{k=1}^{K} c_k x_k,$$

$$\sum_{k=1}^{K} A_k x_k = b, \qquad (P_1)$$

$$x_k \in X_k.$$

Предположим, что каждое множество X_k есть выпуклый политоп:

$$X_k = \{x_k \mid D_k x_k = d_k;\ x_k \geqslant 0\},$$

но X_k может быть определен также и другими способами (см. обобщения в п. 3.5).

Если сослаться на экономическую интерпретацию из п. 2.7, то можно сказать, что метод разложения Данцига — Вольфе состоит в неявном получении оптимального распределения редких ресурсов между подсистемами посредством введения цен, приписываемых каждому типу ресурсов.

Метод, который мы сейчас собираемся изложить, состоит в явном присвоении некоторой доли глобальных ресурсов каждой подсистеме. При заданном распределении ресурсов задача распадается в решение K независимых подзадач.

Тогда мы итеративно ищем распределение ресурсов, минимизирующее сумму стоимостей полученных частных решений.

На каждом шаге решения K подзадач позволяют либо заключить, что текущее распределение является оптимальным (и в этом случае получить оптимальные уровни активов для каждой подсис-

темы), либо определить новое распределение, приводящее к уменьшению стоимости.

3.2. Эквивалентная формулировка задачи. Определим для каждой из подсистем $k = 1, \ldots, K$ вектор y_k как набор доль правых частей b (ресурсов), представляемых этой подсистеме. Каждый из векторов y_k есть вектор той же размерности, что и b, причем должно выполняться условие

$$\sum_{k=1}^{K} y_k = b.$$

Для данного распределения $y = (y_1, \ldots, y_K)$ каждая из подсистем k должна выбрать свои уровни активов как оптимальное решение подзадачи

$$\begin{aligned} c_k x_k &\to \min, \\ A_k x_k &= y_k, \qquad P_k(y_k) \\ x_k &\in X_k. \end{aligned}$$

Каждая из подзадач $P_k(y_k)$ есть задача линейного программирования, которая, независимо от остальных подзадач, может быть решена с помощью симплексного алгоритма.

Обозначим через Y_k $(k = 1, \ldots, K)$ множество таких векторов y_k, что задача $P_k(y_k)$ имеет решение, и обозначим для $y_k \in Y_k$ через $v_k(y_k)$ значение оптимального решения $P_k(y_k)$.

Тогда задача (P_1) равносильна следующей задаче (главной задаче):

$$\begin{aligned} &\min v(y) = \sum_{k=1}^{K} v_k(y_k), \\ &\sum_{k=1}^{K} y_k = b, \qquad (PM) \\ &y_k \in Y_k, \qquad k = 1, \ldots, K. \end{aligned}$$

Чтобы устранить ограничения $y_k \in Y_k$, которые в задаче (PM) учитывать трудно, заметим, что достаточно ввести штрафы с достаточно высокой стоимостью за недопустимые распределения ресурсов, т. е. приводящие к подзадачам $P_k(y_k)$, не имеющим решения.

Этого легко можно добиться, добавляя в каждой из подзадач $P_k(y_k)$ векторы искусственных переменных $\xi_k^+ \geqslant 0$ и $\xi_k^- \geqslant 0$ к правой части y_k. Этим искусственным переменным приписывается очень высокая стоимость $h > 0$. Таким образом, получается задача

$$\begin{aligned} &\min \left(c_k x_k + H\left(\xi_k^+ + \xi_k^-\right)\right), \\ &A_k x_k + \xi_k^+ - \xi_k^- = y_k, \qquad P_k'(y_k) \\ &x_k \in X_k, \end{aligned}$$

где H — вектор-строка той же размерности, что и y_k (следовательно, той же размерности, что и b), в которой все компоненты равны

h. Заметим, что, благодаря присутствию искусственных переменных, задача $P_k'(y_k)$ всегда имеет решение.

При достаточно большом $h > 0$ необходимо получаем, что если $P_k(y_k)$ имеет решение, то в точке оптимума $\xi_k^+ = 0$ и $\xi_k^- = 0$. Следовательно, если мы обозначим через $v_k'(y_k)$ оптимальное значение $P_k'(y_k)$, то мы получим, что

$$v_k'(y_k) = v_k(y_k) \quad \text{для} \quad y_k \in Y_k,$$
$$v_k'(y_k) > 0 \quad \text{и очень велик для} \quad y_k \notin Y_k.$$

Отсюда следует, что для достаточно больших $h > 0$ задача (P_1) эквивалентна задаче

$$\min v'(y) = \sum_{k=1}^{K} v_k'(y_k), \qquad (PM)'$$
$$\sum_{k=1}^{K} y_k = b.$$

Для каждого распределения $y = (y_k)$, удовлетворяющего ограничениям $\sum y_k = b$, значение функции $v'(y)$ определяется с помощью решений K независимых подзадач $P_k'(y_k)$.

При всем том есть два вопроса:

— как узнать, является ли распределение y оптимальным?

— если не является, то как определить лучшее распределение?

3.3. Определение субградиента функции $v'(y)$. Для данного распределения $\bar{y}$ решение семейства подзадач

$$\min\left(c_k x_k + H\left(\xi_k^+ + \xi_k^-\right)\right),$$
$$A_k x_k + \xi_k^+ - \xi_k^- = \bar{y}_k, \qquad P_k'(\bar{y}_k) \quad (5)$$
$$x_k \in X_k,$$

с помощью симплексного алгоритма приводит к оптимальным дуальным переменным $\bar{u}_k$, связанным с ограничениями (5).

Заметим тогда, что функция $v_k'(y_k)$ есть не что иное, как функция возмущений, связанная с задачей $P_k'(y_k)$ (гл. 5, п. 2.4). Так как речь идет о задаче линейного программирования, а следовательно, о задаче выпуклого математического программирования, то функция $v'(y)$ — а потому и каждая из функций $v_k'(y_k)$ — есть выпуклая функция. С другой стороны, можно использовать результаты гл. 5, пп. 2.5 и 2.8, связывающие седловые точки функции Лагранжа с субградиентами функции возмущений. В частности, известно, что $\bar{u}_k$ тогда и только тогда является вектором оптимальных дуальных переменных $P_k'(y_k)$, когда $(-\bar{u}_k)$ есть субградиент функции возмущений $v_k'(y_k)$ в точке $\bar{y}_k$.

Таким образом, как промежуточный результат в ходе вычисления $v'(\bar{y})$ мы получаем и субградиент функции $v'(y)$ в точке $\bar{y}$:

это — вектор $(-\bar{u}) = -(\bar{u}_1, \ldots, \bar{u}_K)$, где $\forall k$ вектор $\bar{u}_k$ есть вектор оптимальных дуальных переменных задачи $P'_k(\bar{y}_k)$.

3.4. Решение главной задачи. Решение главной задачи

$$v'(y) = \sum_{k=1}^{K} v'_k(y_k) \to \min,$$
$$\sum_{k=1}^{K} y_k = b \qquad (PM)'$$

сводится тогда к минимизации выпуклой (не всюду дифференцируемой) функции, для которой в каждой точке можно определить некоторый ее субградиент. Как мы сейчас увидим, присутствие дополнительных ограничений $\sum_{k=1}^{K} y_k = b$ не осложняет задачу сколько-нибудь серьезно.

По существу могут быть применены два типа методов.

1) Обобщенное линейное программирование.

Действительно, задача $(PM)'$ аналогична общей задаче, исследованной в п. 1.6 настоящей главы. Следовательно, она может быть решена с помощью обобщенного линейного программирования, что приводит к итеративному решению линейной задачи, ограничения которой, постепенно добавляемые в ходе решения, соответствуют субградиентам функции $v'(y)$.

Учет ограничений $\sum y_k = b$ очень прост: достаточно добавлять их к задаче линейного программирования, которая подлежит решению на каждом шаге итерации.

Неудобство метода состоит в том факте, что если задача (P_1) содержит некоторое число m существенных соединенных ограничений, то получаемая из нее задача обобщенного линейного программирования может иметь очень большую размерность (Km переменных). В этом случае приводимый ниже второй метод часто обнаруживает бо́льшую эффективность при быстром получении хороших приближенных решений.

2) Алгоритмы субградиента.

Методы субградиента, описанные в гл. 4, § 3, могут быть очевидным образом применены для оптимизации функции $v'(y)$.

Если на шаге t мы находимся в точке y^t, при отсутствии других ограничений, то точка y^{t+1} определяется формулой

$$y^{t+1} = y^t + 0_t \gamma^t,$$

где γ^t — субградиент функции $v'(y)$ в точке y^t, а 0_t — шаг перемещения, выбираемый, вообще говоря, априори (два различных способа выбирать шаг 0_t обсуждены в гл. 4, п. 3.4).

В этой итеративной схеме очень легко учесть дополнительные ограничения $\sum_{k=1}^{K} y_k = b$.

Так как $\bar{y} = y^t + 0_t \gamma^t$, вообще говоря, не удовлетворяет дополнительным ограничениям $\sum y_k = b$, то мы получим y^{t+1}, проектируя

полученный вектор $\bar{y}$ на (выпуклое) множество

$$S = \left\{ y \mid y = (y_1, \ldots, y_K);\ \sum_{k=1}^{K} y_k = b \right\}.$$

Эта задача, имеющая вид

$$\min \| y - \bar{y} \|^2,$$
$$\sum_{k=1}^{K} y_k = b,$$

сводится для каждой компоненты вектора y к проектированию точки пространства $\mathbf{R}^k$ на некоторую гиперплоскость (при этом на переменные y нет никаких ограничений типа положительности).

Хельд, Вольфе и Краудер [20] были первыми, кто предложил использовать методы субградиента в алгоритмах разложения по ресурсам.

Кеннингтон и Шалаби [25] применили затем этот метод — с большим успехом — к решению задачи о мультипотоках на графах.

3.5. Распространение метода. Методы п. 3.4 без труда обобщаются на случай больших разложимых нелинейных задач типа

$$\sum_{k=1}^{K} f_k(x_k) \to \min,$$
$$\sum_{k=1}^{K} g_k(x_k) = b,$$
$$x_k \in X_k, \qquad k = 1, \ldots, K,$$

при условии, что подзадачи, возникающие при распределении правых частей $y = (y_1, \ldots, y_K)$:

$$v_k(y_k) = \min f_k(x_k),$$
$$g_k(x_k) = y_k,$$
$$x_k \in X_k,$$

суть задачи выпуклого математического программирования.

На самом деле это и есть условие того, что оптимальные двойственные переменные $\bar{u}_k$ приводят к субградиентам функции $v_k(y_k)$ в точке $\bar{y}_k$.

Тогда остается решить задачу

$$v(y) = \sum_{k=1}^{K} v_k(y_k) \to \min,$$
$$\sum_{k=1}^{K} y_k = b.$$

Если функция v выпукла, а вектор $-\bar{u}=(-\bar{u}_1, \ldots, -\bar{u}_k)$ есть субградиент функции v в точке $\bar{y}$, то можно использовать, как и в п. 3.4,

— либо обобщенное линейное программирование,

— либо алгоритм субградиента.

§ 4. Разложение с помощью разделения переменных (алгоритм Бендерса)

4.1. Постановка задачи. В этом параграфе мы будем интересоваться *исключительно разложимыми задачами типа 2, иначе говоря,* содержащими только соединенные переменные.

Следовательно, эти задачи в полной общности имеют вид

$$\min z = cx + fy,$$
$$Dx + Fy = d,$$
$$x \geqslant 0,$$
$$y \in Y \subset \mathbf{R}^m.$$

Матрица

$$D = \begin{bmatrix} D_1 & & 0 \\ & D_2 & \\ 0 & & \ddots \\ & & & D_K \end{bmatrix}$$

блочно-диагональна, и она индуцирует разбиения на блоки — $(x_1, \ldots, x_K)$ для вектора x, $(c_1, \ldots, c_k)$ для вектора c, $\begin{bmatrix} F_1 \\ \cdots \\ F_K \end{bmatrix}$ для матрицы F и $\begin{bmatrix} d_1 \\ \cdots \\ d_K \end{bmatrix}$ для вектора правых частей d.

Множество Y возможных векторов y может быть определено различными способами (см. п. 4.7). Для определенности можно предположить сначала, что Y есть все $\mathbf{R}^m$.

Различая разные блоки в матрице ограничений, мы можем переписать задачу в виде

$$\min \sum_{k=1}^{K} c_k x_k + fy,$$
$$D_1 x_1 + F_1 y = d_1,$$
$$D_2 x_2 + F_2 y = d_2,$$
$$\cdots\cdots\cdots\cdots$$
$$D_K x_K + F_K y = d_K,$$
$$x_1, \ldots, x_K \geqslant 0, \quad y \in Y.$$

Мы можем сразу сделать следующее замечание: с того момента, как значение переменных y фиксировано, и как только различные системы

$$D_k x_k = d_k - F_k y$$

имеют — все как одна — решения $x_k \geqslant 0$, так оказывается, что решение задачи, ограниченной на переменные x, разлагается на K независимых задач линейного программирования (вспомогательных или служебных задач).

Именно это наблюдение и лежит в основе техники разложения с помощью разделения переменных, которую мы сейчас изложим.

Благодаря тому обстоятельству, что всякий раз, когда приходится решать при фиксированном y задачу

$$\begin{aligned} &\min cx, \\ &Dx = d - Fy, \qquad Q(y) \\ &x \geqslant 0 \end{aligned}$$

(или двойственную к ней), то это сводится к разложению на K независимых подзадач, мы в дальнейшем удовлетворимся изучением задачи в матричной форме (P_2), избегая усложнений в обозначениях, связанных с нумерацией блоков.

4.2. Использование необходимых и достаточных условий Фаркаша и Минковского. Чтобы иметь возможность применять излагаемую ниже технику разделения переменных, нельзя брать сколько попало переменных y из Y. Нужно по крайней мере, чтобы задача

$$\begin{aligned} &\min cx, \\ &Dx = \bar{d} = d - Fy, \qquad Q(y) \\ &x \geqslant 0, \end{aligned}$$

имела непустое множество решений.

Для преобразования этого условия используем теорему Фаркаша и Минковского (приложение 1). Свяжем с каждым ограничением задачи $Q(y)$ дуальное переменное u_i (без ограничений на его знак) и обозначим через u вектор-строку дуальных переменных (размерность этого вектора равна числу ограничений в $Q(y)$). Тогда теорема Фаркаша и Минковского формулируется следующим образом:

$Q(y)$ имеет решение $x \geqslant 0$ тогда и только тогда, когда $u(d - Fy) \leqslant 0$ для всех векторов-строк u, удовлетворяющих условию $uD \leqslant 0$.

Так как конус $C = \{u | uD \leqslant 0\}$ многогранен, то он имеет конечное число порождающих, которые будут обозначаться $u^1, \ldots, u^p$ (любой элемент $u \in C$ есть линейная комбинация с неотрицательными коэффициентами элементов u^i $(i = 1, \ldots, p)$).

Необходимое и достаточное условие в теореме Фаркаша и Минковского эквивалентно тогда следующей системе неравенств:

$$\begin{aligned} &u^1(d - Fy) \leqslant 0, \\ &u^2(d - Fy) \leqslant 0, \qquad (III) \\ &\cdots\cdots\cdots \\ &u^p(d - Fy) \leqslant 0. \end{aligned}$$

Вообще говоря, в этой системе содержится огромное количество неравенств — по числу порождающих минимального конуса C.

Если система (III) не имеет решения по y, то это, по построению, означает, что не существует таких $y \in \mathbf{R}^m$, для которых задача $Q(y)$ имеет решение. Следовательно, и сама задача (P_2) не имеет решения.

В дальнейшем мы будем предполагать, что система (III) имеет решение.

Замечание. Если задана блочно-диагональная форма матрицы D, то конус C разлагается в прямую сумму конусов

$$C_k = \{u_k | u_k D_k \leqslant 0\},$$

где u_k — часть вектора u, соответствующая блоку k.

Иначе говоря,

$$C = \{u | u = (u_1, \ldots, u_K);\ u_1 \in C_1, \ldots, u_K \in C_K\}.$$

4.3. Эквивалентность с задачей линейного программирования большей размерности по переменным y. Пусть R — множество векторов $y \in Y$, удовлетворяющих системе (III) (если (P_2) имеет решение, то необходимо $R \neq \phi$). Тогда задача (P_2) эквивалентна задаче

$$\min_{y \in R} \left\{ fy + \min_x \{cx \mid Dx = d - Fy;\ x \geqslant 0\} \right\}.$$

Это полностью соответствует исходной идее, состоящей в том, чтобы фиксировать y, затем решить задачу линейного программирования $Q(y)$ (разложением на K независимых подзадач), затем выбрать лучшее значение для y, и т. д.

Для фиксированного $y \in R$ задача, двойственная к задаче

$$\begin{aligned} &\min cx, \\ &Dx = d - Fy, \qquad Q(y) \\ &x \geqslant 0, \end{aligned}$$

может быть переписана в виде

$$\begin{aligned} &\max v(d - Fy), \\ &vD \leqslant c, \qquad Q^*(y) \\ &v \text{ любого знака}, \end{aligned}$$

где v обозначает вектор дуальных переменных, связанных с ограничениями задачи $Q(y)$.

Политоп ограничений задачи $Q^*(y)$, а именно политоп $V = \{v | vD \leqslant c\}$, не зависит от выбора y, и определенные выше $(u^1, \ldots, u^p)$ суть его экстремальные лучи.

Если V пуст, то, согласно теореме двойственности,

— либо задача $Q(y)$ не имеет решения,

— либо $Q(y)$ не ограничена.

Но по определению $Q(y)$ имеет решение для $y \in R$; следовательно, если V пусто, то это означает, что $Q(y)$ не ограничена для всех

значений $y \in R$. Следовательно, в этом случае и сама задача (P_2) не ограничена.

Тогда, договорившись приписывать максимуму $Q^*(y)$ значение $-\infty$ в случае, если задача $Q^*(y)$ не имеет решения, и используя теорему двойственности, мы можем переписать задачу (P_2) в виде

$$\min_{y \in R} \{fy + \max \{v(d - Fy) \mid vD \leqslant c\}\}.$$

Максимум $Q^*(y)$ не может быть неограниченным $(+\infty)$ для $y \in R$. (Действительно, если бы $Q^*(y)$ была не ограничена, то отсюда следовало бы, что $Q(y)$ не имеет решения, что противоречило бы тому, что $y \in R$.)

Следовательно, этот максимум достигается в некоторой крайней точке политопа $V = \{v \mid vD \leqslant c\}$.

Предположим, что V непуст, и обозначим через $v^1, \ldots, v^q$ крайние точки (их число конечно) политопа V.

Тогда (P_2) можно переписать в виде

$$\min_{y \in R} \left\{fy + \max_{i=1,\ldots,q} \{v^i(d - Fy)\}\right\},$$

и эта задача оказывается эквивалентной задаче **линейного программирования**

$$\begin{gathered} \min z, \\ z \geqslant fy + v^1(d - Fy), \\ z \geqslant fy + v^2(d - Fy), \\ \cdots\cdots\cdots \\ z \geqslant fy + v^q(d - Fy), \\ y \in R. \end{gathered}$$

Выражая условие $y \in R$ с помощью системы неравенств (III), получаем эквивалентность задачи (P_2) решению задачи линейного программирования (главной задачи):

$$\begin{gathered} \min_{(z,y)} z, \\ fy + v^1(d - Fy) - z \leqslant 0, \\ \cdots\cdots\cdots\cdots \\ fy + v^q(d - Fy) - z \leqslant 0, \\ u^1(d - Fy) \leqslant 0, \\ \cdots\cdots\cdots\cdots \\ u^p(d - Fy) \leqslant 0, \\ y \in Y. \end{gathered} \qquad (PM)$$

Число ограничений в этой задаче, равное числу крайних точек и экстремальных лучей политопа V, вообще говоря, очень велико. Однако известно, что в оптимуме число выполняемых (активных) ограничений не превосходит $m + 1$, где m — число переменных y. На этом основании задача (PM) может быть решена при явном выражении относительно небольшого числа этих ограничений — в соот-

ветствии с принципом обобщенного линейного программирования (§ 1).

4.4. Решение с помощью обобщенного линейного программирования: алгоритм Бендерса. Для применения обобщенного линейного программирования нужно с полной строгостью рассмотреть задачу, двойственную к задаче (PM). Но так как для понимания метода гораздо более предпочтительно работать с самой задачей (PM), то мы сейчас введем алгоритм порождения ограничений.

Однако не следует упускать из виду, что в практической реализации этого метода на самом деле решается именно задача, двойственная к (PM), и притом решается методом порождения столбцов.

Предположим, что на некотором этапе только несколько ограничений (PM) известно явным образом (это могут быть различные ограничения, порожденные в ходе предыдущих шагов вычисления).

Таким образом, возникает ограниченная главная задача (PR), образованная с помощью подмножеств $I \subset \{1, \ldots, p\}$ и $J \subset \{1, \ldots, q\}$ ограничений задачи (PM):

$$\begin{gathered} \min z, \\ u^i(d - Fy) \leqslant 0 \qquad \forall i \in I, \\ fy + v^j(d - Fy) - z \leqslant 0 \qquad \forall j \in J, \\ y \in Y. \end{gathered} \tag{PR}$$

Для $Y = \mathbf{R}^m$ задача (PR) есть задача линейного программирования, которая может быть решена с помощью симплексного алгоритма (на практике, как мы уже сказали, должна решаться именно задача, двойственная к (PR)).

Пусть $(\bar{y}, \bar{z})$ — оптимальное решение задачи (PR). (Если задача (PR) не имеет решения, то и задача (PM) не имеет решения, и точно так же и (P_2) не имеет решения. В этом случае продолжать вычисления бесполезно.)

Так как (PR) образована из (PM) устранением некоторого количества ограничений, то $\bar{z}$ является минорантой для оптимального значения z^* задачи (PM) и (P_2): $\bar{z} \leqslant z^*$.

Необходимое и достаточное условие того, чтобы $(\bar{y}, \bar{z})$ представляло оптимальное решение задачи (PM) и потому задачи (P_2), состоит в том, что $(\bar{y}, \bar{z})$ удовлетворяет всем ограничениям задачи (PM), которые не содержатся явно в (PR). Как проверить это условие?

Мы сейчас увидим, что достаточно просто решить для $y = \bar{y}$ задачу

$$\begin{gathered} \max u(d - F\bar{y}), \\ uD \leqslant c, \\ u \text{ любого знака.} \end{gathered} \tag{$Q^*(\bar{y})$}$$

(Напомним, что решение $Q^*(\bar{y})$ приводит к разложению на K независимых подзадач, так как эта задача двойственна к $Q(\bar{y})$.)

Если исключить ситуацию, в которой $Q^*(\bar{y})$ не имеет решения (тогда либо (P_2) не имеет решения, либо (P_2) не ограничена, и вычисление останавливается), то могут представиться три случая.

Случай 1. Оптимальное значение $Q^*(\bar{y})$ не ограничено $(+\infty)$. Симплексный алгоритм, примененный к $Q^*(\bar{y})$, приводит тогда к такому экстремальному лучу $\bar{u}$ политопа V, что

$$\bar{u}(d - F\bar{y}) > 0, \quad \bar{u}D \leqslant 0.$$

Следовательно, ограничение $\bar{u}(d - Fy) \leqslant 0$ не удовлетворяется текущим решением $\bar{y}$ задачи (PR). Следовательно, пара $(\bar{y}, \bar{z})$ не является решением задачи (PM). Тогда ограничение $\bar{u}(d - Fy) \leqslant 0$ должно быть присоединено к (PR) для образования новой ограниченной задачи («расширенной»).

Случай 2. Оптимум задачи $Q^*(\bar{y})$ имеет конечное значение (следовательно, он достигается в крайней точке $\bar{v}$ политопа V), и выполняется неравенство

$$f\bar{y} + \bar{v}(d - F\bar{y}) - \bar{z} \leqslant 0.$$

Тогда можно написать

$$f\bar{y} + v^j(d - F\bar{y}) - \bar{z} \leqslant 0$$

для всех $j = 1, \ldots, q$ (так как $\bar{v}$ — крайняя точка V, минимизирующая значение величины $v(d - F\bar{y})$ на множестве крайних точек).

С другой стороны, имеем $u^i(d - F\bar{y}) \leqslant 0$ для всех экстремальных лучей u^i политопа V $(i = 1, \ldots, p)$. Действительно, если бы выполнялось неравенство $u^i(d - F\bar{y}) > 0$ для некоторого u^i, то тогда оптимум задачи $Q^*(\bar{y})$ был бы не ограничен $(+\infty)$.

Отсюда следует, что $(\bar{y}, \bar{z})$ есть оптимальное решение задачи (PM); следовательно, $\bar{y}$ есть оптимальное решение задачи (P_2), и алгоритм завершен.

Случай 3. Оптимум $Q^*(\bar{y})$ имеет конечное значение и достигается в крайней точке $\bar{v}$ политопа V, но, в отличие от случая 2, выполняется неравенство

$$f\bar{y} + \bar{v}(d - F\bar{y}) - \bar{z} > 0.$$

Это показывает, что ограничение

$$fy + \bar{v}(d - Fy) - z \leqslant 0$$

не удовлетворяется текущим решением $(\bar{y}, \bar{z})$ задачи (PR).

Следовательно, к задаче (PR) следует добавить ограничение

$$fy + \bar{v}(d - Fy) - z \leqslant 0,$$

чтобы образовать новую ограниченную задачу («расширенную»).

Итог.

До тех пор пока не будет удовлетворяться тест на оптимальность (случай 2), можно добавлять к ограниченной задаче ограничения, которым текущее решение $(\bar{y}, \bar{z})$ не удовлетворяет. Тогда новая ограниченная задача решается по (y, z), и т. д.

Если на некотором шаге ограниченная задача не имеет решения, то задача (P_2) не имеет решения, и алгоритм останавливается.

Наконец, если политоп V пуст (что устанавливается уже на первой итерации алгоритма, когда мы в первый раз решаем задачу $Q^*(y)$), и если (P_2) имеет решение, то можно заключить, что (P_2) не ограничена, и работа алгоритма завершена.

Замечание. Если напомнить, что эта процедура порождения ограничений для задачи (PM) есть не что иное, как процедура порождения столбцов для двойственной к ней задачи, то мы видим, что порождающий алгоритм из § 1 состоит здесь в симплексном алгоритме, примененном к задаче $Q^*(\bar{y})$.

4.5. **Сходимость.** Легко установить конечную сходимость этого метода. Она следует из того факта, что (PM) имеет конечное число ограничений и все последовательно порождаемые ограничения необходимо различны между собой.

Для задач очень большого размера сходимость может оказаться относительно медленной; с другой стороны, ошибки округления приводят к тому, что тест на остановку (случай 2), вообще говоря, не может быть достигнут. Следовательно, может оказаться интересным выработать другой, менее требовательный, тест на остановку. Но его легко получить через двустороннюю оценку оптимума z^* задачи (P_2). На каждой итерации, как мы видели, оптимальное решение $\bar{z}$ ограниченной задачи является минорантой для z^*. С другой стороны, если на некоторой данной итерации задача $Q(\bar{y})$ имеет оптимальное решение $\bar{x}$, то $(\bar{x}, \bar{y})$ есть оптимальное решение задачи (P_2) и $f\bar{y} + c\bar{x}$ есть мажоранта для z^*. Тогда итерации можно прервать, как только интервал $\bar{z} \leqslant z^* \leqslant f\bar{y} + c\bar{x}$ станет меньше некоторой заданной заранее точности ε.

Отметим также, что можно доказать сходимость метода Бендерса к приближенному решению задачи (P_2) даже и в случае, если подзадачи $Q^*(y)$ решаются не точно, а лишь приближенным образом (см. [33]).

4.6. **Связь с алгоритмом разложения Данцига — Вольфе.** Пусть структура задачи (P_2) задана. Можно задаться вопросом, не будет ли более разумно искать решение задачи (P_2), применяя алгоритм разложения Данцига — Вольфе (§ 2) к задаче, двойственной к (P_2), поскольку эта последняя задача имеет требуемую структуру (только соединенные ограничения).

На самом деле можно показать, что в случае, когда $Y = \mathbf{R}^m$, оба подхода абсолютно эквивалентны. Иначе говоря, алгоритм Бендерса есть не что иное, как алгоритм разложения Данцига — Вольфе, примененный к задаче, двойственной к (P_2) (см., например, [27], гл. 7, для детального изучения этой эквивалентности).

Однако, как мы сейчас увидим, интерес к алгоритму Бендерса (который появился несколько позже алгоритма Данцига — Вольфе) основан, по существу, на том факте, что этот алгоритм применим не только к большим разложимым линейным задачам, но также и ко многим другим, например к смешанным линейным задачам и к частично нелинейным задачам.

4.7. **Другие применения алгоритма Бендерса.** В этих приложениях матрица D не всегда имеет блочно-диагональную структуру:

тогда решение задачи $Q(y)$ не имеет связи с разложением. Тем не менее алгоритм Бендерса очень полезен в этих задачах, поскольку в них возможна техника разделения переменных,— если есть две группы переменных, играющие существенно различные роли.

1°) Одной из наиболее важных областей применений является смешанное линейное программирование — иначе говоря, решение задач линейного программирования, в которых часть переменных подвергается ограничению принимать только целые значения. Именно в этом контексте и было, кроме того, предложено применение этого метода при его первом описании [5].

Тогда речь идет о решении задач типа (P_2), где множество Y возможных векторов y ограничено множеством $\mathbf{Z}^m$ всех m-векторов с целыми координатами (или подмножеством множества $\mathbf{Z}^m$).

Легко проверить, что принцип алгоритма Бендерса остается неизменным, за исключением того, что ограниченная главная задача, которую нужно решать на каждом шаге итерации, является теперь «чистой» целочисленной задачей линейного программирования (т. е. задачей, содержащей только целочисленные переменные) типа

$$\begin{aligned} &\min z, \\ &u^i(d-Fy)\leqslant 0 \qquad \forall i\in I, \\ &fy+v^j(d-Fy)-z\leqslant 0 \qquad \forall j\in J, \\ &y\in Y\subset \mathbf{Z}^m. \end{aligned} \tag{PNE}$$

Это позволяет преобразовать решение смешанной задачи в решение ряда «частных» целочисленных задач.

Очевидно, что решение задачи (PNE) может привести на практике к трудностям, если m (число переменных y) очень велико, поскольку техника целочисленного программирования гораздо менее эффективна, чем техника «непрерывного» линейного программирования (см. п. 7). Однако, несмотря на эти ограничения, алгоритм Бендерса с успехом применялся в многочисленных приложениях смешанного линейного программирования (см., например, [14]).

2°) Другой важный класс приложений составляют частично нелинейные задачи вида

$$\begin{aligned} &\min z = cx + f(y), \\ &Dx + F(y) = d, \\ &x \geqslant 0, \\ &y \in Y, \end{aligned} \tag{P_2'}$$

где f — нелинейная функция переменных y, принимающая вещественные значения, а $F(y)$ — вектор (той же размерности, что и d), все компоненты которого суть нелинейные функции переменных y. В этом случае множество Y есть либо все $\mathbf{R}^m$, либо выпуклое подмножество $\mathbf{R}^m$.

И в этом случае принцип алгоритма Бендерса остается практически неизменным. Единственная разница состоит в том, что ограниченная главная задача становится нелинейной задачей с ограни-

чениями следующего вида:

$$\begin{aligned} &\min z, \\ &u^i(d - F(y)) \leqslant 0 \qquad \forall i \in I, \\ &f(y) + v^j(d - F(y)) - z \leqslant 0 \qquad \forall j \in J, \\ &y \in Y \subset \mathbf{R}^m. \end{aligned} \tag{PNL}$$

Задача (*PNL*) может оказаться более или менее трудной для решения в зависимости от числа переменных y и вида функций f и F.

Интерес к алгоритму Бендерса связан в этом случае с тем, что он позволяет решить задачу (P_2'), разделяя линейную и нелинейную части.

§ 5. Примеры приложений методов разложения: задачи оптимизации больших сетей

Чтобы проиллюстрировать ценность различных описанных в настоящей главе методов разложения, а также показать способы, которыми эти методы могут быть вовлечены в решение практических задач, мы предлагаем рассмотреть несколько задач, выбранных среди наиболее характерных в области оптимизации больших сетей (транспортные сети, сети телекоммуникации или телеинформатики и т. д.).

Прежде всего мы рассмотрим задачи о согласованных мультипотоках и покажем, как эти задачи могут быть решены с помощью метода разложения Данцига — Вольфе. Затем мы рассмотрим задачи о мультипотоках минимальной стоимости (с ограничениями на емкость), для которых мы убедимся, что методы разложения по действию ресурсов приводят к простым и эффективным алгоритмам решения. Наконец, задача оптимального синтеза сети (пробегаемой не одновременными мультипотоками) предоставляет хороший пример приложения метода разложения Бендерса. За более углубленным изложением, касающимся задач о мультипотоках на графах, мы отсылаем к книге [18], гл. 6, и к обзорным статьям [24, 3].

5.1. Задачи о согласованных мультипотоках. Эти задачи типичным образом возникают в следующем контексте. Изучаемая сеть представляется ориентированным графом $G = [X, U]$, где X — множество вершин (начал и концов путей движения) и U — множество дуг.

(Если G — не ориентированный граф, то можно свести дело к ориентированному случаю, выбирая произвольную ориентацию на каждой стрелке.)

Пусть $|X| = N$ — число вершин, а $|U| = M$ — число дуг.

Сопоставим каждой дуге $u = (i, j)$, представляющей множество транспортных возможностей (в транспортной сети) или возможностей передачи информации (в сети телекоммуникаций) между началом $i \in X$ и концом $j \in X$, величину, называемую *емкостью*, т. е. вещественное число $c_u \geqslant 0$.

С другой стороны, пусть задано множество из K *запросов* (например, транспортных запросов в сети путей сообщения), где для каждого $k = 1, \ldots, K$ обозначим через r_k значение полного запроса (полный поток транспорта), который должен быть пропущен через сеть между двумя данными вершинами s_k и t_k.

Таким образом, каждый из запросов отдельно представляется простым потоком со значением r_k между началом s_k (истоком) и концом t_k (стоком), но мы предположим, что различные потоки имеют разную природу. На практике это означает, что на каждой дуге добавляется соответствующее абсолютное значение, а не значение со знаком.

Замечание 1. Предыдущая модель может быть легко изменена, так как можно рассматривать как единый простой поток запросы с общим началом в данном графе. Таким образом, всегда можно ограничиться случаем, когда число K простых потоков, составляющих мультипоток, не превосходит числа N вершин графа (см., например, [18], гл. 6).

Задача о согласованном мультипотоке состоит тогда в решении вопроса: достаточно ли емкостей, помещенных на сеть, для одновременного стока всех запросов?

Сейчас мы увидим, что эта задача может быть представлена в виде задачи линейного программирования, вообще говоря, большой размерности, причем обладающей специальной структурой.

Пусть A — матрица инцидентности между вершинами и дугами в графе G: это — матрица с N строками и M столбцами, где каждый столбец соответствует дуге $u = (i, j) \in U$ и содержит только два ненулевых элемента со значениями $+1$ и -1 соответственно в строках i и j.

Обозначим через δ_k N-вектор, все компоненты которого равны нулю, кроме компонент s_k и t_k, со значениями соответственно $+1$ и -1.

С другой стороны, обозначим через ψ_u^k значение простого потока k $(1 \leqslant k \leqslant K)$ на дуге $u \in U$ (ψ_u^k может быть величиной любого знака).

Для того чтобы набор $\psi^k = (\psi_u^k)_{u \in U}$ образовывал простой поток, необходимо и достаточно, чтобы выполнялся закон сохранения потока в каждой вершине (первый закон Кирхгофа), что выражается матричным соотношением

$$A\psi^k = r_k \delta_k.$$

Тогда речь идет о том, чтобы найти такие числа ψ_u^k $(u = 1, \ldots, M;\ k = 1, \ldots, K)$, чтобы на каждой дуге сумма абсолютных значений потоков не превосходила емкости дуги, т. е. о решении системы

$$\sum_{k=1}^{K} |\psi_u^k| \leqslant c_u \qquad \forall u \in U,$$

$$A\psi^k = r_k \delta_k \qquad \forall k = 1, \ldots, K.$$

Чтобы свести эту задачу к системе обычных линейных уравнений и неравенств, достаточно осуществить следующее преобразование.

Пусть $G' = [X, U']$ — ориентированный граф, построенный на том же множестве вершин X, что и G, множество дуг которого получается приписыванием каждой дуге $u = (i, j)$ из графа G двух дуг $u^+ = (i, j)$ и $u^- = (j, i)$ из G'.

Следовательно, имеем $U' = U^+ \cup U^-$, где U^+ (соответственно U^-) — множество дуг графа G' вида u^+ (соответственно u^-).

Обозначим тогда через φ^k_+ неотрицательный M-вектор с компонентами

$$\varphi^k_{u^+} = \psi^k_u, \qquad \psi^k_u \geqslant 0,$$
$$\varphi^k_{u^+} = 0, \qquad \psi^k_u < 0,$$

а через φ^k_- — неотрицательный M-вектор с компонентами

$$\varphi^k_{u^-} = -\psi^k_u, \qquad \psi^k_u \leqslant 0,$$
$$\varphi^k_{u^-} = 0, \qquad \psi^k_u \geqslant 0;$$

тогда можно написать $\forall u \in U$

$$|\psi^k_u| = \varphi^k_{u^+} + \varphi^k_{u^-}.$$

Обозначая через $A' = [A, -A]$ матрицу инцидентности между вершинами и дугами в G' (где подматрица A соответствует дугам из U^+, а подматрица $-A$ соответствует дугам из U^-) и обозначая $\varphi^k = \begin{bmatrix} \varphi^k_+ \\ \varphi^k_- \end{bmatrix}$, можем написать

$$A'\varphi^k = A\varphi^k_+ - A\varphi^k_- = A\psi^k = r_k\delta_k,$$

что доказывает, что φ^k — поток с неотрицательными компонентами и со значением r_k между s_k и t_k на G'.

Задача о согласованном мультипотоке сводится тогда к решению системы линейных уравнений и неравенств

$$\sum_{k=1}^{K} (\varphi^k_+ + \varphi^k_-) \leqslant c,$$
$$A'\varphi^k = r_k\delta_k \qquad \forall k = 1, \ldots, K,$$
$$\varphi^k = \begin{bmatrix} \varphi^k_+ \\ \varphi^k_- \end{bmatrix} \geqslant 0 \qquad \forall k = 1, \ldots, K.$$

Заметим тогда, что эта задача может быть сведена к решению задачи линейного программирования относительно набора

переменных (φ, η):

$$\min \eta,$$

$$\sum_{k=1}^{K} (\varphi_+^k + \varphi_-^k) - \eta \cdot \mathbf{1} \leqslant c, \quad (6)$$

$$A'\varphi^k = r_k \delta_k \qquad \forall k = 1, \ldots, K, \quad (7) \qquad (MC)$$

$$\varphi^k = \begin{bmatrix} \varphi_+^k \\ \varphi_-^k \end{bmatrix} \geqslant 0 \qquad \forall k = 1, \ldots, K, \quad (8)$$

где $\mathbf{1}$ обозначает M-вектор, все компоненты которого равны 1; η — вещественное переменное, имеющее следующий смысл: мы ищем такое наименьшее значение η, что сеть, снабженная емкостями $c_u + \eta$ на каждой дуге $u \in U$, допускает допустимый мультипоток.

Пусть (φ^*, η^*) — оптимальное решение задачи (MC).

Если $\eta^* > 0$, то мы можем заключить, что исходная задача о согласованном мультипотоке не имеет решения.

Если $\eta^* \leqslant 0$, то задача имеет решение, и φ^* — такое решение.

Как следует из рис. 4, задача линейного программирования (MC) имеет весьма специфичную структуру (структуру типа 1, см.

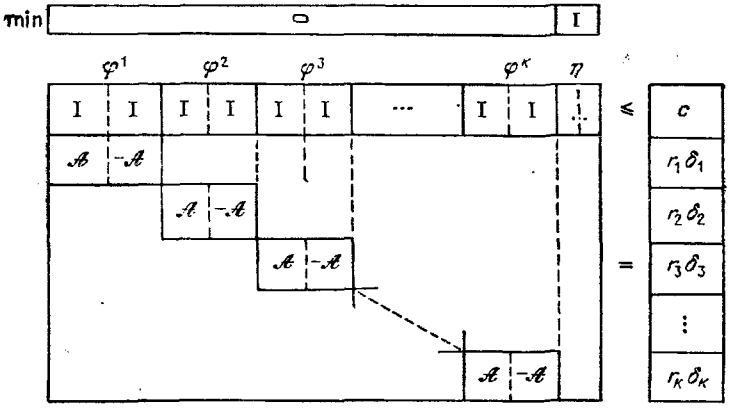


Рис. 4. Структура задачи о согласованном мультипотоке. Матрица A есть матрица инцидентности вершин и дуг графа

п. 2.1). Ограничения (7) имеют блочно-диагональную форму, а ограничения (6) суть соединенные ограничения. Эта структура полностью согласуется, таким образом, с разложением по методу Данцига — Вольфе.

Заметим здесь, что обращение к некоторому методу разложения тем более необходимо, что число переменных и ограничений (соответственно $2KM$ и $M + KN$) может оказаться очень большим. Тогда для графа, содержащего $N = 50$ вершин и $M = 200$ дуг с $K = N = 50$, получается задача с 20 000 переменными и 2700 ограничениями.

Чтобы объяснить, каким образом метод разложения Данцига — Вольфе может быть здесь применен, рассмотрим дуальную задачу, получаемую лагранжевым ослаблением соединенных ограничений (6) и связыванием с ними вектора (строки) множителей Лагранжа $\pi = (\pi_u)_{u \in U}$.

Для любого фиксированного $\pi \geqslant 0$ функция Лагранжа равна

$$L(\varphi, \eta, \pi) = \eta(1 - \pi \cdot 1) + \sum_{k=1}^{K} \pi\left(\varphi_+^k + \varphi_-^k\right) - \pi c,$$

а дуальная функция определяется формулой

$$w(\pi) = \min_{(\varphi, \eta)} \{L(\varphi, \eta, \pi)\},$$

где минимум берется по множеству функций φ, удовлетворяющих ограничениям (7) и (8).

Так как η — переменное произвольного знака, то $w(\pi)$ может принимать конечные значения только при условии, что $\pi \cdot 1 = 1$ (условие нормализации).

При этом условии получаем формулу разложения

$$w(\pi) = -\pi c + \sum_{k=1}^{K} \min_{\varphi^k \in \Phi^k} \{\pi\left(\varphi_+^k + \varphi_-^k\right)\},$$

где $\Phi^k = \{\varphi^k \mid A'\varphi^k = r_k \delta_k,\ \varphi^k \geqslant 0\}$.

Следовательно, для данного $\pi \geqslant 0$ вычисление величины $w(\pi)$ сводится к решению K независимых друг от друга подзадач

$$\begin{aligned} &\pi\left(\varphi_+^k + \varphi_-^k\right) \to \min, \\ &A'\varphi^k = r_k \delta_k, \qquad (SP_k) \\ &\varphi^k = \begin{bmatrix} \varphi_+^k \\ \varphi_-^k \end{bmatrix} \geqslant 0. \end{aligned}$$

Для произвольного k $(1 \leqslant k \leqslant K)$ задача (SP_k) есть задача о простом потоке с минимальной стоимостью без ограничений на емкости; следовательно, ее можно свести к задаче о кратчайшем пути.

Более того, для определения оптимального решения $\bar{\varphi}^k(\pi)$ задачи (SP_k) мы начинаем с поиска кратчайшего пути между s_k и t_k в графе G', приписывая дугам u^+ и u^- длину $\pi_u \geqslant 0$.

Тогда решение $\bar{\varphi}^k(\pi)$ есть вектор, компоненты которого равны r_k на дугах кратчайшего пути и нулю на других дугах.

Лагранжево ослабление приводит, таким образом, к разложению на K подзадач о кратчайшем пути. Тогда получаем

$$w(\pi) = -\pi c + \sum_{k=1}^{K} \pi\left(\bar{\varphi}_+^k(\pi) + \bar{\varphi}_-^k(\pi)\right),$$

откуда, обозначая

$$\rho(\pi) = \sum_{k=1}^{K} \left[\bar{\varphi}_+^k(\pi) + \bar{\varphi}_-^k(\pi)\right],$$

(M — вектор объединенных потоков на дугах графа), получаем, что

$$w(\pi) = \pi(\rho(\pi) - c).$$

Так как в графе G' имеется лишь конечное число элементарных путей, то число возможных векторов ρ конечно (можно заметить также, что эти векторы соответствуют крайним точкам многогранника Ф, определенного условиями (7) и (8)).

Дуальная функция $w(\pi)$ может рассматриваться, следовательно, как нижняя оболочка конечного семейства аффинных функций: мы заново получаем, что w есть вогнутая, не всюду дифференцируемая функция и что $\forall \pi \geqslant 0$ вектор $\rho(\pi) - c$ есть субградиент w в точке π. Субградиент w в каждой точке $\pi \geqslant 0$ получается, таким образом, как промежуточный результат вычисления значения $w(\pi)$.

Тогда можно осуществить решение дуальной задачи:

— либо точно (как в п. 2.6) с помощью обобщенного линейного программирования (секущие плоскости), что составляет не что иное, как метод Данцига — Вольфе, примененный к линейной задаче (MC);

— либо приближенно, с помощью алгоритма субградиента (см. гл. 4).

Пусть π^* — оптимальное решение дуальной задачи (D) и $w(\pi^*)$ — оптимальное значение дуальной задачи.

Тогда и исходная, и дуальная задачи имеют решения, и (по теореме двойственности) имеем

$$w(\pi^*) = \eta^*.$$

Для того чтобы задача о согласованном мультипотоке имела решение, необходимо и достаточно, чтобы $w(\pi^*) \leqslant 0$. Если это условие выполнено, то эффективное определение оптимального решения исходной задачи (MC) представляет собой, как видно из п. 2.6, линейную комбинацию последовательно порождаемых алгоритмом решений $\bar{\varphi}^k(\pi)$, а коэффициенты этой линейной комбинации суть оптимальные исходные переменные ограниченной главной задачи.

Замечание 2. Даже приближенное решение дуальной задачи (полученное, например, с помощью алгоритма субградиента) часто представляет интерес само по себе. Например, если полученное приближенное решение $\bar{\pi}$ удовлетворяет соотношению $w(\bar{\pi}) > 0$, то может оказаться, что искомого согласованного мультипотока нет. С другой стороны, во всех случаях величина $w(\bar{\pi})$ дает миноранту оптимального значения задачи (MC), что позволяет оценить качество решений исходной задачи, полученных с помощью приближенных алгоритмов (таких, как алгоритмы, описанные в [31]), и дать интервал для точного оптимального значения.

Замечание 3. Из предыдущих рассуждений можно вывести необходимое и достаточное условие существования согласованного мультипотока на графе. Действительно, условие

$$w(\pi^*) = \max_{\substack{\pi|\pi \cdot 1 = 1 \\ \pi \geqslant 0}} w(\pi) \leqslant 0$$

выполняется тогда и только тогда, когда

$$w(\pi) = \pi(\rho(\pi) - c) \leqslant 0 \qquad \forall \pi \geqslant 0,$$

откуда и следует необходимое и достаточное условие

$$\pi c \geqslant \theta(\pi) \qquad \forall \pi \geqslant 0, \tag{9}$$

где через

$$\theta(\pi) = \pi\rho(\pi) = \sum_{k=1}^{K} \pi\left(\bar{\varphi}_+^k(\pi) + \bar{\varphi}_-^k(\pi)\right)$$

обозначена сумма оптимальных значений K подзадач (SP_k) о кратчайших путях.

На самом деле можно показать, что достаточно установить предыдущее соотношение для конечного числа векторов π (крайних точек политопа решений дуальной задачи для (MC)) — обозначим их $\pi^1, \ldots, \pi^l$. Таким образом, емкость c тогда и только тогда достаточна для допущения существования согласованного мультипотока, когда

$$\begin{aligned} \pi^1 c &\geqslant \theta(\pi^1), \\ \pi^2 c &\geqslant \theta(\pi^2), \\ &\cdots\cdots \\ \pi^l c &\geqslant \theta(\pi^l). \end{aligned}$$

Условие (9) будет использовано в этой форме в п. 5.3 для получения разложения Бендерса в задаче оптимального синтеза сети.

5.2. **Задачи о согласованных мультипотоках с минимальной стоимостью.** Речь идет о расширении предыдущей задачи, в которой с каждой дугой $u \in U$ связывается единичная цена γ_u (стоимость единицы потока по дуге u). Если задача о согласованном мультипотоке имеет решение, то речь идет тогда о том, чтобы найти среди всех возможных решений такое решение, стоимость которого

$$\sum_{u \in U} \gamma_u \sum_{k=1}^{K} |\varphi_u^k|$$

минимальна.

Эта задача часто возникает в приложениях, например, когда нужно учесть стоимость транспорта или передачи информации между различными центрами или когда желательно отдать предпочтение дорогам с малой длиной и т. д.

Если использовать модель из п. 5.1, то наша задача немедленно представляется в виде

$$\begin{aligned} &\sum_{u \in U} \gamma_u \sum_{k=1}^{K} \left(\varphi_{u+}^k + \varphi_{u-}^k\right) \to \min, \\ &\sum_{k=1}^{K} \left(\varphi_+^k + \varphi_-^k\right) \leqslant c, \\ &A'\varphi^k = r_k \delta_{kt} \qquad k = 1, \ldots, K, \\ &\varphi^k = \begin{bmatrix} \varphi_+^k \\ \varphi_-^k \end{bmatrix} \geqslant 0, \qquad k = 1, \ldots, K. \end{aligned} \tag{MCM}$$

Эта задача линейного программирования имеет ту же матрицу ограничений, что и задача (MC). Следовательно, можно аналогичным образом применить к ней метод разложения Данцига — Вольфе. Однако результаты сравнительных вычислений, проведенных в [2], показывают, что особенно эффективными для получения хороших приближенных решений являются методы разложения по ресурсам, использующие алгоритмы субградиента ([20, 25]). Поэтому мы сейчас опишем именно метод разложения по ресурсам.

Свяжем с каждым ограничением по запросу $A'\varphi^k = r_k\delta_k$ искусственное вещественное переменное $\alpha^k \geqslant 0$ с ценой $H > 0$. Для достаточно большого $H > 0$ и в предположении, что задача (MCM) имеет решение, эта задача (MCM) эквивалентна задаче

$$\begin{gathered}
\sum_{u\in U}\gamma_u\sum_{k=1}^{K}\left(\varphi_{u^+}^k+\varphi_{u^-}^k\right)+H\sum_{k=1}^{K}\alpha^k\to\min,\\
\sum_{k=1}^{K}\left(\varphi_+^k+\varphi_-^k\right)\leqslant c,\\
A'\varphi^k+\alpha^k\delta_k=r_k\delta_k,\qquad k=1,\ldots,K,\\
\varphi^k=\begin{bmatrix}\varphi_+^k\\ \varphi_-^k\end{bmatrix}\geqslant 0,\qquad k=1,\ldots,K.
\end{gathered}\tag{MCM}'$$

(Действительно, для достаточного большого $H > 0$ все искусственные переменные должны обращаться в нуль в оптимуме задачи $(MCM)'$, и таким образом получается оптимальное решение и для задачи (MCM).)

Тогда идея решения состоит в разложении $(MCM)'$ с помощью сопоставления каждому потоку φ^k доли $y^k \geqslant 0$ наличных емкостей на сети, причем должно удовлетворяться условие

$$c=\sum_{k=1}^{K}y^k.$$

Для данного распределения $y=(y^1,\ldots,y^K)\geqslant 0$ определим функцию $g(y)$ формулой

$$g(y)=\sum_{k=1}^{K}g^k(y^k),$$

где $\forall k=1,\ldots,K$ величина $g^k(y^k)$ есть оптимальное значение подзадачи

$$\begin{gathered}
g^k(y^k)=\min\sum_{u\in U}\gamma_u\left(\varphi_{u^+}^k+\varphi_{u^-}^k\right)+H\alpha^k,\\
A'\varphi^k+\alpha^k\delta_k=r_k\delta_k,\\
0\leqslant\varphi_+^k\leqslant y^k,\\
0\leqslant\varphi_-^k\leqslant y^k,\\
\alpha^k\geqslant 0,
\end{gathered}\tag{S_k}$$

которая представляет собой не что иное, как задачу о простом потоке с минимальной стоимостью с ограничениями на емкости на графе G', которую можно эффективно решить с помощью алгоритма типа «out-of-kilter» («в беспорядке», см. [10] и [18], гл. 5). Вычисление $g(y)$ при данном распределении емкостей y также сводится к решению K независимых задач о простых потоках.

Задача $(MCM)'$ эквивалентна тогда поиску распределения емкостей с минимальной стоимостью, иначе говоря, к решению задачи (главной задачи)

$$g(y) \to \min,$$
$$\sum_{k=1}^{K} y^k = c, \qquad (PP)$$
$$y^k \geqslant 0, \qquad k = 1, \ldots, K.$$

Согласно § 3, мы знаем, что g — выпуклая, не всюду дифференцируемая функция y. С другой стороны, если $\forall k = 1, \ldots, K$ обозначить соответственно через μ_+^k и μ_-^k векторы оптимальных дуальных переменных, связанные с ограничениями емкости вида $\varphi_+^k \leqslant y^k$ и $\varphi_-^k \leqslant y^k$ (для каждой дуги u не более одного из этих двух условий обращается в равенство, следовательно, μ_{u+}^k и μ_{u-}^k не могут одновременно оказаться положительными), и если ввести $\mu^k = \mu_+^k + \mu_-^k$, то вектор $\mu = \begin{bmatrix} \mu^1 \\ \cdots \\ \mu^K \end{bmatrix}$ есть субградиент g в точке y.

Напомним, что оптимальные дуальные переменные в задаче о потоке с минимальной стоимостью суть оптимальные решения задачи о напряжении с максимальной стоимостью и получаются одновременно с оптимальными потоками, если использовать исходно-двойственный алгоритм «out-of-kilter» (см. [10]). Следовательно, и в этом случае субградиент в y получается как промежуточный результат вычисления величины $g(y)$. Отыскание минимума g (это функция от $K \times M$ переменных) может быть, таким образом, осуществлено (согласно Хельду, Вольфе и Краудеру [20], которые предложили это впервые) с применением алгоритма субградиента. Как показали Кеннингтон и Шалаби [25], этот метод — один из наиболее эффективных для решения приближенным способом задач большой размерности о мультипотоках (заметим, что его можно применить также и к задаче о согласованном мультипотоке).

5.3. Оптимальный синтез сети, пробегаемой не одновременными мультипотоками. Эта задача возникает в многочисленных приложениях, в частности при отыскании построения сети с минимальной стоимостью по заданным ограничениям безопасности: тогда нужно принять меры предосторожности в сети так, чтобы она могла продолжать действовать (т. е. пропускать как минимум заданную долю номинального дорожного движения) даже в случае выхода из строя одного или нескольких ее элементов (о приложении к сетям телекоммуникации см., например, [35]).

В общем виде задача может быть сформулирована следующим образом (см. [32]).

Пусть задан ориентированный граф $G = [X, U]$ и p заданных мультипотоков (каждый из мультипотоков определен списком составляющих его простых потоков и их значениями); нужно определить емкости $Y_u \geqslant 0$, приписываемые каждой из дуг $u \in U$ графа G таким образом, что:

a) сеть (G, Y) позволяет протечь индивидуально каждому из p заданных мультипотоков;

b) полная стоимость сети

$$\gamma Y = \sum_{u \in U} \gamma_u Y_u$$

минимальна ($\forall u \in U$ величина γ_u есть стоимость единицы потока по дуге u).

Эта задача является обобщением на мультипотоки задачи, исследованной в случае потоков Гомори и Ху [17].

Структура матрицы ограничений в такой задаче представлена на рис. 5. Речь идет о блочно-диагональной структуре, в которой

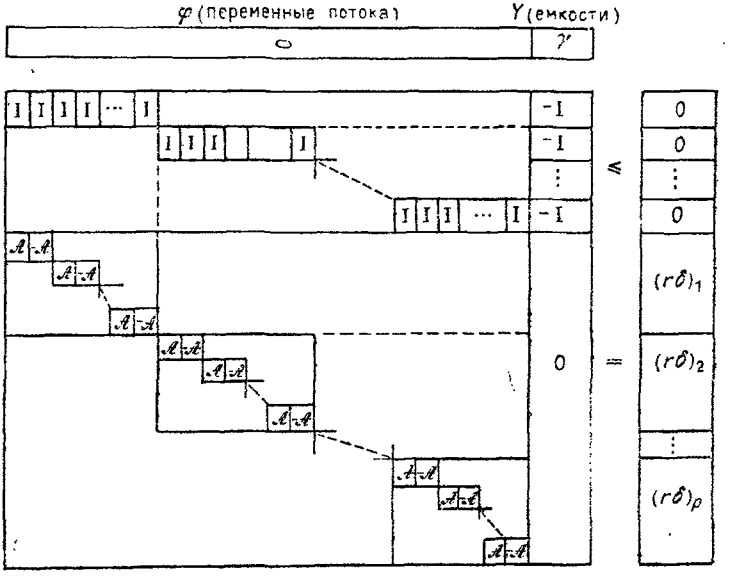


Рис. 5. Структура задачи об оптимальном синтезе сети, пробегаемой p не одновременными мультипотоками: блочно-диагональная структура (p блоков) с соединенными переменными Y

каждый из p блоков соответствует матрице ограничений в задаче о мультипотоке, а переменные емкости Y суть соединенные переменные. Следовательно, эта задача может привести к задачам линейного программирования очень большого размера (порядка $2MNp$ переменных и $p(N^2 + M)$ ограничений). Однако можно заметить,

что если переменные Y фиксированы, то задача распадается на p задач о независимых мультипотоках: следовательно, это наблюдение подсказывает применение техники разложения с помощью разделения переменных (Бендерс).

Как и в методе Бендерса, мы собираемся ограничиться главной задачей только по переменным Y (но содержащей огромное количество ограничений), которая будет затем решаться с помощью обобщенного линейного программирования (порождением ограничений). Используем для этого необходимое и достаточное условие существования *согласованного мультипотока*, данное в замечании 3 в п. 5.1.

Рассмотрим какой-нибудь из p данных мультипотоков, например мультипоток с номером 1. Для того чтобы емкости Y допускали существование согласованного мультипотока, необходимо и достаточно, чтобы

$$\pi Y \geqslant \theta_1(\pi)$$

для конечного числа векторов π, скажем $\pi^{1,1}, \pi^{1,2}, \ldots, \pi^{1,l_1}$. Точно так же для потока с номером 2 должны выполняться соотношения

$$\pi Y \geqslant \theta_2(\pi)$$

для конечного числа векторов π, скажем $\pi^{2,1}, \pi^{2,2}, \ldots, \pi^{2,l_2}$, и т. д.

Задача эквивалентна тогда следующей задаче линейного программирования (главной задаче):

$$\gamma Y = \sum_{u \in U} \gamma_u Y_u \to \min,$$

$$\left.\begin{array}{l} \pi^{1,1} Y \geqslant \theta_1(\pi^{1,1}), \\ \pi^{1,2} Y \geqslant \theta_1(\pi^{1,2}), \\ \cdots\cdots\cdots \\ \pi^{1,l_1} Y \geqslant \theta_1(\pi^{1,l_1}) \end{array}\right\} \text{мультипоток 1,}$$

$$\left.\begin{array}{l} \pi^{2,1} Y \geqslant \theta_2(\pi^{2,1}), \\ \cdots\cdots\cdots \\ \pi^{2,l_2} Y \geqslant \theta_2(\pi^{2,l_2}), \end{array}\right\} \text{мультипоток 2,} \qquad (PM)$$

$$\cdots\cdots\cdots$$

$$\left.\begin{array}{l} \pi^{p,l_p} Y \geqslant \theta_p(\pi^{p,l_p}), \end{array}\right\} \text{мультипоток } p$$

$$Y \geqslant 0$$

(здесь мы разделяем ограничения, связанные с различными мультипотоками).

Эта задача, содержащая только M переменных, но, вообще говоря, огромное количество ограничений, может быть решена с помощью порождения ограничений (обобщенное линейное программирование) следующим образом (см. [32]).

Предположим, что на текущем этапе q ограничений задачи (PM) уже порождены.

Итерация алгоритма состоит тогда в следующем:

а) решается ограниченная главная задача с этими q ограничениями:

$$\begin{gathered}\gamma Y \to \min,\\ \pi^1 Y \geqslant 0(\pi^1),\\ \cdots\cdots\\ \pi^q Y \geqslant \theta(\pi^q),\\ Y \geqslant 0\end{gathered} \qquad (PR)$$

(где для простоты опущены индексы, относящиеся к различным мультипотокам). Пусть $\bar{Y}$ — полученное оптимальное решение;

b) проверяется, достаточны ли емкости Y для (индивидуального) протекания каждого из p заданных мультипотоков. Это сводится к решению (независимо одна от другой) p подзадач о согласованных мультипотоках (разложение). Для каждого мультипотока j $(1 \leqslant j \leqslant p)$ обозначим через $\bar{\pi}^j$ оптимальное решение задачи, двойственной к задаче о согласованном мультипотоке (относительно емкостей $\bar{Y}$), полученное методом п. 5.1.

Могут представиться два случая.

1) Если $\forall j = 1, \ldots, p$ имеем

$$\bar{\pi}^j \bar{Y} \geqslant 0_j(\bar{\pi}^j),$$

то все мультипотоки согласованы, так что $\bar{Y}$ — оптимальное решение задачи (PM).

2) Если для индекса j $(1 \leqslant j \leqslant p)$ выполняется неравенство

$$\bar{\pi}^j \bar{Y} < 0_j(\bar{\pi}^j),$$

то именно мультипоток j не согласован с емкостями $\bar{Y}$. Тогда мы добавляем к задаче (PR) все ограничения вида

$$\bar{\pi}^j Y \geqslant \theta_j(\bar{\pi}^j)$$

для всех таких индексов j, что

$$\bar{\pi}^j \bar{Y} < 0_j(\bar{\pi}^j).$$

Переходим к следующей итерации с полученной новой ограниченной главной задачей.

Мы получим конечную сходимость этого алгоритма (см. п. 4.5), если на каждом шаге итерации и ограниченная задача, и подзадачи о согласованных мультипотоках решаются точно (симплексным алгоритмом). Однако точное решение даже одной только задачи о согласованном мультипотоке в больших размерностях само по себе представляет трудную задачу по причине численных трудностей (ошибки округления, явления вырожденности), и требуемое время вычисления может оказаться запрещенным. Отсюда возникает идея

улучшить эффективность метода Бендерса, решая подзадачи лишь приближенно. Действительно, можно показать (см. [33]), что если на каждом шаге итерации удовлетворяться ε-оптимальным решением различных подзадач, то метод Бендерса сходится к ε-решению задачи (PM), т. е. к решению, удовлетворяющему всем ограничениям задачи (PM) с точностью до ε. Этот результат составляет теоретическое обоснование алгоритма, который был предложен и применен Мину и Серро [34, 35],— алгоритма, состоящего в использовании алгоритмов субградиента для приближенного решения подзадач о согласованных мультипотоках и ограниченной главной задачи. Этот метод обнаруживает высокую эффективность при получении хороших приближенных решений (до нескольких процентов) в задачах оптимального синтеза, имеющих реальные приложения (см. [35]) столь больших размерностей, что ни один из существующих кодексов линейного программирования не мог бы позволить получить точное решение.

Замечание. Если заметить, что сами по себе подзадачи о согласованных мультипотоках могут быть решены, как в п. 5.1, с помощью разложения Данцига — Вольфе, то предыдущий алгоритм может рассматриваться как метод разложения на двух уровнях: на первом уровне разложения (Бендерса) подзадачи суть задачи о согласованных мультипотоках; на втором уровне разложения (Данцига — Вольфе) подзадачи суть задачи о простых потоках с минимальной стоимостью (кратчайшие пути).

СПИСОК ЛИТЕРАТУРЫ

1. Abadie J., Williams A. C. (1968), Dual and Parametric Methods in Decomposition, in: *Recent Advances in Mathematical Programming*, (R. L. Graves, P. Wolfe eds.), McGraw-Hill, New York.
2. Ali A., Helgason R., Kennington J., Lall H. (1980), Computational Comparison among three Multicommodity Network Flow Algorithms, *Operations Research* 28, n° 4, p. 995-1000.
3. Assad A. A. (1978), Multicommodity Network Flows-A Survey, *Networks* 8, p. 37-91.
4. Balas E. (1977), Benders' Method Revisited, *Management Science Research Report*, n° 401, Carnegie-Mellon University, Pittsburgh, Pennsylvania.
5. Benders J. F. (1962), Partitioning procedures for solving mixed variables programming problems, *Numerische Mathematik* 4, p. 238-252.
6. Cohen G. (1978), Optimization by Decomposition and Coordination: a Unified Approach, *IEEE Transactions on Automatic Control*, vol. AC-23, n° 2, p. 222-232.
7. Dantzig G. B., Van Slyke R. M. (1967), Generalized upper Bounding Techniques, *Journal of Computer and System Sciences* 1, p. 213-226.
8*. Dantzig G. B. (1963), *Linear programming and extensions*, Princeton University Press.
9. Dantzig G. B., Wolfe P. (1961), The decomposition algorithm for linear programming, *Econometrica*, vol. 29, n° 4, p. 767-778.
10. Ford L. R., Fulkerson D. R., (1962), *Flows in Networks*, Princeton University Press.
11. Geoffrion A. M. (1968), Primal resource-directive approaches for optimizing nonlinear decomposable systems, *Memorandum RM-5829-PR, the RAND Corp., Santa Monica, Calif. & Operations Research* 18, 1970, p. 375-403.
12. Geoffrion A. M. (1970), Elements of Large scale Mathematical Programming I & II, *Management Science* 16, p. 652-691.

13. Geoffrion A. M. (1972), Generalized Bendres Decomposition, *J. Optimization Theory and Appl.* 10, p. 237-260.
14. Geoffrion A. M., Graves G. W. (1974), Multicommodity distribution systems design by Benders decomposition, *Management Science*, vol. 20, n° 5, p. 822-844.
15. Gilmore P. C., Gomory R. E. (1963), A Linear programming approach to the cutting-stock problem: part II, *Operations Research* 11, p. 863-888.
16. Gilmore P. C., Gomory R. E. (1965), Multistage cutting-stock problems in two or more dimensions, *Operations Research* 13, p. 94-120.
17. Gomory R. E., Hu T. C. (1962), An application of generalized linear programming to network flows, *Journal S.I.A.M.* 10, n° 2, p. 260-283.
18. Gondran M., Minoux M. (1979), *Graphes et algorithmes*, Eyrolles, Paris.
19. Grinold R. C. (1972), Steepest ascent for large scale linear probrams, *S.I.A.M. Review* 14, p. 447-464.
20. Held M., Wolfe P., Crowder H. P. (1974), Validation of subgradient optimization, *Mathematical Programming* 6, p. 62-88.
21. Himmelblau D. M. (1973), *Decomposition of Large Scale Problems*, North Holland, Amsterdam.
22. Huard P. (1980), Solving large scale linear programs without structure, in: *Large scale linear programming*, (G. B. Dantzig, M. A. Dempster, M. Kallio eds.), Proc. IIASA Workshop, 2-6 june 1980, appeared in: IIASA Collaborative Proceedings Series, Laxenburg (Austria), 1981.
23. Kelley J. E. (1960), The cutting plane method for solving convex programs, *Journal S.I.A.M.* 8, p. 703-712.
24. Kennington J. L. (1978), A Survey of Linear Cost Multicommodity Network Flows, *Operations Research* 26, p. 209-236.
25. Kennington J., Shalaby M. (1977), An effective subgradient procedure for minimal cost multicommodity flow problems, *Management Science*, vol. 23, n° 9, p. 994-1004.
26. Lasdon L. S. (1968), Duality and Decomposition in Mathematical Programming, *IEEE Trans. Syst. Sc. and Cybernetics* 4, p. 86-100.
27*. Lasdon L. S. (1970), *Optimization theory for large systems*, Macmillan Series in Operations Research, London.
28. Legendre J. P., Minoux M. (1977), Une application de la notion de dualité en programmation discrète : sélection et affectation optimales d'une flotte d'avions, *R.A.I.R.O.* 11, n° 2, p. 201-222.
29. Magnanti T. L., Shapiro J. F., Wagner M. H. (1976), Generalized Linear Programming Solves the Dual, *Management Science* 22, p. 1195-1203.
30. Mesarovic M. D., Macko D., Takahara Y. (1970), *Theory of multilevel hierarchical systems*, Academic Press, New York.
31. Minoux M. (1975), Résolution des problèmes de multiflots en nombres entiers dans les grands réseaux, *R.A.I.R.O.*, vol. 3, p. 21-40.
32. Minoux M. (1979), Optimum synthesis of a network with nonsimultaneous multicommodity flow requirements, Workshop *Applications of Graph theory and Combinatorics to Management*, *Applications of Graph theory and Combinatorics to Management*, Brussels, March 20-21, paru dans : *Studies on graphs and discrete programming*, (P. Hansen ed.), Annals of Discrete Mathematics 11, (1981), North Holland, p. 269-277.
33. Minoux M. (1981), Subgradient optimization and Bendes decomposition in large scale programming, *International Congress on Mathematical Programming*, Rio de Janeiro, Brazil, April 6-8, 1981.
34. Minoux M., Serreault J. Y. (1980), Subgradient optimization and large scale programming: application to multicommodity network synthesis with security constraints, *R.A.I.R.O.*, vol. 15, n° 2, p. 185-203.
35. Minoux M., Serreault J. Y. (1981), Synthèse optimale d'un réseau de Télécommunications avec contraintes de sécurité, *Annales des Télécommunications*, t. 36, n° 3-4, p. 211-230.
36. Polito J., McCarl B. A., Morin T. L. (1980), Solution of Spatial Equilibrium Problems with Benders Decomposition, *Management Science*, vol. 26, n° 6, p. 593-605.

37. Ritter K. (1967), A decomposition method for linear programming problems with coupling constraints and variables, *Mathematics Research Center, Report* 739, *University of Wisconsis.*
38. Rosen J. B. (1963), Convex partition programming, in: *Recent Advances in Mathematical Programming*, (R. L. Graves & P. Wolfe eds.), McGraw. Hill, New York, p. 159-176.
39. Rosen J. B. (1964), Primal parition programming for block-diagonal matrices, *Numerische Mathematik* 6, p. 250-260.
40. Sakarovitch M., Saigal R. (1967), An Extension of Generalized Upper Bounding Techniques for Structured Linear Programms, *S.I.A.M.* 15, p. 906-914.
41. Serreault J. Y., Minoux M. (1980), Le programme d'admissibilité avec contraintes de sécurité et coûts de mutation. Application aux réseaux mixtes, *Annales des Télécommunications*, t. 35, n° 1-2, p. 23-38.
42. Silverman G. J. (1968), Primal decomposition of mathematical programs by resource allocation, *Techn. Memorandum* 116, *Operations Research Dept, Case Western Reserve University.*
43. Tolla P. (1974), Méthodes de triangularisation et programmes d'ordinateur assurant l'amélioration de la précision des résultats en programmation linéaire, *Thèse* 3e *cycle*, Université Paris VI.
44. Watanabe N., Nishimura Y., Matsubara M. (1978), Decomposition in Large System Optimization Using The Method of Multipliers, *Journal Optimization Theory and Appl.* 25, n° 2, p. 181-193.
45. Wismer D. A. (1971), *Optimization Methods for Large Scale Systems*, McGraw Hill, New York.

ГЛАВА 9
ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Динамическое программирование — это не алгоритм. Речь идет скорее об общем принципе, допускающем приложения ко многим задачам оптимизации с ограничениями, линейным или нелинейным, с непрерывными или дискретными переменными, но обладающими некоторым свойством, называемым разложимостью (см. п. 2.1).

Термин «динамическое программирование» по происхождению связан с тем, что этот метод первоначально применялся к оптимизации динамических систем, т. е. систем, меняющихся в ходе времени, эволюция которых может управляться некоторыми переменными управления (см. п. 4.1). Однако в этой главе мы увидим, что принцип динамического программирования носит более общий характер и может применяться к задачам, в которых время не участвует, например к задачам целочисленной оптимизации (см. § 1 и пп. 4.2—4.4).

Хотя принцип оптимальности, лежащий в основе динамического программирования (см. п. 2.5), был, без всякого сомнения, представлен очень давно у столь известных авторов, как Ферма, Эйлер, Мопертюи, Маклорен (их цитирует Пойя [49]), а также недавно во Франции у П. Массе [36], по именно работы Р. Беллмана [2] подняли его на уровень общей теории для решения задач оптимизации.

Начиная с Р. Беллмана, динамическому программированию посвящено множество работ, среди которых мы укажем следующие: [25, 4, 23, 29, 46, 31, 12].

§ 1. Введение и примеры

В этом параграфе мы начнем с введения на характерном примере существенных идей динамического программирования, к которым мы вернемся в более общей форме в § 2.

1.1. Пример: решение задачи о рюкзаке с помощью динамического программирования. Рассмотрим классическую задачу, известную под названием «задачи о рюкзаке» («knapsack problem» в англоязычной литературе):

$$\sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_j x_j \leqslant b, \qquad (KP)$$

$$\forall j = 1, \ldots, n\colon\ x_j = 0 \text{ или } 1.$$

Предположим, что все коэффициенты a_j $(j = 1, \ldots, n)$ целые и что b целое. С другой стороны, предположим, что $a_j \geqslant 0$ $(\forall j)$ (всегда можно ограничиться именно этим случаем: если $a_j < 0$ для индекса j, то достаточно осуществить замену переменных $x_j = 1 - x'_j$). Следовательно, правая часть b должна быть неотрицательным целым. Коэффициенты c_j $(j = 1, \ldots, n)$ могут быть любыми вещественными. Обозначим через F^* оптимальное значение задачи (KP).

Основная идея динамического программирования состоит в попытке свести решение задачи (KP), содержащей n переменных, к решению ряда более простых задач оптимизации, например одномерных задач оптимизации.

Для этого нужно осуществить две операции.

1°) Нужно вложить предложенную задачу в семейство задач той же природы.

В случае рассматриваемой нами задачи (KP) можно взять, например (это не единственная возможность), семейство $n(b+1)$ задач

$$F_i(E) = \max \sum_{j=i}^{n} c_j x_j,$$

$$\sum_{j=1}^{n} a_j x_j \leqslant b - E, \qquad KP_i(E)$$

$$x_j = 0 \quad \text{или} \quad 1, \qquad j = i, \ldots, n,$$

где i меняется от 1 до n, а E — целое, меняющееся от 0 до b.

Величина E, параметризующая в этом примере семейство задач, носит классическое название *переменной состояния* (если эта величина — векторная, как в п. 1.3, то говорят о *векторе состояния*). Заметим — и это замечание носит общий характер, как мы увидим в пп. 2.3 и 2.4,— что речь идет о величине, связанной с ограничениями задачи. Множество $\mathscr{E} = \{0, 1, \ldots, b\}$ возможных значений величины E называется *пространством состояний*, связанным с данной задачей.

Исходная задача (KP) очевидным образом входит в рассматриваемое семейство (если взять $i = 1$ и $E = 0$).

Следовательно, имеем $F^* = F_1(0)$.

С другой стороны, $\forall i = 1, \ldots, n$ договоримся считать, что $F_i(E) = -\infty$ для $E < 0$, а также если $E \geqslant 0$, но задача $KP_i(E)$ не имеет решения (в частности, это так для $E > b$).

2°) Нужно найти возвратное соотношение, связывающее между собой оптимальные значения этих различных задач.

В случае семейства $KP_i(E)$ легко получаем искомое возвратное соотношение:

$$F_i(E) = \max_{x_i = 0 \text{ или } 1} \{c_i x_i + F_{i+1}(E + a_i x_i)\} =$$

$$= \max \{F_{i+1}(E);\ c_i + F_{i+1}(E + a_i)\}.$$

Это соотношение является просто выражением того факта, что в оптимальном решении $(x_i, x_{i+1}, \ldots, x_n)$ задачи $KP_i(E)$ должно

быть либо $x_i = 0$, либо $x_i = 1$, и в каждом из этих случаев набор $(x_{i+1}, \ldots, x_n)$ должен быть оптимальным решением задачи, ограниченной переменными $i+1, \ldots, n$.

Таким образом, для любого значения E (где $0 \leqslant E \leqslant b$) определение величины $F_i(E)$ по известным значениям $F_{i+1}(E')$ (для $0 \leqslant E' \leqslant b$) сводится к простейшей задаче оптимизации (сравнению двух чисел).

Полученное возвратное соотношение позволяет тогда решить предложенную задачу с помощью алгоритма за n шагов.

Алгоритм 1. (Решение задачи о рюкзаке с помощью динамического программирования.)

а) Инициализация (шаг n):

$\forall E$ $(0 \leqslant E \leqslant b)$ вычислить $F_n(E)$ следующим образом:

$$F_n(E) = 0 \quad \text{для } E > b - a_n,$$
$$F_n(E) = c_n \quad \text{для } E \leqslant b - a_n \quad \text{и } c_n \geqslant 0,$$
$$F_n(E) = 0 \quad \text{для } E \leqslant b - a_n \quad \text{и } c_n < 0.$$

b) Для $i = n-1, n-2, \ldots, 2$ последовательно вычислять (шаг i):

$\forall E (0 \leqslant E \leqslant b)$:

$$F_i(E) = \max \{F_{i+1}(E);\ c_i + F_{i+1}(E + a_i)\}.$$

c) Шаг 1:

Вычислить F^* следующим образом:

$$F^* = F_1(0) = \max \{F_2(0);\ c_1 + F_2(a_1)\}.$$

Заметим, что предыдущий алгоритм позволяет не только решить предложенную задачу (KP), но также — посредством очень небольших дополнительных вычислений — и все задачи вида $KP_1(E)$ для $E = 0, 1, \ldots, b$, иначе говоря, все семейство задач, отличающихся от (KP) только значением правой части ограничения.

Для этого было бы достаточно заменить шаг 1 на

c) Шаг 1′:

Вычислить для $E (0 \leqslant E \leqslant b)$:

$$F_1(E) = \max \{F_2(E);\ c_1 + F_2(E + a_1)\}.$$

1.2. Определение оптимального решения x^*. Алгоритм 1 позволяет определить оптимальное значение F^* задачи (KP), но не приводит явным образом к соответствующему вектору x^*.

Первая идея получить вектор x^* состоит в том, чтобы просто определять на каждом шаге алгоритма $i = 1, \ldots, n-1$ и для любого E $(0 \leqslant E \leqslant b)$ величины $\bar{x}_i(E)$ формулой

$$\bar{x}_i(E) = 0, \quad F_i(E) = F_{i+1}(E),$$
$$\bar{x}_i(E) = 1, \quad F_i(E) = c_i + F_{i+1}(E + a_i),$$

а на шаге n — формулой

$$\bar{x}_n(E) = 0, \quad F_n(E) = 0,$$
$$\bar{x}_n(E) = 1, \quad F_n(E) = c_n.$$

Тогда оптимальное решение x^* можно получить в конце работы алгоритма 1 с помощью следующей процедуры.

Процедура 1.

a) $E = 0$.

b) Для $i = 1, \ldots, n$ последовательно выполнять:

$$x_i^* \leftarrow \bar{x}_i(E),$$

$$E \leftarrow E + a_i x_i^*.$$

Мы видим, что этот метод нуждается в запоминании некоторого существенного объема информации, а именно $n(b+1)$ величин $\bar{x}_i(E)$.

В случае задач о рюкзаке с неограниченными переменными в [20] предложена процедура, требующая объема информации порядка b.

1.3. Анализ сложности и границы динамического программирования. Для изучения эффективности некоторого алгоритма динамического программирования представляет интерес оценка *сложности* этого алгоритма, т. е. числа необходимых элементарных операций (сложений, сравнений), а также объема памяти, требуемого при запуске этого алгоритма на вычислительной машине.

В алгоритме 1 на каждом шаге и для каждого значения E $(0 \leqslant E \leqslant b)$ требуется одно сложение и одно сравнение.

Так как у нас n шагов, то число элементарных операций растет как $n(b+1)$. Говорят, что его сложность есть $O(nb)$.

С точки зрения загромождения памяти алгоритм 1 требует в каждый момент не более $2(b+1)$ мест в памяти $(b+1)$ значений $F_{i+1}(E)$ и $b+1$ значений $F_i(E)$.

Если использовать процедуру 1 для явного определения x^*, то объем требуемой памяти оказывается, таким образом, равным $(n+2)(b+1)$.

Выводы, которые можно извлечь из предыдущего анализа задачи о рюкзаке, носят общий характер: сложность вычислений, как и загромождение памяти алгоритмом динамического программирования, непосредственно зависят от числа значений E (переменного или вектора состояния), которые мы рассматриваем, т. е. от мощности пространства состояний, связанного с данной задачей.

Следовательно, по существу именно этот параметр и ставит самые строгие границы применению динамического программирования.

Рассмотрим, например, задачу о «многомерном рюкзаке», которая является прямым расширением предыдущей задачи и которая может быть записана в матричной форме:

$$cx = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$Ax \leqslant b, \qquad (MKP)$$

$$x_j = 0 \text{ или } 1, \qquad j = 1, \ldots, n,$$

где A — матрица $m \times n$ с неотрицательными целыми коэффициентами, x — n-мерный вектор размерности n, b — вектор с целыми положительными компонентами размерности m.

В таком виде могут быть представлены многие задачи целочисленного линейного программирования (см., например, [22], гл. 10).

С теоретической точки зрения все предыдущее немедленно обобщается на эту задачу.

Достаточно рассмотреть семейство задач

$$F_i(E) = \max \sum_{j=i}^{n} c_j x_j,$$

$$\sum_{j=i}^{n} A_j x_j \leqslant b - E, \qquad MKP_i(E)$$

$$x_j = 0 \text{ или } 1, \qquad j = i, \ldots, n,$$

для i, меняющегося от 1 до n, где E есть m-мерный вектор состояния $E = (E_1, \ldots, E_m)$, каждая компонента которого E_k может принимать $b_k + 1$ значений: 0, 1, 2, ..., b_k (A_j означает j-й столбец матрицы A).

Возвратное соотношение между значениями $F_i(E)$ имеет вид

$$F_i(E) = \max \{F_{i+1}(E);\ c_i + F_{i+1}(E + A_i)\},$$

и отсюда следует, что алгоритм 1 может быть применен к этой задаче, если просто заменить скаляр a_i на m-вектор A_i (i-й столбец матрицы A) и рассматривать E и b как m-векторы. Однако мы видим, что практическое применение динамического программирования остается ограниченным задачами, в которых число m ограничений очень невелико (для ясности, $m < 10$). Действительно, в этом случае мощность пространства состояний равна

$$|\mathscr{E}| = (b_1 + 1) \ldots (b_m + 1).$$

Например, для $m = 50$ и $b_k = 1$ ($\forall k = 1, \ldots, m$) получаем значение $|\mathscr{E}| \simeq 10^{15}$, что намного превосходит возможности наиболее мощных из имеющихся в нашем распоряжении вычислительных машин.

Это наблюдение, касающееся задачи о многомерном рюкзаке, носит совершенно общий характер: во многих случаях число состояний растет экспоненциально с ростом числа ограничений, т. е. с ростом размерности вектора состояния.

В § 3 мы изучим некоторые технические приемы, позволяющие решать задачи, для которых число состояний априори очень велико.

1.4. Динамическое программирование и задачи о кратчайших (длиннейших) путях в графах. Интересно интерпретировать алгоритм 1 из п. 1.1 для задачи о рюкзаке как поиск длиннейшего пути в графе $G = [X, U]$, множество X вершин которого и множество U дуг которого строятся следующим образом.

Множество X состоит из:

— особой вершины, которую мы обозначим (0, 0) (начальная вершина);

— $n(b+1)$ вершин, соответствующих каждой паре (E, i), где $0 \leqslant E \leqslant b$ и $1 \leqslant i \leqslant n$.

С другой стороны, каждой паре вершин вида $(0, 0)$, $(E, 1)$ мы сопоставляем:

— дугу $[(0, 0), (E, 1)]$ длины 0, если $E = 0$;

— дугу $[(0, 0), (E, 1)]$ длины c_1, если $E = a_1$ и $0 \leqslant a_1 \leqslant b$.

Наконец, для $i = 1, 2, \ldots, n-1$ каждой паре вершин вида $\{(E, i), (E', i)\}$ мы сопоставляем:

— дугу $[(E, i), (E', i+1)]$ длины 0, если $E' = E$;

— дугу $[(E, i), (E', i+1)]$ длины c_{i+1}, если $E' = E + a_{i+1}$ и $0 \leqslant E + a_{i+1} \leqslant b$;

подмножество X_t вершин вида (E, n) для $0 \leqslant E \leqslant b$ называется множеством *конечных вершин* (состояний).

Пример этого построения для задачи

$$2x_1 + 3x_2 + 6x_3 \to \max,$$

$$3x_1 + 2x_2 + 4x_3 \leqslant 5,$$

$$x_1, x_2, x_3 = 0 \text{ или } 1$$

дан на рис. 1 (для упрощения представления мы изображаем только дуги, принадлежащие пути с началом $(0, 0)$).

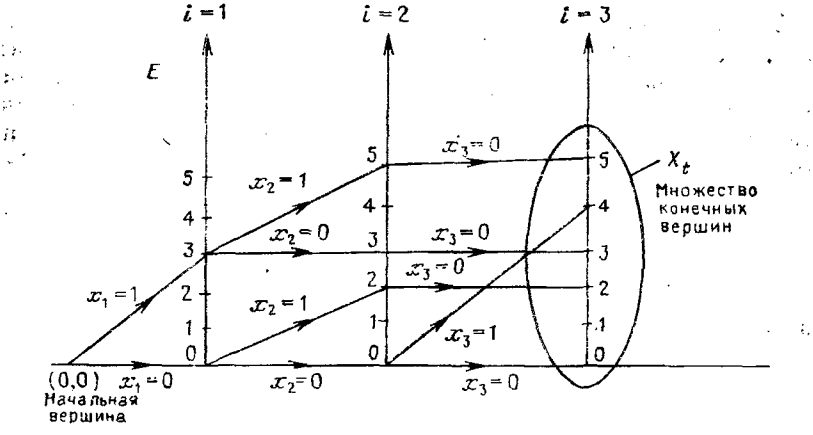


Рис. 1. Пример секвенциального графа, соответствующего решению задачи о рюкзаке с помощью динамического программирования (в этом примере $n = 3$, $b = 5$)

Такие графы, как изображенный на рис. 1,— в которых множество вершин разбито на $n+1$ подмножеств $0, 1, \ldots, n$ и в которых не существует дуг, которые бы соединяли элементы не следующих друг за другом подмножеств,— называются *секвенциальными графами* (следует заметить, что секвенциальный граф необходимо не имеет замкнутых путей).

Задача о рюкзаке (KP) эквивалентна тогда поиску пути на графе G с максимальной длиной между вершиной $(0, 0)$ и множеством

X_t вершин вида (E, n) для $E = 0, 1, \ldots, b$ (множеством конечных вершин).

Действительно, каждый путь с началом в $(0, 0)$ и концом в X_t соответствует решению задачи (KP), и длина этого пути есть не что иное, как стоимость связанного с ним решения.

Алгоритм 1 может быть тогда интерпретирован как алгоритм наиболее длинного пути, рекурсивно определяющий множества наиболее длинных путей, концы которых лежат в X_t, а мощности (число дуг) равны последовательно 1, 2, 3, ... Этот алгоритм известен в теории графов под названием *алгоритма Беллмана* (см. [3, 22], гл. 2).

Заметим наконец, что тот же граф G позволяет представить все семейство задач $KP_1(E)$ для всех $0 \leqslant E \leqslant b$. Действительно, для $0 \leqslant E \leqslant b$ достаточно выбрать в качестве нового множества конечных вершин множество вершин вида (E', n) с $0 \leqslant E' \leqslant b - E$.

Более общим образом мы покажем в п. 2.7, что задача динамического программирования с конечным числом состояний всегда может рассматриваться как задача о кратчайшем (длиннейшем) пути в секвенциальном графе.

§ 2. Теоретические основания динамического программирования

2.1. Теорема оптимальности. Случай без ограничений. Как мы видели на примере в § 1, основная идея динамического программирования состоит в попытке заменить оптимизацию функции n переменных решением некоторого числа задач оптимизации, которые решаются проще, например задач оптимизации с одним переменным. В некотором смысле этот метод может рассматриваться как метод разложения.

Чтобы пояснить эту идею, рассмотрим задачу оптимизации без ограничений:

$$f(x, y_1, \ldots, y_k) \to \min,$$

где f — вещественная функция вещественного переменного x и k вещественных переменных $y_1, \ldots, y_k$ (каждое из переменных может принимать произвольные вещественные значения).

Предположим тогда, что функция f *разделяема* на две функции f_1 и f_2, т. е. допускает представление в виде

$$f(x, y) = f_1(x, f_2(y)),$$

где y обозначает k-вектор с компонентами $y_1, \ldots, y_k$.

Есть ли возможность свести исходную задачу с $k+1$ переменными к задачам оптимизации, содержащим меньшее число переменных? Иначе говоря, нужно выяснить, когда можно записать

$$\min_{(x,y)} f(x, y) = \min_{x} \left\{ f_1 \left(x, \min_{y} \{ f_2(y) \} \right) \right\}. \tag{1}$$

Достаточное (но тем не менее весьма общее) условие того, что соотношение (1) справедливо, дано Миттеном [38] и включает понятие разложимой функции.

Определение 1. Функция f называется *разложимой* на f_1 и f_2, если f разделяема ($f(x, y)=f_1(x, f_2(y))$ и если, кроме того, функция f_1 монотонно не убывает по отношению к своему второму аргументу.

Тогда можно сформулировать следующий фундаментальный результат.

Теорема 1 («теорема оптимальности»). *Пусть f — вещественная функция от x и $y=(y_1, \ldots, y_k)$. Если f разложима и $f(x, y)=f_1(x, f_2(y))$, то тогда*

$$\operatorname*{Opt}_{(x,y)} f(x, y) = \operatorname*{Opt}_{x} \left\{ f_1 \left(x, \operatorname*{Opt}_{y} \{ f_2(y) \} \right) \right\}$$

(Opt = min *или* max).

Доказательство. Возьмем для определенности Opt = min. По определению минимума для $f(x, y)=f_1(x, f_2(y))$ можем записать

$$\min_{(x,y)} f_1(x, f_2(y)) \leqslant f_1(x_0, f_2(y_0)) \qquad \forall x_0, y_0.$$

В частности, беря $y_0=\bar{y}$, где $\bar{y}$ определяется формулой $f_2(\bar{y}) = \min_y \{f_2(y)\}$, получаем

$$\min_{(x,y)} f_1(x, f_2(y)) \leqslant f_1\left(x_0, f_2(\bar{y})\right) \qquad \forall x_0.$$

Это неравенство остается справедливым, если заменить x_0 значением, которое делает правую часть минимальной. Отсюда следует, что

$$\min_{(x,y)} f_1(x, f_2(y)) \leqslant \min_{x} \left\{ f_1 \left(x, \min_{y} \{ f_2(y) \} \right) \right\}.$$

Покажем, что это неравенство справедливо также и в противоположном смысле.

Так как функция f разложима, то функция f_1 монотонно не убывает по своему второму аргументу. Следовательно, можно записать

$$f_1\left(x_0, \min_y \{f_2(y)\}\right) \leqslant \min_y f_1(x_0, f_2(y)) \qquad \forall x_0, \forall y_0.$$

Это соотношение выполняется, в частности, если взять в качестве y_0 значение, которое делает правую часть минимальной. Следовательно,

$$f_1\left(x_0, \min_y \{f_2(y)\}\right) \leqslant \min_y f_1(x_0, f_2(y)) \quad \forall x_0.$$

Беря теперь в качестве x_0 значение $\bar{x}_0$, которое делает правую часть минимальной по x, видим, что

$$\min_x \left\{ f_1 \left(x, \min_y \{ f_2(y) \} \right) \right\} \leqslant f_1\left(\bar{x}_0, \min_y \{f_2(y)\}\right) \leqslant \min_{(x,y)} f_1(x, f_2(y)),$$

что завершает доказательство.

Случай $\mathrm{Opt} = \max$ рассматривается совершенно так же.

2.2. Распространение теоремы оптимальности на случай с ограничениями. Теорема оптимальности немедленно распространяется на случай (более интересный на практике) оптимизации при наличии ограничений.

Рассмотрим задачу

$$\mathrm{Opt}\, f(x,\ y),$$
$$(x,\ y) \in \Omega,$$

где $\mathrm{Opt} = \max$ или $\min$ и где Ω — множество решений задачи ($\Omega \subset \subset \mathbf{R}^{n+1}$, $\Omega \neq \varnothing$).

Для любого вещественного x введем обозначение

$$\Omega_x = \{y \mid y \in \mathbf{R}^n;\ (x,\ y) \in \Omega\}$$

(заметим, что для некоторых значений x может случиться, что $\Omega_x = \varnothing$).

Тогда введем следующие соглашения:

$$\operatorname*{Opt}_{y \in \Omega_x} \{f_2(y)\} = +\infty,\ \Omega_x = \varnothing,\ \mathrm{Opt} = \min,$$

$$\operatorname*{Opt}_{y \in \Omega_x} \{f_2(y)\} = -\infty,\ \Omega_x = \varnothing,\ \mathrm{Opt} = \max$$

и, с другой стороны, $\forall x$

$$f_1(x, +\infty) = +\infty,$$
$$f_1(x, -\infty) = -\infty.$$

Теперь мы можем сформулировать теорему.

Теорема 1′ (теорема оптимальности: случай с ограничениями). *Если f разложима и $f(x, y) = f_1(x, f_2(y))$, то*

$$\operatorname*{Opt}_{(x,y) \in \Omega} \{f(x, y)\} = \operatorname*{Opt}_{x} \left\{f_1\left(x,\ \operatorname*{Opt}_{y \in \Omega_x} \{f_2(y)\}\right)\right\}. \tag{2}$$

Доказательство. Шаг за шагом аналогично доказательству теоремы 1.

2.3. Понятие состояния в динамическом программировании. Вернемся к примеру предыдущего пункта, в котором отыскивалось решение задачи

$$\mathrm{Opt}\, f(x,\ y),$$
$$(x,\ y) \in \Omega,$$

где f разложима и допускает представление в виде

$$f(x,\ y) = f_1(x,\ f_2(y)).$$

С практической точки зрения (если предположить, что множество решений Ω априори произвольно) поиск оптимума функции f на Ω по формуле разложения (2)

$$\operatorname*{Opt}_{x} \left\{f_1\left(x,\ \operatorname*{Opt}_{y \in \Omega_x} \{f_2(y)\}\right)\right\}$$

эквивалентен

— определению для каждого x значения функции φ:

$$\varphi(x) = \operatorname*{Opt}_{y \in \Omega_x} \{f_2(y)\};$$

— затем определению оптимума по x функции

$$\psi(x) = f_1(x, \varphi(x)).$$

Заметим тогда, что это равносильно вычислению минимума $f(x, y)$ «перечислением» всех $(x, y) \in \Omega$. Формула (2) представляет собой не что иное, как уточнение порядка, в котором осуществляется «перечисление» элементов множества Ω.

Ясно, что такая процедура не может привести к практически эффективному методу решения.

На самом деле, как мы сейчас увидим, ценность теоремы оптимальности (для построения алгоритмов разложения) существенно зависит от вида области решений Ω, более точно, от возможности рассматривать эту область как элемент более широкого семейства областей $\Omega_t(E)$, параметризуемых некоторым вектором E — так называемым *вектором состояния*.

Рассмотрим в полной общности задачу оптимизации при наличии ограничений с переменными (x, y) в виде

$$\begin{aligned} F^* &= \operatorname{Opt} f(x, y), \\ g(x, y) &\in \mathscr{E}_t, \end{aligned} \qquad (\text{п})$$

где g: $\mathbf{R}^{k+1} \to \mathbf{R}^m$ и $\mathscr{E}_t \subset \mathbf{R}^m$. (Если $\mathscr{E}_t$ сводится к одноточечному множеству, то получаем задачу с ограничениями в виде равенств.)

Предположим тогда, что функция g *разделяема* и допускает представление в виде

$$g(x, y) = h_2(y, h_1(x)),$$

где

$$h_1(x)\colon \mathbf{R} \to \mathbf{R}^m,$$
$$h_2(y, z)\colon \mathbf{R}^{k+m} \to \mathbf{R}^m.$$

Мы видим, что для данных значений x и y функция $g(x, y)$ может быть вычислена с помощью действий

$$\begin{aligned} E_1 &= h_1(x), \\ E_2 &= h_2(y, E_1), \\ g(x, y) &= E_2. \end{aligned}$$

Пусть $\mathscr{E}$ — такое подмножество $\mathbf{R}^m$, что

$$\forall x \in \{x \mid \exists y\colon (x, y) \in \Omega\} \Rightarrow h_1(x) \in \mathscr{E},$$
$$\forall E \in \mathscr{E},\ \forall y \in \{y \mid \exists x\colon (x, y) \in \Omega\} \Rightarrow h_2(y, E) \in \mathscr{E}.$$

Тогда множество $\mathscr{E}$ будет называться *множеством состояний*, а элемент $E \in \mathscr{E}$ — *вектором состояния*. Функции h_1 и h_2 будут называться *функциями перехода*. Множество $\mathscr{E}_t$ есть множество допустимых *конечных* состояний (эта терминология заимствована из области оптимизации динамических систем (см. п. 4.1) и мар-

ковских процессов (см. п. 4.5), которые, как мы уже говорили, были первыми областями применения динамического программирования).

Тогда идея состоит в том, чтобы заменить решение задачи (π), содержащей $k+1$ переменных, решением семейства задач оптимизации с k переменными:

$$F_2(E_1) = \operatorname{Opt}\{f_2(y)\}, \quad h_2(y, E) \in \mathscr{E}_1 \qquad \pi_2(E_1)$$

для E_1, пробегающего множество $\mathscr{E}$.

Введем обозначение

$$\Omega_2(E_1) = \{y \mid h_2(y, E_1) \in \mathscr{E}_1\}$$

для множества решений задачи $\pi_2(E_1)$.

Тогда можно заметить, что имеет место равенство

$$\Omega_x = \Omega_2(h_1(x)),$$

где следует напомнить, что Ω_x при каждом фиксированном есть обозначение множества $\{y \mid (x, y) \in \Omega\}$.

Договоримся считать, что

$$F_2(h_1(x)) = +\infty \quad \text{(в случае минимизации)},$$
$$F_2(h_1(x)) = -\infty \quad \text{(в случае максимизации)},$$

если $\Omega_x = \Omega_2(h_1(x))$ пусто.

Так как мы предполагали, что f разложима, то формулу (2) можно переписать в виде

$$F^* = \operatorname*{Opt}_x \left\{ f_1 \left(x, \operatorname*{Opt}_{y \in \Omega_2(h_1(x))} \{f_2(y)\} \right) \right\}.$$

Но по определению

$$\operatorname*{Opt}_{y \in \Omega_2(h_1(x))} \{f_2(y)\} = F_2(h_1(x)).$$

Следовательно, если предположить, что все задачи $\pi_2(E_1)$ для всех $E_1 \in \mathscr{E}$ решены, то можно решить задачу (π) как задачу оптимизации с единственным переменным x по формуле

$$F^* = \operatorname*{Opt}_x \{f_1(x, F_2(h_1(x))\}. \tag{3}$$

Сейчас мы увидим, что если применить этот результат рекурсивно, то задача оптимизации с n переменными может быть решена за n шагов, причем каждый шаг состоит в решении некоторого количества задач оптимизации с одним переменным.

2.4. Функциональное уравнение динамического программирования. Рассмотрим задачу оптимизации с n переменными

$$F^* = \operatorname{Opt} f(x_1, \ldots, x_n), \quad g(x_1, \ldots, x_n) \in \mathscr{E}_t \subset \mathbf{R}^m. \qquad (\pi)$$

Предположим, что функция f разложима в виде

$$f(x_1, \ldots, x_n) = f_1(x_1, \bar{f}_2(x_2, \ldots, x_n)),$$
$$\bar{f}_2(x_2, \ldots, x_n) = f_2(x_2, \bar{f}_3(x_3, \ldots, x_n)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\bar{f}_n(x_n) = f_n(x_n).$$

Предположим, кроме того, что существует такое подмножество $\mathscr{E} \subset \mathbb{R}^m$ (пространство состояний), что g может быть рекурсивно вычислена для любого множества значений $x_1, \ldots, x_n$ по формулам

$$E_1 = h_1(x_1) \in \mathscr{E},$$
$$E_2 = h_2(x_2, E_1) \in \mathscr{E},$$
$$E_3 = h_3(x_3, E_2) \in \mathscr{E},$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$E_n = h_n(x_n, E_{n-1}) \in \mathscr{E},$$
$$g(x_1, \ldots, x_n) = E_n,$$

где функции $h_1, \ldots, h_n$ называются *функциями перехода*.

Обозначим для любого $i = 2, 3, \ldots, n$ через $\bar{h}_i(x_i, x_{i+1}, \ldots, x_n, E)$ функцию, рекурсивно определенную равенствами

$$E_i = h_i(x_i, E),$$
$$E_{i+1} = h_{i+1}(x_{i+1}, E_i),$$
$$\ldots\ldots\ldots\ldots$$
$$E_n = h_n(x_n, E_{n-1}),$$
$$\bar{h}_i(x_i, x_{i+1}, \ldots, x_n, E) = E_n.$$

Заметим, что $\forall i$ имеем

$$\bar{h}_i(x_i, \ldots, x_n, E) = \bar{h}_{i+1}(x_{i+1}, \ldots, x_n, h_i(x_i, E)).$$

Рассмотрим тогда задачу (π), которую можно представить в виде

$$F^* = \operatorname{Opt} f_1(x_1, \bar{f}_2(x_2, \ldots, x_n)),$$
$$\bar{h}_2(x_2, \ldots, x_n, h_1(x_1)) \in \mathscr{E}_1.$$

Отождествляя x с x_1, а y с $(x_2, \ldots, x_n)$, мы можем применить в этой задаче результаты п. 2.3 (формулу (3)). Мы получим соотношение

$$F^* = \operatorname*{Opt}_{x_1} \{f_1(x_1, F_2(h_1(x_1))\},$$

где для любого $E_1 \in \mathscr{E}$ величина $F_2(E_1)$ есть оптимальное значение задачи

$$F_2(E_1) = \operatorname{Opt} \bar{f}_2(x_2, \ldots, x_n),$$
$$\bar{h}_2(x_2, \ldots, x_n, E_1) \in \mathscr{E}_1. \qquad \pi_2(E_1)$$

Рассмотрим теперь задачу $\pi_2(E_1)$ для фиксированного E_1. Эта задача сама по себе представляется в виде

$$F_2(E_1) = \operatorname{Opt} f_2(x_2, \bar{f}_3(x_3, \ldots, x_n)),$$
$$\bar{h}_3(x_3, \ldots, x_n, h_2(x_2, E_1)) \in \mathscr{E}_t.$$

Отождествляя x с x_2 и y с $(x_3, \ldots, x_n)$ и применяя результаты п. 2.3 (формулу (3)), получаем соотношение

$$F_2(E_1) = \operatorname*{Opt}_{x_2} \{f_2(x_2, F_3(h_2(x_2, E_1)))\},$$

где для любого $E_2 \in \mathscr{E}$ величина $F_3(E_2)$ есть оптимальное значение задачи

$$\pi_3(E_2) \quad \begin{array}{l} F_3(E_2) = \operatorname{Opt} \bar{f}_3(x_3, \ldots, x_n), \\ \bar{h}_3(x_3, \ldots, x_n) \in \mathscr{E}_t. \end{array}$$

Продолжая таким образом, видим, что значения $F_i(E_{i-1})$ (для $2 \leqslant i \leqslant n-1$) могут быть для любого $E_{i-1} \in \mathscr{E}$ определены равенством

$$F_i(E_{i-1}) = \operatorname*{Opt}_{x_i} \{f_i(x_i, F_{i+1}(h_i(x_i, E_{i-1})))\}, \tag{4}$$

а для $i = n$ — равенством

$$F_n(E_{n-1}) = \operatorname*{Opt}_{\substack{x_n \\ h_n(x_n, E_{n-1}) \in \mathscr{E}_t}} \{f_n(x_n)\}, \tag{5}$$

где для любого i $(2 \leqslant i \leqslant n)$ через $F_i(E)$ обозначено оптимальное значение задачи

$$\pi_i(E) \quad \begin{array}{l} \operatorname{Opt} \bar{f}_i(x_i, x_{i+1}, \ldots, x_n), \\ \bar{h}_i(x_i, x_{i+1}, \ldots, x_n, E) \in \mathscr{E}_t. \end{array}$$

Тогда мы видим, что оказывается возможным по рекуррентности определить шаг за шагом функции F_i для $i = n, n-1, \ldots, 2$, а затем, исходя из F_2, определить оптимальное значение F^* задачи (π).

Уравнение (4) называется *функциональным уравнением динамического программирования.*

Его решение по рекуррентности, исходя из условий (5), приводит к алгоритмической процедуре со следующими n шагами.

Алгоритм 2 (процедура «с конца»).

а) Инициализация (шаг n):

$\forall E \in \mathscr{E}$ вычислить: $F_n(E) = \operatorname*{Opt}_{\substack{x_n \\ h_n(x_n, E) \in \mathscr{E}_t}} \{f_n(x_n)\}$ (если не существует такого x_n, что $h_n(x_n, E) \in \mathscr{E}_t$, то $F_n(E) = +\infty$ для $\operatorname{Opt} = \min$, $F_n(E) = -\infty$ для $\operatorname{Opt} = \max$).

b) Для $i = n-1, n-2, \ldots, 2$ последовательно вычислить (шаг i)

$$F_i(E) = \operatorname*{Opt}_{x_i} \{f_i(x_i, F_{i+1}(h_i(x_i, E)))\} \qquad \forall E \in \mathscr{E}.$$

c) Шаг 1:
Вычислить F^* по формуле

$$F^* = \operatorname*{Opt}_{x_1} \{f_1(x_1, F_2(h_1(x_1)))\}.$$

Этот алгоритм, представляющий собой не что иное, как вычисление по рекуррентности, осуществляемое в порядке убывания индексов переменных, называется процедурой «с конца» в противоположность процедуре «с начала», которая будет изложена в п. 2.8.

Таким образом, как и в примере из п. 1.1, общая методология динамического программирования состоит в следующем:

— вложить предложенную задачу в семейство задач той же природы;

— связать возвратным соотношением оптимальные значения этих задач;

— решить полученное возвратное уравнение (функциональное уравнение) для определения оптимума поставленной задачи.

Так как число элементарных подзадач оптимизации (с одним переменным), подлежащих решению, растет как произведение $n \times p$ (где p — число состояний, которые должны рассматриваться на каждом этапе), то алгоритм, подобный алгоритму 2, удается применять лишь в ситуациях, удовлетворяющих довольно жестким требованиям, например:

— если множество $\mathscr{E}$ состояний конечно и имеет достаточно малую мощность (тогда речь идет о динамическом программировании с конечным числом состояний, см. п. 2.7);

— если множество $\mathscr{E}$ есть подмножество $\mathbf{R}^m$, содержащее бесконечно много элементов (например, компактное подмножество в $\mathbf{R}^m$), но m (размерность векторов состояния) достаточно мала, так что есть возможность приблизить функции $F_i(E)$ на $\mathscr{E}$ дискретизацией в конечном (и по возможности не слишком большом) числе точек $E^1, \ldots, E^q$. Тогда вычисление $F_i(E)$ в других состояниях E проводится с помощью линейной или полиномиальной интерполяции (по поводу детального обсуждения этих технических приемов в контексте динамического программирования см., например, [4], гл. 12).

В других случаях для улучшения эффективности алгоритмов динамического программирования должны применяться специальные приемы (см. § 3).

2.5. **Принцип оптимальности.** Алгоритм 2, законность которого вытекает из теоремы оптимальности, может быть интерпретирован как построение оптимального решения задачи (л) с помощью оптимальных частных решений.

Таким образом, среди всех возможных решений $(x_i, x_{i+1}, \ldots, x_n)$, осуществляющих переход из состояния E (на шаге $i-1$) в подмножество состояний $\mathscr{E}_i$ (на шаге n) (иначе говоря, таких решений, что $h_i(x_i, x_{i+1}, \ldots, x_n, E) \in \mathscr{E}_i$), мы сохраняем только информацию об оптимальном решении (со значением $F_i(E)$): все

остальные решения оказываются, таким образом, «забытыми» в дальнейших шагах решения.

Тогда полученное оптимальное решение согласуется с *принципом оптимальности*, который был сформулирован Р. Беллманом [2]: оптимальная политика обладает тем свойством, что, каковы бы ни были начальное состояние и начальное решение, оставшиеся решения должны образовывать оптимальную политику, отвечающую состоянию, возникающему в результате первого решения.

Этот принцип можно подытожить следующим образом: «Любое оптимальное решение может быть образовано только оптимальными частичными решениями».

Однако, как было замечено в [50] и [41], не всегда имеет место эквивалентность между принципом оптимальности и функциональным уравнением динамического программирования (4).

Действительно, условия, позволяющие установить справедливость уравнения (4) (особенно существенна гипотеза разложимости функции стоимости, т. е. сепарабельность + монотонность, см. п. 2.1), недостаточны для установления принципа оптимальности.

Пример, предложенный в [41], позволяет пояснить это обстоятельство. Рассмотрим граф на рис. 2, в котором ищется путь с минимальной стоимостью между вершинами 1 и 4, причем стоимость

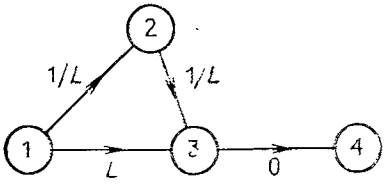


Рис. 2. Пример оптимального решения (путь {1, 3, 4}), не удовлетворяющего ни функциональному уравнению динамического программирования, ни принципу оптимальности (подпуть {1, 3} не является путем между 1 и 3 с минимальной стоимостью)

пути равна произведению чисел, связанных с каждой из дуг (L — положительная постоянная, которая выбирается достаточно большой). Для этой функции стоимости выполнены предположения разделяемости и монотонности, и при этом путь {1, 3, 4}, являющийся оптимальным (со стоимостью 0) между вершинами 1 и 4, содержит частичный путь {1, 3}, который не является оптимальным между вершинами 1 и 3.

Чтобы удостовериться в справедливости принципа оптимальности, нужно заменить условие монотонности в широком смысле, входящее в доказательство теоремы оптимальности (теорема 1 п. 2.1), предположением строгой монотонности.

Мы будем говорить, что функция f *строго разложима*, если f разделяема в виде

$$f(x, y) = f_1(x, f_2(y))$$

и если f_1 — строго монотонная функция по своему второму аргументу, т. е.

$$\forall x:\ z_1 < z_2 \Rightarrow (x, z_1) < f_1(x, z_2),$$
$$z_1 = z_2 \Rightarrow f_1(x, z_1) = f_1(x, z_2).$$

Тогда можно показать [41], что если выполнено предположение строгой разложимости, то любое оптимальное решение удовлетворяет принципу оптимальности и соответствует поэтому решению функционального уравнения (4).

Кроме того, можно показать, что при этих условиях алгоритм 2 (п. 2.4) позволяет определить все оптимальные решения рассматриваемой задачи (это все решения, дающие оптимальные значения величинам $F_i(E)$ — решениям функционального уравнения).

2.6. Примеры разложимых функций. Основной общий класс разложимых функций, используемый на практике, образован функциями вида

$$f(x_1, \ldots, x_n) = f_1(x_1) \square f_2(x_2) \square \ldots \square f_n(x_n),$$

где $\square$ есть оператор композиции (это обозначение предполагает ассоциативность операции $\square$, но это свойство не необходимо).

Такие функции очевидным образом разделимы, и поэтому остается проверить, что операция композиции $\square$ удовлетворяет условию монотонности.

Изучим кратко некоторые частные случаи, наиболее часто встречающиеся в приложениях.

a) Случай аддитивных функций.

Тогда $\square$ — сложение вещественных чисел. Такие функции очевидным образом разложимы и строго разложимы.

b) Произведение неотрицательных вещественных функций.

Тогда $\square$ — умножение вещественных чисел. В этом случае для установления свойства монотонности нужно предположить, что элементарные функции, композиция которых дает функцию f, неотрицательны. Для установления строгой монотонности нужно исключить элемент 0, иначе говоря, предположить, что все элементарные функции $f_1, \ldots, f_n$ строго положительны. Заметим, что случай произведения вещественных функций, которые все строго положительны, немедленно сводится к аддитивной функции, если от данной функции взять логарифм (так как логарифм есть монотонно возрастающая функция, то все равно, что именно оптимизировать — функцию f или $\ln(f)$).

c) Минимум (соответственно максимум) n вещественных функций.

В случае, если операция $\square$ — минимум (соответственно максимум) двух вещественных чисел, наша функция f имеет вид

$$f(x_1, \ldots, x_n) = \mathrm{Opt}\{f_1(x_1); \ldots; f_n(x_n)\}$$

(где $\mathrm{Opt} = \min$ или $\max$), и можно написать

$$f(x_1, \ldots, x_n) = \mathrm{Opt}\{f_1(x_1); z\} = f'(x_1, z)$$

с $z = \mathrm{Opt}\{f_2(z_2); \ldots; f_n(x_n)\}$.

Функция f' удовлетворяет условию монотонности, так как

$$z' \geqslant z'' \Rightarrow f'(x_1, z') = \mathrm{Opt}\{f_1(x_1); z'\} \geqslant \mathrm{Opt}\{f_1(x_1); z''\} = f'(x_1, z'').$$

Отсюда следует, что функция f разложима. Заметим, однако, что функция f' не является строго монотонной и, следовательно,

принцип оптимальности (см. п. 2.5) может не удовлетворяться всеми оптимальными решениями.

Замечание. В дальнейшем в этой главе мы для упрощения изложения ограничимся оптимизацией разложимых функций аддитивного типа, т. е. таких, что

$$f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n).$$

Вследствие всего предыдущего обобщение на разложимые функции вида

$$f_1(x_1) \square \ldots \square f_n(x_n)$$

оказывается очевидным: для этого достаточно заменить сложение $+$ подходящей операцией композиции $\square$.

2.7. Динамическое программирование с конечным числом состояний и поиск кратчайшего (или длиннейшего) пути в графе. Мы обобщим здесь построение, данное в п. 1.4.

Рассмотрим случай, в котором множество состояний $\mathscr{E}$ имеет конечную мощность $|\mathscr{E}| = p$.

Кроме того, в соответствии с замечанием из п. 2.6, предположим, что f имеет вид

$$f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n).$$

Тогда рассмотрим граф $G = [X, U]$, множество вершин X и множество дуг U которого строятся следующим образом.

Множество X состоит из:

a) особой вершины, которую мы обозначим I (начальная вершина);

b) np вершин, каждая из которых соответствует паре (E, i), где $E \in \mathscr{E}$ и $1 \leqslant i \leqslant n$.

Множество U содержит:

a) множество дуг вида $[I, (E, 1)]$, где $E = h_1(x_1)$ для всех возможных значений x_1 (иначе говоря, таких, что $h_1(x_1) \in \mathscr{E}$).

Дуге $[I, (E, 1)]$ приписывается длина, равная $f_1(x_1)$;

b) для $i = 2, 3, \ldots, n$ и для любого $E \in \mathscr{E}$ — множество дуг вида $[(E, i-1), (E', i)]$, где $E' = h_i(x_i, E)$ для всех возможных значений x_i (иначе говоря, таких, что $h_i(x_i, E) \in \mathscr{E}$).

Дуге $[(E, i-1), (E', i)]$ приписывается длина, равная $f_i(x_i)$.

Подмножество конечных вершин X_t образовано всеми вершинами вида (E, n) с $E \in \mathscr{E}_t$ ($\mathscr{E}_t$ — множество конечных состояний).

Заметим тогда, что:

— любой путь между I и X_t является решением задачи (т. е. удовлетворяет ограничениям $g(x_1, \ldots, x_n) \in \mathscr{E}_t$);

— длина этого пути равна $f_1(x_1) + \ldots + f_n(x_n)$.

Отсюда следует, что решение данной задачи динамического программирования равносильно поиску кратчайшего (или длиннейшего) пути между I и X_t в графе G.

2.8. Второй алгоритм динамического программирования. Процедура «с начала». Можно использовать эквивалентность с задачей о кратчайшем (длиннейшем) пути, предложенную в предыдущем

пункте, и ввести с ее помощью второй алгоритм динамического программирования, в котором вычисления осуществляются на этот раз в соответствии с порядком возрастания номеров переменных (отсюда и название процедуры: «с начала»). Как мы сейчас увидим, этот алгоритм отличается от алгоритма 2, по существу, только тем, что получается из него простым обращением порядка следования переменных.

Рассмотрим снова случай динамического программирования с конечным числом состояний и рассмотрим граф G из п. 2.7.

Как и многие другие алгоритмы кратчайшего (длиннейшего) пути, следующий алгоритм является алгоритмом маркировки: каждой вершине типа (E, i) сопоставляется число (метка) $\pi(E, i)$.

В конце $(i-1)$-го шага величина $\pi(E, i-1)$ явно представляет длину оптимального пути между начальной вершиной I и $(E, i-1)$. В начале i-го шага имеем $\forall E$: $\pi(E, i)=+\infty$ (случай кратчайшего пути) или $\pi(E, i)=-\infty$ (случай длиннейшего пути).

Шаг i состоит в изменении этих меток, исходя из оптимальных значений $\pi(E, i-1)$, причем изменения происходят так, чтобы к концу шага i величина $\pi(E, i)$ представляла явным образом длину кратчайшего (длиннейшего) пути, соединяющего начальную вершину с данной вершиной.

Алгоритм 3 (процедура «с начала»).

а) Шаг $i=1$:

$\forall E \in \mathscr{E}$ выполнять: $\pi(E, 1)=+\infty$ (случай $\mathrm{Opt}=\min$),

$\pi(E, 1)=-\infty$ (случай $\mathrm{Opt}=\max$).

Затем для всех возможных значений x_1 (т. е. таких x_1, что $h_1(x_1)\in\mathscr{E}$) определить $E=h_1(x_1)$ и выполнить $\pi(E, 1) \leftarrow f_1(x_1)$.

b) Для $i=2, 3, \ldots, n$.

Шаг i:

Выполнять $\forall E \in \mathscr{E}$: $\pi(E, i)=+\infty$ $(\mathrm{Opt}=\min)$,

$\pi(E, i)=-\infty$ $(\mathrm{Opt}=\max)$.

Для любого такого E, что $|\pi(E, i-1)|<+\infty$, и для всех возможных значений x_i (т. е. таких x_i, что $E'=h_i(x_i, E)\in\mathscr{E}$):

— вычислить: $E'=h_i(x_i, E)$;

— выполнять: $\pi(E', i) \leftarrow \mathrm{Opt}\{\pi(E', i);\ \pi(E, i-1)+f_i(x_i)\}$.

c) Вычислить F^* с помощью формулы

$$F^* = \operatorname*{Opt}_{E\in\mathscr{E}_I} \{\pi(E, n)\}.$$

Если мы хотим получить не только оптимальное значение F^*, но и само оптимальное решение x^*, то нужно определять для каждого $i=1, \ldots, n$ и для каждого $E\in\mathscr{E}$ величины $\bar{x}_i(E)$, дающие значение переменного x_i, позволяющие получить оптимальное значение $\pi(E, i)$.

На шаге 1 положим $\bar{x}_1(E)=x_1$ (где $E=h_1(x_1)$), а на шаге $(2\leqslant i\leqslant n)$ положим $\bar{x}_i(E)=x_i$ всякий раз, когда метка $\pi(E', i)$ улучшается и заменяется на $\pi(E, i-1)+f_i(x_i)$.

Это позволяет в конце алгоритма 3 получить явное оптимальное решение $x^* = (x_1^*, \ldots, x_n^*)$ с помощью процедуры

a) $E = E^*$, где E^* удовлетворяет условию

$$F^* = \operatorname*{Opt}_{E \in \mathscr{E}_t} \{\pi(E, n)\} = \pi(E^*, n);$$

b) для $i = n, n-1, \ldots, 1$ последовательно выполнять

$x_i^* \leftarrow \bar{x}_i(E)$,

$E \leftarrow E'$, где E' удовлетворяет условию $E = h_i(x_i^*, E')$.

Напомним, что в некоторых случаях восстановление оптимального решения x^* может потребовать меньшего объема информации (см. п. 1.2 и [20]).

§ 3. Техника редукции вычислений в динамическом программировании

Для задач, приводящих к пространству состояний очень большой мощности (и, в частности, если число ограничений, т. е. размерность пространства состояния, велико), развиты специальные технические приемы либо для уменьшения размерности пространства состояния, либо для удаления из вычислений существенной доли априори возможных состояний.

Среди наиболее интересных приемов можно указать:

— технику множителей Лагранжа (лагранжево ослабление ограничений) (см. [2, 16]), оно будет изучено в п. 3.1;

— использование в динамическом программировании методов разделения и оценки (см. гл. 7, § 2). См., например, [51, 54, 45, 52]. Это будет предметом п. 3.2;

— алгоритм A^* (или алгоритм допустимого поиска), используемый для поиска кратчайших или длиннейших путей в графах больших размерностей [24, 47, 35], приложения которого в области решения задач искусственного интеллекта весьма многочисленны (см., например, [34]). Его приложения к динамическому программированию будут изучены в п. 3.3;

— методы ослабления в пространстве состояний ([9, 10]), которые будут предметом п. 3.4.

3.1. Метод множителей Лагранжа. Общий принцип носит совершенно классический характер и уже был детально изложен в гл. 5, 6. Посмотрим кратко, как он может быть использован в задачах динамического программирования. Предположим, что мы хотим решить с помощью динамического программирования задачу вида (случай минимизации)

$$\begin{aligned} f(x_1, \ldots, x_n) &\to \min, \\ g(x_1, \ldots, x_n) &\leqslant 0, \end{aligned} \tag{I}$$

где g: $\mathbf{R}^n \to \mathbf{R}^m$.

Таким образом, есть m нелинейных ограничений вида $g_i(x_1, \ldots, x_n) \leqslant 0$ $(i = 1, \ldots, m)$ — если явно выписать все m компонент функции g.

Если непосредственно применить к этой задаче динамическое программирование, то мы придем к рассмотрению пространства состояний размерности m. Отсюда следует, что число состояний, вообще говоря, недопустимо велико, если m превосходит несколько единиц (см. п. 1.3)).

Тогда рассматриваемая идея состоит в попытке заменить решение задачи (I) решением последовательности задач, содержащих намного меньшее число ограничений.

Пусть $I_1 \subset \{1, \ldots, m\}$ — подмножество индексов ограничений, которые мы выбрали, чтобы их устранить $(|I_1| = m_1)$, и I_0 — множество индексов оставшихся ограничений $(|I_0| = m_0)$. С каждым ограничением $i \in I_1$ свяжем множитель Лагранжа $\lambda_i \geqslant 0$ и рассмотрим дуальную функцию $w(\lambda)$, значение которой (при данном λ) есть оптимальное решение задачи

$$\begin{aligned} &w(\lambda) = \min f(x) + \sum_{i \in I_1} \lambda_i g_i(x), \\ &g_i(x) \leqslant 0 \qquad \forall i \in I_0 \end{aligned} \tag{II}$$

(λ означает вектор размерности m_1, компоненты которого суть λ_i при $i \in I_1$).

Заметим, что целевая функция задачи (II) есть не что иное, как функция Лагранжа, связанная с задачей

$$\begin{aligned} &f(x) \to \min, \\ &g_i(x) \leqslant 0 \qquad \forall i \in I_1, \\ &x \in S, \end{aligned}$$

где множество S определяется равенством

$$S = \{x \mid g_i(x) \leqslant 0, \quad \forall i \in I_0\}$$

(см. гл. 5, § 2).

Если множества I_1 и I_0 выбраны так, что задача (II) содержит лишь очень небольшое число ограничений, то тогда эта задача может быть решена средствами динамического программирования.

При фиксированном λ обозначим через $\bar{x}(\lambda)$ оптимальное решение задачи (II).

Если поставленная задача (I) допускает седловую точку (этот случай имеет место, в частности, для выпуклых задач, но существуют и не выпуклые задачи, допускающие седловые точки (см. гл. 5, § 2)), то решение задачи (I) эквивалентно поиску седловой точки, т. е. решению дуальной задачи:

$$\begin{aligned} &w(\lambda) \to \max, \\ &\lambda \geqslant 0 \end{aligned} \tag{I'}$$

(см. гл. 6, § 2).

Для оптимизации дуальной задачи можно использовать один из методов оптимизации без ограничений, описанных в гл. 4: метод

наискорейшего спуска, сопряженного градиента или квазиньютоновский метод — если дуальная функция w дифференцируема, метод субградиента — если w не всюду дифференцируема.

Решение задачи (I) сводится тогда к решению (с помощью динамического программирования) последовательности задач, для которых размерности связанных с ними пространств состояний очень малы.

Заметим, что полезность метода, к счастью, не ограничена задачами, для которых седловая точка существует. Действительно, мы знаем (см. гл. 6), что для произвольного фиксированного λ вектор $\bar{x}(\lambda)$ есть оптимальное решение задачи (называемой «возмущенной» задачей)

$$\begin{aligned} & f(x) \to \min, \\ & g_i(x) \leqslant g_i(\bar{x}(\lambda)) \qquad \forall i \in I_1, \\ & g_i(x) \leqslant 0 \qquad\qquad \forall i \in I_0. \end{aligned} \tag{III}$$

Но в окрестности оптимума дуальной функции w значения $g_i(\bar{x}(\lambda))$ (которые являются компонентами градиента или субградиента функции w в точке λ, см. гл. 6, п. 2.7), вообще говоря, близки к нулю. Это означает, что рассматриваемый метод совершенно естественно приводит к хорошим приближенным решениям рассматриваемой задачи (решениям, в которых ограничения удовлетворяются с точностью до ε). Это свойство тем более интересно, что в многочисленных приложениях можно удовлетвориться выполнением некоторых ограничений с точностью до ε.

Заметим здесь, что выбор удаляемых ограничений (выбор подмножества I_1), вообще говоря, не безразличен с точки зрения скорости сходимости или качества заключительного результата. Вследствие всего сказанного выше мы можем использовать правило, состоящее в удалении в первую очередь ограничений, для которых **не** требуется точное выполнение.

Наконец, для комбинаторных (целочисленных) задач типа (I) можно использовать оптимальное значение дуальной задачи (I') как оценочную функцию (миноранту) в методах разветвленного поиска (см. гл. 7).

3.2. Комбинация динамического программирования с методами разделения и оценки (branch and bound). Динамическое программирование приводит к алгоритмам, которые являются существенно перечислительными. Как и алгоритмы перечислительного типа, описанные в гл. 7, § 2 (методы разветвления с помощью разделения и оценки), они могут быть, следовательно, существенно улучшены с использованием оценок снизу (минорант) в случае задач минимизации или оценок сверху (мажорант) в случае задач максимизации.

Возьмем случай, когда число состояний конечно, и рассмотрим приложение алгоритма 3 (процедуры «с начала») к задаче вида

$$\begin{aligned} & f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n) \to \min, \\ & g(x_1, \ldots, x_n) \in \mathscr{E}_t \subset \mathbf{R}^m. \end{aligned}$$

Предположим, что:

— известно некоторое решение x^0 этой задачи, например, решение, получаемое хорошим приближенным (эвристическим) алгоритмом; следовательно, значение $f(x^0)$ является мажорантой (неизвестного) оптимального значения $f(x^*)$;

— для каждого $i = 1, \ldots, n$ и для любого $E \in \mathscr{E}$ мы располагаем эффективным алгоритмом получения оценки снизу (миноранты) оптимального значения $F_i(E)$ задачи

$$\left.\begin{aligned} &f_i(x_i) + f_{i+1}(x_{i+1}) + \ldots + f_n(x_n) \to \min, \\ &\bar{h}_i(x_i, x_{i+1}, \ldots, x_n, E) \in \mathscr{E}_i. \end{aligned}\right\} \quad \pi_i(E)$$

Напомним (см. п. 2.4), что функция $\bar{h}_i$ рекурсивно определяется равенствами

$$\begin{aligned} E_i &= h_i(x_i, E), \\ E_{i+1} &= h_{i+1}(x_{i+1}, E_i), \\ &\ldots\ldots\ldots\ldots \\ E_n &= h_n(x_n, E_{n-1}), \\ \bar{h}_i(x_i, x_{i+1}, \ldots, x_n, E) &= E_n. \end{aligned}$$

Мы обозначим через $ev(E, i)$ значение этой оценки снизу (следовательно, $ev(E, i) \leqslant F_i(E)$).

Рассмотрим тогда оптимальные метки $\pi(E, i)$, получаемые в конце каждого этапа — с номером i — в алгоритме 3, и предположим, что для состояния $E \in \mathscr{E}$ выполняется соотношение

$$\pi(E, i) + ev(E, i+1) > f(x^0),$$

откуда следует, что

$$\pi(E, i) + F_{i+1}(E) > f(x^0).$$

Однако в связанном с рассматриваемой задачей графе G (п. 2.7) величина $\pi(E, i)$ представляет собой длину кратчайшего пути между начальной вершиной I и вершиной (E, i), а $F_{i+1}(E)$ представляет собой длину кратчайшего пути между вершиной (E, i) и множеством X_t конечных вершин.

Предыдущее неравенство, таким образом, показывает, что любой путь между I и X_t, подчиняющийся требованию проходить через вершину (E, i), имеет длину, бо́льшую $f(x^0)$, и, следовательно, не может составлять оптимального решения задачи.

Отсюда заключаем, что никакой оптимальный путь (никакое оптимальное решение) не будет отброшен, если мы удалим вершину (E, i) из всех последующих рассмотрений (такая вершина будет называться «бездействующей»).

Таким образом, рассматриваемый метод позволяет избежать в алгоритме 3 испытания большого числа вершин графа G и, таким образом, решать задачи, для которых число состояний $|\mathscr{E}| = p$ априори очень велико.

Очевидно, что для достижения эффективной процедуры на каждом шаге i нужно учитывать только вершины, которые действи-

тельно могут встретиться в оптимальном решении (так называемые «активные» вершины), т. е. такие вершины (E, i), что

$$\pi(E, i) + ev(E, i+1) \leqslant f(x^0).$$

Для этого мы используем структуру данных следующего типа.

Активные вершины будут перенумерованы индексом $k = 1, 2, \ldots, K$ в порядке их появления при развертывании алгоритма. Активная вершина, соответствующая начальной вершине, получает номер 0.

Пусть в данный момент число порожденных к этому моменту активных вершин равно K. Каждому числу k $(1 \leqslant k \leqslant K)$ сопоставляется следующая информация:

— $\mathrm{Etat}(k)$, $\mathrm{ind}(k)$, обозначающие соответственно значение состояния и номер шага, связанные с активной вершиной k. Иначе говоря, активная вершина с номером k соответствует вершине $(\mathrm{Etat}(k), \mathrm{ind}(k))$ на графе G;

— $\bar{\omega}(k)$, представляющее собой метку вершины $(\mathrm{Etat}(k), \mathrm{ind}(k))$ графа G, соответствующей активной вершине с номером k;

— указатель $\mathrm{pred}(k)$; $k' = \mathrm{pred}(k)$ представляет собой номер активной вершины, непосредственно предшествующей вершине k в оптимальном пути, соединяющем начальную вершину I (активную вершину с номером 0) с активной вершиной k;

— значение $v(k)$ переменного (с индексом $\mathrm{ind}(k)$), дающее возможность перейти от активной вершины $k' = \mathrm{pred}(k)$ к активной вершине k (т. е. от состояния $\mathrm{Etat}(k')$ к состоянию $\mathrm{Etat}(k)$). Эта информация позволяет явно выразить соответствующее решение для оптимального пути (длины $\bar{\omega}(k)$) между I (активной вершиной с номером 0) и активной вершиной k.

Тогда алгоритм 3 принимает следующую форму (случай минимизации).

Алгоритм 3′.

a) $f(x^0)$ есть стоимость решения x^0.

$K = 0$.

Шаг $i = 1$:

Для всех возможных значений переменного x_1 выполнять:

$$K \leftarrow K + 1,$$
$$\mathrm{Etat}(K) \leftarrow h_1(x_1),$$
$$\mathrm{ind}(K) \leftarrow 1,$$
$$\bar{\omega}(K) \leftarrow f_1(x_1),$$
$$\mathrm{pred}(K) \leftarrow 0,$$
$$v(K) \leftarrow x_1.$$

b) $i \leftarrow i + 1$.

Шаг i:

b1) Для всех активных вершин k, удовлетворяющих условиям $\mathrm{ind}(k) = i - 1$ и $\bar{\omega}(k) < +\infty$:

$$E = \mathrm{Etat}(k).$$

Для всех возможных значений переменного x_i выполнять

$$E' = h_i(x_i, E).$$

Выяснить, имеется ли среди вершин l, уже порожденных на шаге i (т. е. таких вершин, что $\mathrm{ind}(l) = i$), такая вершина l_0, что

$$\mathrm{Etat}(l_0) = E'.$$

Если такая вершина l_0 существует и если $\bar{\omega}(k) + f_i(x_i) < \bar{\omega}(l_0)$, то выполнять:

$$\bar{\omega}(l_0) \leftarrow \bar{\omega}(k) + f_i(x_i),$$
$$\mathrm{pred}(l_0) \leftarrow k,$$
$$v(l_0) \leftarrow x_i.$$

Если такой вершины l_0 нет, то тогда породить новую вершину:

$$K \leftarrow K + 1,$$
$$\mathrm{Etat}(K) \leftarrow E',$$
$$\mathrm{ind}(K) \leftarrow i,$$
$$\bar{\omega}(K) \leftarrow \bar{\omega}(k) + f_i(x_i),$$
$$\mathrm{pred}(K) \leftarrow k,$$
$$v(K) \leftarrow x_i.$$

b2) Если $i = n$, то перейти к c). Иначе:

Для всех вершин k, порожденных на шаге i (т. е. таких, что $\mathrm{ind}(k) = i$), выполнять:

$$E = \mathrm{Etat}(k).$$

Вычислить $ev(E, i+1)$.

Если $\bar{\omega}(k) + ev(E, i+1) > f(x^0)$, то выполнять: $\bar{\omega}(k) \leftarrow +\infty$ (такая вершина является бездействующей и не должна рассматриваться на шаге $i+1$).

b3) Вернуться к b) (переход к следующему шагу).

c) $F^* = +\infty$.

Для всех активных вершин k, порожденных на шаге n (т. е. таких, что $\mathrm{ind}(k) = n$), выполнять:

$$E = \mathrm{Etat}(k).$$

Если $E \in \mathscr{E}_1$ и если $\bar{\omega}(k) < F^*$, то

$$F^* \leftarrow \bar{\omega}(k),$$
$$k^* \leftarrow k.$$

В конце алгоритма 3′ можно легко восстановить оптимальное решение x^* с помощью следующей процедуры:

$$k \leftarrow k^*.$$

Для $i = n, n-1, \ldots, 1$ последовательно выполнять:

$$x_i^* := v(k),$$
$$k \leftarrow \mathrm{pred}(k).$$

Этот алгоритм, который применялся ко многим задачам (см., например, [54, 32, 33, 42]), вообще говоря, проявил себя на практике очень эффективно. Морен и Марстен [45] сообщают, что для задач типа «рюкзака» по сравнению с классическими алгоритмами динамического программирования имело место улучшение в 10^3 раз по объему памяти и в 10^2 раз по времени вычислений.

В заключение сообщим, что основная идея, представленная в этом разделе, допускает многочисленные варианты: удаление бездействующих вершин (т. е. не входящих в оптимальное решение) может осуществляться и с помощью других тестов — не только таких, которые основаны на оценке снизу; можно применить тесты реализуемости, использовать различные необходимые условия оптимальности и т. д.

Замечание 1. Хранение информации вида $\mathrm{Etat}(k)$ для всех активных вершин k может потребовать значительного объема памяти, особенно если вектор состояния имеет сколько-нибудь значительную размерность. Вообще говоря, можно избежать хранения этой информации, так как для определения состояния, связанного с активной вершиной k, достаточно знать последовательность принятых решений, которые позволили достичь этой вершины, иначе говоря, знать значения переменных. Но эти последние сразу получаются исходя из информации о $\mathrm{pred}(\cdot)$ и $v(\cdot)$ с помощью действий

$$l \leftarrow k,$$
$$i = \mathrm{ind}(k),$$

для $j = i, i-1, \ldots, 1$ последовательно выполнять:

$$x_j = v(l),$$
$$l \leftarrow \mathrm{pred}(l).$$

Замечание 2. Можно разными способами определять оценки снизу. Если поставленная задача является задачей целочисленного линейного программирования, то в качестве оценки снизу можно использовать значение непрерывного оптимального решения, получаемого симплексным алгоритмом (см. гл. 2).

Очень часто используемый в последнее время в литературе (и, вообще говоря, более эффективный) метод состоит в лагранжевом ослаблении и (приближенном) решении двойственной (к ослаблению) задачи с помощью алгоритма субградиента (см. гл. 4, § 3, гл. 6 и п. 3.1 настоящей главы).

Другой технический прием, который был с успехом применен при решении некоторых комбинаторных задач (см. [9, 10]), — это ослабление в пространстве состояний; ему посвящен п. 3.4 ниже.

3.3. Алгоритмы допустимого поиска. Алгоритм A^*. Алгоритмы допустимого поиска (admissible search algorithms) были развиты в области искусственного интеллекта, где очень многие задачи (игры, автоматическая демонстрация, общие программы решения задач) могут рассматриваться как поиск оптимальных путей в графах слишком больших размерностей (см. [48, 47]).

Вообще говоря, эти графы описываются неявно следующим образом:

— начальная вершина 0 дана;

— для любой вершины s (в частности, для $s=0$) мы умеем автоматически порождать множество $\Gamma(s)$ всех вершин, следующих за s, т. е. множество таких вершин s', что (s, s') есть дуга рассматриваемого графа (предполагается, что $\Gamma(s)$ имеет конечную мощность для любого s. Такой граф называется тогда *локально конечным*, но он может очевидным образом содержать бесконечно много дуг и вершин).

С другой стороны, предполагается, что для каждого $s' \in \Gamma(s)$ мы умеем вычислять длину $l(s, s')$ дуги (s, s').

Пусть дано подмножество T вершин, которое тоже определено неявно (это множество конечных вершин или «целей»); мы хотим определить кратчайший путь между 0 и T.

Основная идея метода состоит в том, что нет необходимости явно описывать весь граф, чтобы определить оптимальный путь.

Предположим, что для любой вершины s у нас есть возможность вычислить величину $\varphi(s)$, являющуюся приближением снизу (минорантой) длины $\varphi^*(s)$ кратчайшего пути между s и подмножеством вершины T.

В следующем алгоритме, который можно рассматривать как обобщение алгоритма Мура — Дийкстра (см. [40, 13]), последовательно шаг за шагом строится такое подмножество вершин S, что для любого $s \in S$ уже определен кратчайший путь между 0 и s и его длина $\pi(s)$. На каждом шаге мы присоединяем к S такую вершину s, что величина

$$\psi(s) = \pi(s) + \varphi(s)$$

минимальна (среди всех вершин $s \notin S$).

Следовательно, алгоритм завершает работу, когда одна из вершин, входящих в T, отбирается (на шаге b)), чтобы быть включенной в множество S.

Алгоритм 4. (Алгоритм A^*.)

a) $S = \varnothing$, $\bar{S} = \{0\}$,

$$\pi(0) = 0,$$

$$\psi(0) = \pi(0) + \varphi(0).$$

b) Определить такую вершину $s \in \bar{S}$, что

$$\psi(s) = \min_{s' \in \bar{S}} \{\psi(s')\}.$$

Выполнить: $S \leftarrow S + \{s\}$ и $\bar{S} \leftarrow \bar{S} - \{s\}$.

Если $s \in T$, то закончить: найден некоторый кратчайший путь между 0 и множеством T.

Иначе:

Для всех вершин $s' \in \Gamma(s)$.

Если $s' \notin S \cup \bar{S}$, то вычислить $\varphi(s')$ и выполнять:

$$\bar{S} \leftarrow \bar{S} + \{s'\},$$
$$\psi(s') \leftarrow \pi(s) + l(s, s') + \varphi(s'),$$
$$\pi(s') \leftarrow \pi(s) + l(s, s'),$$
$$\mathrm{pred}(s') \leftarrow s.$$

Если $s' \in S \cup \bar{S}$ и если $\pi(s) + l(s, s') + \varphi(s') < \psi(s')$, то выполнять:

$$\psi(s') \leftarrow \pi(s) + l(s, s') + \varphi(s'),$$
$$\pi(s') \leftarrow \pi(s) + l(s, s'),$$
$$\mathrm{pred}(s') \leftarrow s.$$

Если $s' \in S$, то

$$S \leftarrow S - \{s'\},$$
$$\bar{S} \leftarrow \bar{S} + \{s'\}.$$

c) Вернуться к b).

Тогда можно доказать следующие свойства ([24, 47, 35]):

— если существует оптимальный путь конечной мощности между 0 и T, то алгоритм 4 позволяет получить такой путь за конечное число шагов;

— если оценка $\varphi(s)$ в точности равна длине $\varphi^*(s)$ кратчайшего пути между s и T, то все вершины s, выбираемые на шаге b), являются вершинами, принадлежащими оптимальному пути между 0 и T. Число вершин, встречающихся тогда явным образом в алгоритме, крайне ограничено;

— если $\varphi(s) < \varphi^*(s)$, то число вершин, встречающихся в алгоритме явным образом, тем меньше, чем меньше разность $\varphi^*(s) - \varphi(s)$;

— если функция φ удовлетворяет, кроме того, условию, называемому условием состоятельности:

$$\varphi(s) \leqslant l^*(s, s') + \varphi(s') \qquad \forall s, \forall s'$$

(где $l^*(s, s')$ обозначает длину кратчайшего пути между вершинами s и s'), то любая вершина, выбранная в b) на некотором шаге для того, чтобы быть включенной в S, не может быть удалена из S на более позднем шаге.

Заметим, с другой стороны, что можно применить алгоритм 4, беря в качестве φ просто хорошую аппроксимацию для $\varphi^*(s)$ (вместо приближения снизу). При этих условиях оптимальность полученного решения больше не гарантируется, но, вообще говоря, получаются хорошие приближенные решения.

Применение алгоритма 4 к графу G, связанному с задачей динамического программирования (см. п. 2.7), совершенно очевидно; множество $\Gamma(s)$ вершин, следующих за вершиной $s=(E, i)$, определяется на самом деле как множество вершин вида $s' \in (E', i+1)$, где $E' = h_{i+1}(x_{i+1}, E)$ для всех возможных значений x_{i+1}. С другой стороны, для данной вершины $s=(E, i)$ возьмем $\varphi(s)$ равной величине $ev(E, i+1)$, являющейся оценкой с недостатком оптимального решения $F_{i+1}(E)$ задачи

$$\begin{aligned} &f_{i+1}(x_{i+1}) + \ldots + f_n(x_n) \to \min, \\ &h_{i+1}(x_{i+1}, \ldots, x_n, E) \in \mathscr{E}_i \end{aligned} \qquad \pi_{i+1}(E)$$

(см. п. 3.2).

Приведенные выше различные свойства показывают, что алгоритм 4 может представлять собой очень эффективный метод решения для задач динамического программирования больших размерностей. Этот метод близок методу, описанному в п. 3.2, в том смысле, что он также использует оценку снизу, но следует заметить, что его эффективность не зависит от знания априори хорошего приближенного решения задачи, и в этом его главное преимущество.

Для получения оценок снизу можно воспользоваться замечанием 2 п. 3.2.

3.4. Методы ослабления в пространстве состояний. Рассмотрим задачу вида

$$\begin{aligned} &F^* = \min f(x_1, \ldots, x_n), \\ &g(x_1, \ldots, x_n) \in \mathscr{E}_t \subset \mathbf{R}^m, \end{aligned} \qquad (IV)$$

где $f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n)$ и где g определена набором функций перехода $h_1, \ldots, h_n$ с помощью равенств

$$\begin{aligned} E_1 &= h_1(x_1), \\ E_2 &= h_2(x_2, E_1), \\ &\cdots\cdots\cdots \\ E_n &= h_n(x_n, E_{n-1}), \\ g(x_1, \ldots, x_n) &= E_n, \end{aligned}$$

где

$$\begin{aligned} h_1&: \mathbf{R} \to \mathscr{E}, \\ h_2&: \mathbf{R} \times \mathscr{E} \to \mathscr{E}, \\ &\cdots\cdots\cdots \\ h_n&: \mathbf{R} \times \mathscr{E} \to \mathscr{E}. \end{aligned}$$

Здесь $\mathscr{E} \subset \mathbf{R}^m$ есть множество состояний (имеющее по предположению конечную мощность), а $\mathscr{E}_t$ — множество конечных состояний.

Предположим, что мощность пространства состояний $\mathscr{E}$ слишком велика для того, чтобы допустить непосредственное примене-

вие алгоритма динамического программирования (алгоритма 2 п. 2.4).

Идея методов ослабления в пространстве состояний заключается в преобразовании данной задачи на другом пространстве состояний $\tilde{\mathscr{E}}$, являющемся образом $\mathscr{E}$ при отображении φ (причем φ выбирается так, чтобы мощность $\tilde{\mathscr{E}}$ была много меньше мощности $\mathscr{E}$), и решении задачи-«образа»:

$$F^* = \min f(x_1, \ldots, x_n),$$
$$\tilde{g}(x_1, \ldots, x_n) \in \tilde{\mathscr{E}}_t,$$

где $\tilde{g} = \varphi \circ g$, а $\tilde{\mathscr{E}}_t = \varphi(\mathscr{E}_t)$.

При некоторых условиях на отображение φ (мы их укажем явно ниже) задача (V) может быть решена с помощью динамического программирования на пространстве состояний много меньшей мощности.

Отметим сразу же два следующих свойства.

1) Любое решение задачи (IV) есть также решение задачи (V) (но, вообще говоря, решения задачи (V) могут не быть решениями задачи (IV)), и вследствие этого имеем $F^* \leqslant F^*$.

2) Если $\tilde{x}^*$ — оптимальное решение задачи (V), удовлетворяющее ограничениям задачи (IV), то $\tilde{x}^*$ является также оптимальным решением задачи (IV) (в этом случае имеем $F^* = F^*$).

Задача (V) называется *ослаблением* задачи (IV).

Свойство 2) показывает, что (при хорошо подобранной функции φ) можно получить оптимальное решение исходной задачи, решая ослабленную задачу. С другой стороны, даже если это не так, свойство 1) показывает, что решение ослабленной задачи позволяет получить миноранту искомого оптимального решения.

Эта информация может быть использована тогда в качестве оценки снизу:

— в методах разветвления с помощью разделения и оценки (гл. 7);

— в алгоритмах динамического программирования, описанных в пп. 3.2 и 3.3.

Чтобы иметь возможность решить ослабленную задачу с помощью динамического программирования, нужно найти некоторое представление с помощью функций перехода на множестве $\tilde{\mathscr{E}}$.

Для этого достаточно, например, чтобы функция φ обладала следующим свойством:

$$\forall i (2 \leqslant i \leqslant n), \quad \forall x_i$$
$$E \in \mathscr{E}, \quad E' \in \mathscr{E}, \quad \varphi(E) = \varphi(E') \Rightarrow \varphi[h_i(x_i, E)] = \varphi[h_i(x_i, E')].$$

Из этого свойства следует существование функций перехода $\tilde{h}_i$, определенных на $\mathbf{R} \times \tilde{\mathscr{E}}$ и таких, что

$$\varphi[h_i(x_i, E)] = \tilde{h}_i(x_i, \varphi(E)).$$

Тогда функция $\tilde{g} = \varphi \circ g$ может быть рекурсивно определена соотношениями

$$\tilde{E}_1 = \varphi[h_1(x_1)] = \tilde{h}_1(x_1),$$
$$\tilde{E}_2 = \tilde{h}_2(x_2, \tilde{E}_1),$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$
$$\tilde{E}_n = \tilde{h}_n(x_n, \tilde{E}_{n-1}),$$
$$\tilde{g}(x_1, \ldots, x_n) = \tilde{E}_n,$$

и ослабленная задача $(\overline{V})$ может быть решена средствами динамического программирования на пространстве состояний $\tilde{\mathcal{E}}$ меньшей мощности — с помощью, например, алгоритма 2 п. 2.4 или алгоритма 3 п. 2.8.

Методы ослабления в пространстве состояний могут быть с успехом применены для решения некоторых комбинаторных задач, таких, как задачи прокладывания пути при наличии ограничений в графах, задачи коммивояжера при наличии ограничений (см. п. 4.3 и [9, 10]).

Заметим также, что алгоритмы, основанные на представлении задач целочисленного программирования с помощью конечных групп (см. гл. 7, § 4), могут рассматриваться как примеры ослабления в пространстве состояний в динамическом программировании, где используемые отображения φ суть гомоморфизмы групп.

§ 4. Примеры приложения динамического программирования

Мы приведем ниже некоторые из наиболее характерных примеров приложения динамического программирования, выявляя в каждом случае используемое представление с помощью пространства состояний и функции перехода и выписывая связанное с этим возвратное уравнение.

4.1. Оптимизация динамических систем. Речь идет об оптимизации систем, эволюция которых с течением времени управляется уравнением состояния и поведение которых может изменяться посредством переменных управления. Эта задача возникает во многих областях (планирование в экономике, оптимальное управление в автоматических системах и т. д.) и связана с большим количеством приложений.

Общая модель, соответствующая этому типу задач, формально совпадает с моделью, введенной в п. 2.4, и мы удовлетворимся поэтому тем, что дадим ее интерпретацию на языке теории систем.

Предположим, что изучаемая система может быть в каждый момент времени t охарактеризована заданием m-мерного вектора состояния E (например, если мы хотим управлять траекторией некоторого движущегося тела в пространстве, то состояние системы в каждый момент времени t будет определено, если известна пара положение — скорость).

Предположим, с другой стороны, что можно воздействовать на систему только в определенные моменты времени t_0, t_1, ..., t_n

посредством *переменных* (или *векторов*) *управления* $x_1, \ldots, x_n$ (например, в случае пространственного транспортного средства компоненты вектора управления в некоторый данный момент могут соответствовать усилиям различных реактивных двигателей, влияющих на положение транспортного средства).

Поведение системы определяется тогда заданием функций перехода $h_1, \ldots, h_n$, где для любого $i = 1, \ldots, n$ величина

$$h_i(\bar{x}_i, E_{i-1})$$

дает состояние E_i, в котором оказывается система в момент времени t, исходя из состояния E_{i-1} в момент времени t_{i-1} в результате применения значения $\bar{x}_i$ переменного (или вектора) управления x_i.

С другой стороны, каждому управлению x_i, осуществленному в момент времени t_{i-1}, когда система находилась в состоянии E_{i-1}, соответствует стоимость $c_i(x_i, E_{i-1})$ (например, в случае пространственного транспортного средства $c_i(x_i, E_{i-1})$ может быть количеством топлива, израсходованного между моментами времени t_{i-1} и t_i, если управление x_i применяется в момент времени t_{i-1}).

Исходя из состояния E_0 в момент времени t_0, мы хотим привести систему в подмножество состояний $\mathscr{E}_t$ (мишень) в момент времени t_n, минимизируя при этом полную стоимость.

Для данного множества управлений $x_1, \ldots, x_n$ конечное состояние системы $g(x_1, \ldots, x_n)$ определяется системой равенств

$$\begin{aligned} E_1 &= h_1(x_1, E_0), \\ E_2 &= h_2(x_2, E_1), \\ &\cdots\cdots\cdots \\ E_n &= h_n(x_n, E_{n-1}), \\ g(x_1, \ldots, x_n) &= E_n, \end{aligned}$$

и соответствующая полная стоимость равна $f(x_1, \ldots, x_n)$, определенной соотношением

$$f(x_1, \ldots, x_n) = c_1(x_1, E_0) + c_2(x_2, E_1) + \ldots + c_n(x_n, E_{n-1}).$$

Такая функция очевидным образом разложима (см. п. 2.6), что позволяет применить к задаче

$$\begin{aligned} F^* &= \min f(x_1, \ldots, x_n), \\ g(x_1, \ldots, x_n) &\in \mathscr{E}_t \end{aligned}$$

алгоритм динамического программирования.

Тогда возвратное уравнение динамического программирования записывается в виде

$$F_n(E) = \min_{\substack{x_n \\ h_n(x_n, E) \in \mathscr{E}_t}} \{c_n(x_n, E)\} \quad \forall E;$$

далее, для $1 \leqslant i \leqslant n-1$,

$$F_i(E) = \min_{x_i} \{c_i(x_i, E) + F_{i+1}(h_i(x_i, E))\} \quad \forall E,$$

$$F^* = F_1(E_0).$$

4.2. Кратчайший путь в графе. Пусть $G = [X, U]$ — некоторый ориентированный конечный граф, где X — множество его вершин ($|X| = N$), а U — множество его дуг ($|U| = M$).

Любой дуге $u = (i, j) \in U$ сопоставим ее длину $l_u = l(i, j)$.

Задача состоит тогда в определении пути минимальной длины между данной начальной вершиной 0 и заданным подмножеством вершин $T \subset X$ (множество конечных вершин). Заметим, что T может сводиться к единственной вершине.

Если граф G является секвенциальным графом (или, более общим образом, графом без замкнутых путей), то можно применить алгоритм п. 2.8: в этом случае каждой вершине графа G соответствует пара (состояние, шаг), и состояния, связанные с шагом i, суть вершины k графа G, для которых кратчайший путь (по числу дуг) между 0 и k содержит в точности i дуг.

Если граф G содержит замкнутые пути, то нужно использовать несколько другое представление состояния. Предположим, что в графе G нет замкнутых путей отрицательной длины (это условие необходимо для существования кратчайшего пути, см. [22], гл. 2).

В этом случае необходимо существует элементарный оптимальный путь, т. е. не содержащий замкнутых путей. Такой путь состоит из не более чем $N-1$ дуг.

Для каждой пары (k, i), где $k \in X$ и $1 \leqslant i \leqslant N-1$, обозначим через $F_i(k)$ длину кратчайшего пути между k и T, содержащего в точности i дуг. Если $\Gamma(k)$ обозначает множество вершин k', следующих за k в G (т. е. таких вершин, что (k, k') есть дуга в G), то возвратное соотношение между $F_i(k)$ записывается в виде

$$F_{i+1}(k) = \min_{k' \in \Gamma(k)} \{l(k, k') + F_i(k')\} \qquad \forall k \in X,$$

$$F_1(k) = \min_{k' \in \Gamma(k) \cap T} \{l(k, k')\} \qquad \forall k \in X,$$

$$F_1(k) = +\infty, \text{ если } \Gamma(k) \cap T = \varnothing.$$

Тогда длина кратчайшего пути между 0 и T равна

$$\min_{i=1,\ldots,N-1} \{F_i(0)\}.$$

Алгоритм динамического программирования, из которого вытекают вышеприведенные возвратные уравнения (называемый также алгоритмом *Беллмана — Калаба*), состоит из $N-1$ шага, и с каждым шагом связаны N состояний, каждое из которых соответствует некоторой вершине графа G.

Заметим, что этот алгоритм позволяет решить не только поставленную задачу, но и все задачи о *кратчайшем пути с ограничениями мощности* типа: найти кратчайший путь между 0 и T,

содержащий не более p дуг $(1 \leqslant p \leqslant N-1)$. Действительно, оптимальное значение этой задачи определяется равенством

$$\min_{i=1,\ldots,p} \{F_i(0)\}.$$

Если мы хотим восстановить явным образом оптимальные пути, то, как в п. 1.2, можно использовать таблицу, дающую для каждой вершины k номер вершины k', следующей за k в оптимальном пути от k до T.

4.3. **Задачи о гамильтоновых циклах минимальной длины.** Как и в п. 4.2, рассмотрим ориентированный граф $G=[X, U]$, в котором каждая дуга $u=(i, j)$ снабжена длиной $l(i, j)$.

Гамильтонов цикл есть замкнутый путь, который проходит через каждую вершину графа один и только один раз.

Речь идет тогда об определении гамильтонова цикла, полная длина которого минимальна. Эта задача, известная, вообще говоря, под названием *задачи коммивояжера* (здесь мы занимаемся ориентированным случаем), является предметом обильной литературы (см., например, [22], гл. 8).

Посмотрим здесь, как эту задачу можно — по крайней мере формально — решить с помощью динамического программирования.

Выберем произвольным образом какую-нибудь из вершин — скажем, вершину t — в качестве отправной точки путешествия и для любого подмножества вершин $S \subset X$, содержащего t, обозначим через $F(S, k)$ длину кратчайшего пути с началом k $(k \in S)$ и концом t, проходящего один и только один раз через все вершины S (по договоренности $F(S, k)=+\infty$, если такой путь не существует).

Тогда можно написать возвратное соотношение

$$\begin{aligned} F(S, k) &= \min_{k' \in \Gamma(k) \cap S} \{l(k, k') + F(S - \{k\}, k')\}, \\ F(\{t\}, t) &= 0, \\ F(\{t\}, k) &= +\infty \qquad \forall k \neq t. \end{aligned} \tag{6}$$

Мы видим, что для любого подмножества S мощности p значения $F(S, k)$ могут быть определены, исходя из значений $F(S', k')$ для всех подмножеств S' мощности $p-1$. Оптимальное значение поставленной задачи равно $\min_{k' \in \Gamma(t)} \{l(t, k') + F(X, k')\}$. Следовательно, процесс решения этой задачи с помощью динамического программирования состоит из N шагов (где N — число вершин графа), и на каждом шаге i множество состояний соответствует всем парам (S, k), где S — подмножество X, содержащее t и имеющее мощность i, причем $k \in S$.

Число состояний растет поэтому экспоненциально (как функция от размера задачи — числа вершин в графе), и в этом — причина, по которой рассматриваемая задача, вообще говоря, не может быть решена непосредственно с помощью уравнения (6).

Следовательно, мы применим один из технических приемов сокращения вычислений, обсуждавшихся в § 3. Например, в [10]

предложено использовать метод ослабления в пространстве состояний (п. 3.4) для решения задач коммивояжера с дополнительными ограничениями. Принцип этого применения состоит в следующем.

Рассмотрим новое пространство состояний, являющееся образом предыдущего пространства при отображении φ, заданном формулой

$$\varphi(S, k) = (|S|, k).$$

Таким образом, на каждом шаге i все состояния вида (S, k), где $S \subset X$, $|S| = i$ и $k \in S$, $t \in S$, имеют в качестве образа единственное состояние, обозначаемое (i, k). Таким образом, новое пространство состояний содержит только $N-1$ состояний на каждом шаге i $(1 \leqslant i \leqslant N)$. Следовательно, ослабленная задача может быть эффективно решена средствами динамического программирования, причем соответствующее возвратное уравнение имеет вид

$$\begin{gathered} F(i, k) = \min_{k' \in \Gamma(k)} \{l(k, k') + F(i-1, k')\}, \\ F(1, t) = 0, \quad F(1, k) = +\infty \qquad \forall k \neq t. \end{gathered} \tag{7}$$

Величины $F(i, k)$ можно интерпретировать следующим образом: $F(i, k)$ представляет собой длину кратчайшего пути с началом в k и концом в t и содержит в точности $i-1$ дуг. Оптимальное значение ослабленной задачи, равное $F(N+1, t)$, является оценкой *снизу для длины оптимального гамильтонова цикла*. Эта оценка снизу может быть использована в процедуре типа «разделения и оценки» (см. гл. 7) для точного решения исходной задачи.

4.4. Задачи спаривания, разбиения и покрытия гиперграфа интервалами. Если стандартный алгоритм динамического программирования часто не может быть применен для непосредственного решения этих задач, то, тем не менее, существуют случаи, когда можно добиться результата с помощью более эффективного метода решения. Сейчас мы приведем пример.

Задача о спаривании гиперграфа может быть сформулирована в полной общности как задача оптимизации с целочисленными переменными (0 *или* 1) следующего вида:

$$cx = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$Ax \leqslant 1,$$

$$x_j = 0 \text{ или } 1, \qquad j = 1, \ldots, n,$$

где $A = (a_{ij})$ — $m \times n$-матрица с коэффициентами 0 или 1, 1 есть m-вектор, все компоненты которого равны 1. Задача о спаривании графа есть частный случай задачи (PC), в котором матрица A есть матрица инцидентностей между вершинами и дугами в неориентированном графе (т. е. содержит в каждом столбце в точности две единицы).

Заметим, с другой стороны, что эта задача является также частным случаем задачи о многомерном рюкзаке (п. 1.3) и, следовательно, может быть решена средствами динамического программи-

рования, исходя из возвратного уравнения, установленного в п. 1.3 (так как мощность пространства состояний растет как 2^m, где m — число ограничений, то нужно, вообще говоря, использовать какой-нибудь из технических приемов сокращения вычислений, изложенных в § 3).

Мы изучим здесь частный случай задачи (PC), в котором матрица A обладает тем свойством, что в каждой строке ненулевые элементы (следовательно, элементы «1») идут друг за другом.

Пример.

$$A = \begin{bmatrix} 0&1&1&1&1&1&0&0 \\ 1&1&0&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&1&1&0 \\ 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&0&0 \end{bmatrix}.$$

Эти матрицы встречаются при постановке таких задач, как определение воздушного флота, требуемого для реализации данного множества полетов с минимальной стоимостью (см. [22], гл. 5, упражнения 23 и 29).

Предположим, конечно, что матрица A не является приводимой (т. е. не разбивается на блоки) и что все столбцы содержат по крайней мере один ненулевой элемент.

Как мы сейчас увидим, в этом случае динамическое программирование приводит к эффективному алгоритму для решения задачи (PC).

Для любого $l = 1, 2, \ldots, m$ обозначим через J_l множество $\{j | a_{lj} = 1\}$ (следовательно, J_l — подмножество номеров столбцов, содержащих ненулевой элемент в строке с номером l).

Можно доказать следующий результат.

Лемма 1. *Если для двух строк l и k имеем $J_l \subset J_k$, то ограничение l является избыточным и может быть исключено из задачи.*

Благодаря этому результату можно всегда ограничиться случаем такой матрицы A, что:

— в каждой строке ненулевые элементы идут друг за другом;

— для двух произвольных разных строк k и l имеем $J_l \not\subset J_k$ и $J_k \not\subset J_l$.

В предыдущем примере лемма 1 позволяет устранить строки 3 и 6. Остается матрица

$$A' = \begin{bmatrix} 0&1&1&1&1&1&0&0 \\ 1&1&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0 \\ 0&0&0&0&1&1&1&1 \end{bmatrix}.$$

Расположим теперь строки матрицы A' в порядке возрастания номеров их первых ненулевых элементов. В нашем примере полу-

часм (переставляя строки 1 и 2)

$$A'' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Матрица в форме A'' будет называться матрицей в *канонической форме.*

Заметим, что если матрица A имеет каноническую форму, то A обладает также тем свойством, что не только в каждой строке, но и в каждом столбце ненулевые элементы идут друг за другом. Тогда матрица A является матрицей инцидентности между вершинами и дугами для гиперграфа интервалов (см. [22], гл. 1), и задача (PC) является задачей спаривания на гиперграфе интервалов.

Для любого $k = 1, 2, \ldots, m$ рассмотрим следующее семейство задач:

$$F_k(0) = \max \sum_{j=1}^{n} c_j x_j,$$

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant 1, \qquad i = 1, \ldots, k \qquad P_k(0)$$

$$x_j = 0 \qquad \forall j \in J_k \cup J_{k+1} \cup \ldots \cup J_m,$$

и для каждого $l \in J_k$:

$$F_k(l) = \max \sum_{j=1}^{l} c_j x_j,$$

$$\sum_{j=1}^{l} a_{ij} x_j \leqslant 1, \qquad i = 1, \ldots, k, \qquad (P_k(l))$$

$$x_j = 0 \text{ или } 1, \qquad j = 1, \ldots, l-1,$$

$$x_l = 1.$$

Как непосредственное следствие определения величин $F_k(l)$ и $F_k(0)$ получаем возвратное соотношение

$$F_{k+1}(0) = \max \left\{ F_k(0); \max_{l \in J_k - J_{k+1}} \{F_k(l)\} \right\}$$

и $\forall p \in J_{k+1}$ получаем

$$F_{k+1}(p) = F_k(p), \qquad p \in J_k,$$

$$F_{k+1}(p) = c_p + F_{k+1}(0), \qquad p \notin J_k.$$

Отсюда можно немедленно вывести следующий алгоритм решения задачи (PC) (см. [21]).

Алгоритм 5. (Решение задачи спаривания для гиперграфа интервалов.)

a) Инициализация.
Положить $f_0 = 0$.
$\forall j \in J_1$ выполнять: $f(j) = c_j$
$k = 1$.
b) Шаг k:
Вычислить $h = \max\limits_{j \in J_k - J_{k-1}} \{f(j)\}$.
Выполнять: $f_0 \leftarrow \max\{f_0, h\}$
$\forall j \in J_{k+1} - J_k$ выполнять: $f(j) \leftarrow c_j + f_0$.
Пока $k < m$, выполнять $k \leftarrow k + 1$ и вернуться к b).
c) Вычислить $F^* = \max\left\{f_0, \max\limits_{j \in J_m} \{f(j)\}\right\}$.
Конец.

Следовательно, этот алгоритм динамического программирования содержит m шагов (m — число строк в матрице A), и каждый шаг требует изучения $|J_k| + 1$ состояний.

Доказано (см. [21]), что его сложность есть $O(n)$, т. е. необходимое для его работы число элементарных операций (сложений, сравнений) пропорционально n (числу переменных в задаче). Следовательно, этот алгоритм оптимален в смысле теории сложности (см. [19]).

Во всех тех случаях, когда матрица A есть матрица инцидентности некоторого гиперграфа интервалов, можно аналогичным образом построить основанные на динамическом программировании алгоритмы для решения эффективным образом так называемой задачи *разбиения*

$$\begin{gathered} cx \to \min, \\ Ax = 1 \qquad (PP) \\ x_j = 0 \text{ или } 1, \qquad j = 1, \ldots, n, \end{gathered}$$

а также задачи *покрытия*

$$\begin{gathered} cx \to \min, \\ Ax \geqslant 1, \qquad (PR) \\ x_j = 0 \text{ или } 1, \qquad j = 1, \ldots, n. \end{gathered}$$

За подробностями мы отсылаем к [21].

4.5. Динамическое программирование и стохастические системы: приложение к фильтрации марковских процессов. Динамическое программирование является также полезным инструментом изучения некоторых стохастических систем, в частности марковских систем, существенное свойство которых состоит в том, что в каждый момент времени t вся информация, касающаяся прошлого поведения системы, может быть заменена заданием вектора состояния.

Мы ограничимся здесь приведением примера, приложения которого к теории сигнала и в телекоммуникации особенно важны: это задача фильтрации марковского процесса (состояния и время дискретны), наблюдаемого в шуме без памяти. За более углубленным изложением мы отсылаем к [25, 26].

Рассмотрим процесс, определенный следующим образом.

В каждый момент времени k процесс может находиться в состоянии E_k — любом из конечного множества состояний $\mathscr{E} = \{1, \ldots, m\}$ (следовательно, m — число состояний процесса).

Если этот процесс развертывается на промежутке времени 0, 1, 2, ..., K, то реализация этого процесса есть последовательность состояний $E = (E_0, E_1, \ldots, E_K)$, где E_0 — начальное состояние (в момент времени 0), а E_K — конечное состояние (в конечный момент K).

С другой стороны, наш процесс является *марковским*, т. е. вероятность перехода из состояния E_k в момент времени k в состояние E_{k+1} в момент времени $k+1$ зависит только от состояния E_k: она не зависит от предшествующих состояний $E_0, E_1, \ldots, E_{k-1}$, что выражается равенством

$$P\{E_{k+1} | E_0, E_1, \ldots, E_k\} = P\{E_{k+1} | E_k\}.$$

Переход из состояния E_k в момент времени k к состоянию E_{k+1} в момент времени $k+1$ называется (*марковским*) *переходом*. Такой переход (E_k, E_{k+1}) в момент времени k будет обозначаться x_k. Множество переходов X будет тогда множеством пар (E, E') с $E \in \mathscr{E}$, $E' \in \mathscr{E}$.

Следовательно, марковский процесс с m состояниями полностью определяется заданием $m \times m = m^2$ вероятностей перехода

$$P\{E_{k+1} | E_k\} = P\{x_k\}$$

(если переход не разрешен, то соответствующая вероятность равна нулю).

(Важное замечание: если вероятности перехода не меняются с течением времени, то говорят, что рассматриваемый процесс — стационарный. Но все последующее может быть применено также и к общему случаю нестационарных процессов.)

Немедленное следствие предыдущего определения состоит в том, что априорная вероятность последовательности состояний $E = (E_0, E_1, \ldots, E_K)$, соответствующей последовательности переходов $x = (x_0, x_1, \ldots, x_{K-1})$, равна

$$P\{E\} = P\{x\} = \prod_{k=0}^{K-1} P\{x_k\} = \prod_{k=0}^{K-1} P\{E_{k+1} | E_k\}.$$

Марковские процессы составляют общую модель, хорошо приспособленную к весьма многочисленным приложениям. Можно указать:

— теорию сигнала;

— кодирование и фильтрацию числовых сигналов;

— анализ текстов и распознавание букв.

При передаче сигнала, например, последовательность состояний $E = (E_0, E_1, \ldots, E_K)$ — или, что то же, последовательность переходов $x = (x_0, x_1, \ldots, x_{K-1})$ — может рассматриваться как кодированная форма информации, подлежащей передаче. Так как переход в канале передачи вызывает возмущения (шумы), то полученный

сигнал не совпадает точно с последовательностью x, но является последовательностью $z=(z_0, z_1, \ldots, z_{K-1})$ наблюдений (каждое наблюдение соответствует некоторому переходу передаваемого процесса). Статистические характеристики шума (мы предполагаем, что он белый, т. е. лишен памяти) предполагаются известными с помощью задания априори вероятностей $P\{z_k|x_k\}$ (вероятность того, что в момент времени k наблюдается z_k при условии, что имел место переход x_k).

(Важное замечание: вместо того чтобы передавать непосредственно последовательность x_k по данному каналу, можно — более общим образом — передавать некоторую функцию $y_k=f(x_k)$. Статистические характеристики шума будут тогда резюмированы заданием априори вероятностей $P\{z_k|y_k\}$.)

Задача, которую мы себе ставим (задача фильтрации), состоит тогда в том, чтобы восстановить (оценить) пропущенную информацию, т. е. восстановить последовательность $x=(x_0, x_1, \ldots, x_{K-1})$, исходя из последовательности наблюдений $z=(z_0, z_1, \ldots, z_{K-1})$, которая предполагается известной.

Можно рассматривать различные критерии оценки. Один из наиболее распространенных и часто применяемых состоит в отыскивании среди всех возможных последовательностей переходов $x=(x_0, x_1, \ldots, x_{K-1})$ последовательности x^*, максимизирующей апостериорную вероятность $P\{x^*|z\}$ (*критерий Байеса*).

Вследствие формулы Байеса

$$P\{x\,|\,z\}=\frac{P\{x,\,z\}}{P\{z\}}=\frac{P\{x\}\,P\{z\,|\,x\}}{P\{z\}}$$

это сводится к отысканию x^*, максимизирующего величину $P\{x\}P\{z|x\}$, поскольку z является одним из данных задачи (а именно наблюдением).

Так как шум — белый (память отсутствует), то можно написать

$$P\{z\,|\,x\}=\prod_{k=0}^{K-1}P\{z_k\,|\,x_k\},$$

и так как процесс марковский, то

$$P\{x\}=\prod_{k=0}^{K-1}P\{x_k\}=\prod_{k=0}^{K-1}P\{E_{k+1}\,|\,E_k\}.$$

Следовательно,

$$P\{x\}\,P\{z\,|\,x\}=\prod_{k=0}^{K-1}P\{x_k\}\,P\{z_k\,|\,x_k\}.$$

Напомним, что величины $P\{E_{k+1}|E_k\}$, с одной стороны, и $P\{z_k|x_k\}$, с другой стороны, являются исходными данными задачи.

Чтобы показать, как эта задача может быть решена средствами динамического программирования, рассмотрим следующий (секвенциальный) граф G:

— каждому моменту времени $k=0, 1, \ldots, K$ и каждому состоянию $E\in\mathscr{E}$ соответствует вершина графа G, обозначаемая (E, k);

— каждому моменту времени $k=0, 1, \ldots, K-1$ и каждому возможному переходу $x_k=(E, E')$ из X соответствует некоторая дуга в G между вершинами (E, k) и $(E', k+1)$. Припишем этой дуге длину, равную

$$\ln P\{x_k\}+\ln P\{z_k|x_k\}.$$

Мы видим, что рассматриваемая задача сводится к отысканию длиннейшего пути в этом графе между начальным состоянием E_0 в момент времени 0 (что является, вообще говоря, исходными данными задачи) и множеством $\mathscr{E}$ состояний в момент времени k.

На рис. 3*б* представлен граф G, соответствующий марковскому процессу с четырьмя состояниями, множество X возможных переходов которого задано графом на рис. 3*а*.

Заметим, что длина дуги, соответствующей переходу x_k, зависит на деле и от наблюдений z_k при посредничестве члена $\ln P\{z_k|x_k\}$.

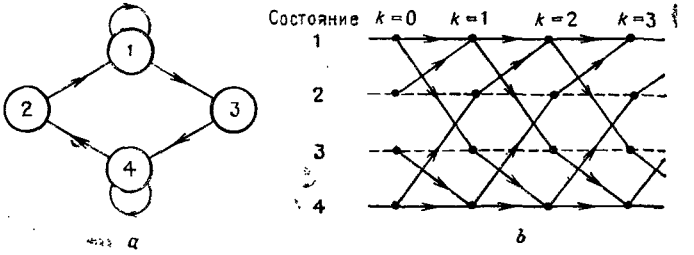


Рис. 3. а. Пример марковского процесса с четырьмя состояниями: граф переходов. b. Граф G, соответствующий решению задачи фильтрации с помощью динамического программирования

Если шум гауссов, то вычисление этой величины особенно просто, поскольку она пропорциональна $(z_k-x_k)^2$.

Особенно важное приложение всего предыдущего к телекоммуникациям касается сверточного кодирования и декодирования числовых сигналов с помощью алгоритма Витерби (см. [53, 18]).

СПИСОК ЛИТЕРАТУРЫ

1. Aris R. (1964), *Discrete Dynamic Programming*, Blaisdell, New York.
2. Bellman R. (1956), Dynamic Programming and Lagrange Multipliers, *Proc. Nat. Acad. Sc.*, USA, vol. 42, p. 767-769.
3. Bellman R. (1958), On a Routing Problem, *Quart. Appl. Math.* 16, p. 87-90.
4. Bellman R. E., Dreyfus S. E. (1962), *Applied Dynamic Programming*, Princeton University Press, Princeton.
5. Bellman R., Kalaba R. (1960), On the k^{th} best policies, *Journal S.I.A.M* 8, 4, p. 582-588.
6. Bertsekas D. P. (1975), Convergence of Discretization Procedures in Dynamic Programming, *IEEE Trans. Automatic Control* 20, p. 415-419.
7. Casti J., Richardson M., Larson R. (1973), Dynamic Programming and Parallel Computers, *J. Optimization Theory and Appl.* 12, p. 423-438.
8. Chevalier A. (1977), *La programmation dynamique*, Dunod, Paris.
9. Christofides N., Mingozzi A., Toth P. (1979), State-Space Relaxations for Combinatorial Problems, *Imperial College, London, Internal Report IC-OR* 79-09.

10. Christofides N., Mingozzi A., Toth P. (1981), State-Space relaxation procedures for the computation of bounds to routing problems, *Networks*.
11. Dano S. (1975), *Nonlinear and Dynamic Programming*, Springer-Verlag.
12. Denardo E. V. (1981), *Dynamic Programming: Theory and Applications*, Prentice Hall.
13. Dijkstra E. W. (1959), A note on two problems in connexion with graphs, *Numerische Mathematik* 1, p. 269-271.
14. Dreyfus S. E. (1975), Computational aspects of dynamic programming, *Operations Research* 5, p. 409-415.
15. Dreyfus S. E., Law A. M. (1977), *The Art and Theory of Dynamic Programming*, Academic Press, New York.
16. Everett H. (1963), Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resoures, *Operations Research* 11, p. 399-417.
17. Fan L. T., Wang C. S. (1964), *The Discrete Maximum Principle: a Study of Multistage Systems Optimization*, John Wiley, New York.
18. Forney G. D. (1973), The Viterbi Algorithm, *Proceedings IEEE*, vol. 61, n° 3, p. 268-278.
19. Garey M. R., Johnson D. S. (1979), *Computers and Intractability: a guide to the theory of NP-completeness*, W. H. Freeman & Co., San Francisco.
20. Gilmore P. C., Gomory R. E. (1966), The theory and computation of knapsack functions, *Operations Research* 13, p. 1045-1074.
21. Gondran M. (1979), Des algorithmes linéaires pour les problèmes de partition, de recouvrement et de couplage dans les hypergraphes d'intervalles, *R.A.I.R.O.*, vol. 13, p. 13-21.
22. Gondran M., Minoux M. (1979), *Graphes et Algorithmes*, Collection Direction Et. Rech. EDF, Eyrolles, Paris.
23*. Hadley G. (1964), *Nonlinear and Dynamic Programming*, Addison Wesley Reading, Mass.
24. Hart P., Nilsson N., Raphael B. A. (1968), A formal basis for the heuristic determination of minimum cost paths, *I.E.E.E. Trans. Syst. Sc. Cybernetics* 2, 4, p. 100-107.
25*. Howard R. A. (1960), *Dynamic Programming and Markov Processes*, John Wiley & Sons, New York.
26. Howard R. A. (1971), *Dynamic Probabilistic Systems*, John Wiley & Sons, New-York, (2 tomes).
27. Jacobson D. H., Mayne D. O. (1970), *Differential Dynamic Programming*, American Elsevier Co., New York.
28. Karp R. M., Held M. (1967), Finite State Processes and Dynamic Programming, *S.I.A.M. J. Appl. Math.* 15, p. 693-718.
29. Kaufmann A., Cruon R. (1965), *La Programmation Dynamique. Gestion scientifique séquentielle*, Dunod, Paris.
30. Larson R. E. (1968), *State Increment Dynamic Programming*, American Elsevier Co., New York.
31. Laurière J. L. (1979), *Éléments de Programmation Dynamique*, Gauthier-Villars, Paris.
32. Marsten R. E., Morin T. L. (1976), MMDP: A computer code for solving multi-constraint knapsack problems in integer variables, *Operations Research Center, MIT*, Cambridge, Mass.
33. Marsten R. E., Morin T. L. (1978), A Hybrid Approach to Discrete Mathematical Programming, *Mathematical Programming* 14, p. 21-40.
34. Martelli A. (1976), An application of heuristic search methods to edge and contour detection, *Comm. ACM*, vol. 19, n° 2, p. 73-83.
35. Martelli A. (1977), On the Complexity of Admissible Search Algorithms, *Artifical Intelligence* 8, p. 1-13.
36. Masse P. (1944), Application des probabilités en chaîne à l'hydrologie statistique et au jeu des réservoirs, *Comm. Société de Statistique*, 21 juin 1944.
37. Masse P. (1959), *Le choix des investissements*, Dunod, Paris.

38. Mitten L. G. (1964), Composition Principles for Synthesis of Optimal Multistage Processes, *Operations Research* 12, p. 610-619.
39. Mitten L. G., Warburton A. R. (1973), Implicit Enumeration Procedures, *Working Paper n° 251, Faculty of Commerce and Business Administration, University of British Columbia*, Vancouver, Canada.
40. Moore E. F. (1957), The shortest path through a maze, *Proc. Int. Symp. on Theory of Swifching*, Part II, April 2-5, 1957, p. 285-292. *The Annals of the Computation Laboratory of Harvard*, Harvard University Press, 1959.
41. Morin T. L. (1977), Monotonicity and the Principle of Optimality, *J. Math. Anal. and Appl.*
42. Morin T. L. (1979), Computational Advances in Dynamic Programming, in *Dynamic Programming and Its Applications*, (M. L. Puterman), Academic Press, New York.
43. Morin T. L., Esogbue A. M. O. (1974), The Imbedded State Space Approach to Reducing Dimensionality in Dynamic Programs of Higher Dimensions, *J. Math. Anal. and Appl.* 48, p. 801-810.
44. Morin T. L., Marsten R. E. (1976a), An Algorithm for Nonlinear Knapsack Problems, *Management Science*, vol. 22, n° 10, p. 1147-1158.
45. Morin T. L., Marsten R. E. (1976b), Branch and Bound Strategies for Dynamic Programming, *Operations Research* 24, p. 611-627.
46. Nemhauser G. L. (1967), *Introduction to Dynamic Programming*, Wiley, New York.
47. Nilsson N. (1971), *Problem-solving Methods in Artificial Intelligence*, McGraw-Hill, New York.
48. Pohl I. (1970), Heuristic search viewed as path-fiding in a graph, *Artificial Intelligence* 1, p. 193-204.
49*. Polya G. (1958), *Les Mathématiques et le raisonnement plausible*, Gauthier-Villars, Paris.
50. Porteus E. (1975), An Informal Look at the Principle of Optimality *Management Science* 21, p. 1346-1348.
51. Proschan F., Bray T. A. (1965), Optimal Redundancy Under Multiple Constraints, *Operations Research* 13, p. 800 814.
52. Rieder U. (1977), Estimates for dynamic programs with lower and upper bounding functions, *Institut für Mathematische Statistik*, Universität Karlsruhe, RFA.
53. Viterbi A. J. (1967), Error bounds for convolutionnal codes and an asymptotically optimum decoding algorithm, *IEEE Trans. Inform. Theory*, vol. IT-13, p. 260-269.
54. Weingartner H. M., Ness D. M. (1967), Methods for the Solution of Multidimensional 0/1 Knapsack Problems, *Operations Research* 15, p. 83-103.
55. White D. J. (1969), *Dynamic Programming*, Holden-Day, USA.
56. Whitet D. J. (1975), Dynamic Programming and Duality in Linear Programming, *Journal Math. Anal. and Appl.* 51, p. 695-704.

ГЛАВА 10

БЕСКОНЕЧНОМЕРНАЯ ОПТИМИЗАЦИЯ И ЕЕ ПРИЛОЖЕНИЯ

§ 1. Введение и примеры

Предыдущие главы были посвящены исключительно конечномерным задачам оптимизации. Тем не менее, как мы сейчас увидим, существуют многочисленные задачи оптимизации, в которых неизвестное уже не является вектором $x=(x_1, \ldots, x_n)^T$ в $\mathbf{R}^n$, а представляет собой, например, функцию $u(t)$ вещественного переменного t на некотором интервале $[a, b]$.

Так как график функции u определяется бесконечным семейством пар $[t, u(t)]$, то мы говорим, что речь идет о задаче оптимизации в бесконечной размерности, или о бесконечномерной задаче оптимизации. Можно заметить, что такая задача может быть поставлена в общем виде следующим образом.

Пусть даны: a) векторное пространство V (бесконечной размерности) и b) функционал J на V, т. е. отображение V в $\mathbf{R}$, которое элементу $u \in V$ сопоставляет $J(u) \in \mathbf{R}$; нужно определить такой $\bar{u} \in V$, что

$$J(\bar{u}) \leqslant J(u) \quad \forall u \in V \text{ (оптимизация без ограничений)},$$

или же такой $\bar{u} \in V$, что

$$J(\bar{u}) \leqslant J(u) \quad \forall u \in U \subset V \text{ (оптимизация при наличии ограничений)}.$$

Мы убеждаемся очевидным образом, что для того, чтобы утверждать существование (а равным образом и единственность) решения u, и для того, чтобы иметь возможность определить это решение с помощью вычислений, необходим какой-то минимум предположений о структуре пространства V (и подмножества $U \subset V$ в случае оптимизации при наличии ограничений). Таким образом, мы приходим к изучению банаховых пространств и особенно гильбертовых пространств.

Но прежде всего интересно рассмотреть некоторые характерные примеры бесконечномерной оптимизации.

1.1. Пример 1. Вариационное исчисление. Типичная задача вариационного исчисления состоит в определении функции $u(t)$ вещественного переменного t на $[a, b]$, минимизирующей функционал вида

$$J(u) = \int_a^b f(u(t), \dot{u}(t), t)\, dt,$$

где $u(a)=\alpha$ и $u(b)=\beta$ заданы и где $\dot{u}(t)=du/dt$. Характерным примером является задача о траектории светового луча (в плоской

среде с меняющимся при изменении абсциссы t показателем (индексом) преломления $n(t)$) между двумя точками: $A[a, u(a)]$ и $B[b, u(b)]$.

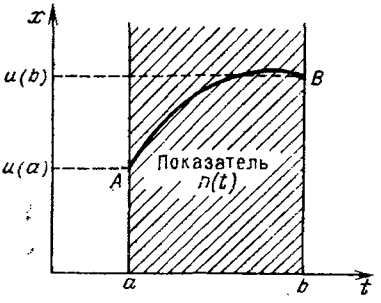


Рис. 1. Траектория светового луча в среде с переменным показателем преломления

Известно, что (в этом состоит *принцип Ферма*) искомая траектория — это траектория, минимизирующая интеграл $\int_A^B n(s)\,ds$, где s — натуральный параметр.

Так как $ds = \sqrt{1+\dot{u}^2}\,dt$ и так как n (показатель преломления) зависит только от t, то мы видим, что рассматриваемая задача сводится к определению функции $u(t)$, минимизирующей функционал

$$J(u) = \int_a^b n(t)\sqrt{1+\dot{u}^2}\,dt.$$

Очевидно, что для того, чтобы эта задача имела смысл, нужно уточнить класс функций, в котором ищется минимум. Например, будем искать u в $C^1[a, b]$ — векторном пространстве непрерывных функций с непрерывными первыми производными на $[a, b]$, причем должны выполняться соотношения $u(a)=\alpha$ и $u(b)=\beta$.

Другая классическая задача вариационного вычисления состоит в определении кратчайшего пути (геодезической линии) между двумя точками A и B на поверхности, определенной (например, в $\mathbf{R}^3$) уравнением вида

$$g(x_1, x_2, x_3) = 0.$$

Эта задача сводится к отысканию кривой $x(t)=(x_1(t), x_2(t), x_3(t))$, так параметризованной с помощью переменного $t \in [0, 1]$, что

$(x_1(0), x_2(0), x_3(0))$ суть координаты точки A,

$(x_1(1), x_2(1), x_3(1))$ суть координаты точки B,

удовлетворяющей $\forall t \in [0, 1]$ условию $g(x_1(t), x_2(t), x_3(t))=0$ и доставляющей минимум интегралу

$$J(x) = \int_0^1 \sqrt{\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2}\,dt.$$

где

$$\dot{x}_1 = dx_1/dt, \quad \dot{x}_2 = dx_2/dt, \quad \dot{x}_3 = dx_3/dt.$$

В отличие от предыдущей задачи здесь речь идет о задаче оптимизации с ограничениями. Решение x можно искать, например, в векторном пространстве непрерывных функций с непрерывными первыми производными на $[0, 1]$ (со значениями в $\mathbf{R}^3$).

1.2. Пример 2. Оптимальное управление системами, определяемыми дифференциальными уравнениями. Рассмотрим систему, эволюционирующую с течением времени. В каждый момент времени t система находится в некотором состоянии, которое определяется заданием компонент $[x_1(t), \ldots, x_n(t)]$ вектора $x(t)$, называемого вектором состояния.

Рассмотрим пример прилунения пространственного транспортного средства. Момент начала маневра прилунения примем за начало отсчета времени. В момент $t = 0$ пространственное транспортное средство имеет некоторую высоту $z(0) = z_0$ и некоторую скорость $v(0) = v_0$ $(v_0 < 0)$ (рис. 2).

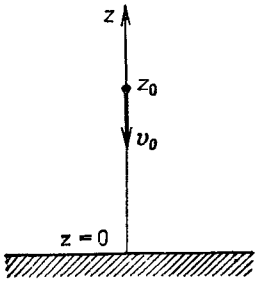


Рис. 2

Если мы допустим для упрощения, что бокового движения нет, то состояние пространственного транспортного средства определяется в каждый момент времени t парой высота — скорость. Поэтому возьмем в качестве вектора состояния $x(t) = [z(t), v(t)]$.

Если начиная с момента $t = 0$ предоставить транспортное средство самому себе, т. е. не подвергать его никакому управлению, то оно будет раздавлено при ударе о поверхность Луны. Для пилота управление состоит в том, чтобы усилить истечение (положительное) из реактивного двигателя так, чтобы достичь высоты 0 с нулевой скоростью.

Пусть $p(t)$ — сила реактивной тяги, выбранная пилотом в момент времени t. Обозначая через M массу транспортного средства (которая предполагается не зависящей от времени), а через g — ускорение свободного падения на Луне (которое предполагается не зависящим от высоты), мы можем записать уравнение динамики в виде

$$M\, dv/dt = -Mg + p(t).$$

Мы видим, таким образом, что эволюция состояния системы управляется системой дифференциальных уравнений вида

$$\dot{z} = dz/dt = v,$$

$$\dot{v} = \frac{dv}{dt} = -g + \frac{p(t)}{M} = u(t)$$

или еще, в очевидных матричных обозначениях,

$$\dot{x} = Ax + Bu,$$

где $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

с начальными условиями

$$z(0) = z_0,$$
$$v(0) = v_0.$$

Следовательно, речь идет об определении такой функции управления $u(t)$ в каждый момент времени t, которая позволила бы привести транспортное средство в состояние $x = [0, 0]$, исходя из состояния $x(0) = [z_0, v_0]$.

Очевидно, что для этого существует бесконечно много способов действия. Можно дождаться последнего момента (транспортное средство оказывается в нескольких метрах над лунной почвой с большой скоростью), а затем дать команду — очень большой реактивной тяги в течение очень короткого времени. Можно, напротив (и это предпочтительнее с точки зрения удобства пассажиров!), провозгласить «мягкость», запимаясь непрерывной реактивной тягой двигателя.

Сопоставляя каждой возможной траектории (в пространстве состояний) некоторый критерий качества, мы можем заняться поисками наилучшей с точки зрения данного критерия траектории, или, что равносильно, отысканием наилучшего управления $u(t)$. Говорят, что целью решения задачи является *оптимальное управление*.

В задаче о пространственном транспортном средстве реалистический критерий качества состоит в том, чтобы привести систему в состояние $z = 0$, $v = 0$ с наименьшим расходом топлива. Если допустить, что количество топлива, выбрасываемое в единицу времени, пропорционально силе тяги, развиваемой реактивным двигателем, то критерием минимизации оказывается функционал

$$J(u) = \int_0^T [g + u(t)]\, dt,$$

где T — момент, в который достигается конечное состояние $z = 0$, $v = 0$. Очевидно, что неизвестная функция $u(t)$ подчиняется тому условию, что решение $x(t)$ дифференциального уравнения

$$\dot{x} = Ax + Bu$$

с начальным условием $x(0) = [z_0, v_0]$ удовлетворяет соотношению $x(T) = [0, 0]$.

Задача о пространственном транспортном средстве в этой упрощенной форме является одним из очень частных примеров задачи оптимального управления. Действительно, заметим, что дифференциальное уравнение, управляющее эволюцией системы во времени, линейно и инвариантно во времени (т. е. его коэффициенты от времени не зависят). Очевидно, что если в этом примере снять гипотезу, что ускорение свободного падения не зависит от высоты z, то полученная система дифференциальных уравнений оказывается нелинейной:

$$\dot{z} = 0, \quad \dot{v} = -g(z) + \frac{p}{M},$$

и задача становится существенно более сложной.

В общем виде задача оптимального управления может быть поставлена следующим образом.

Пусть $x(t)=[x_1(t), \ldots, x_n(t)]$ — вектор состояния, описывающий систему в момент времени t. Эволюция системы во времени управляется дифференциальным уравнением (называемым *уравнением состояния*), которое имеет вид

$$\dot{x}=f(x, u, t).$$

Здесь $u(t)=[u_1(t), \ldots, u_m(t)]$ составляют различные управления, которые позволяют воздействовать на эволюцию системы (в предыдущем примере имелась только одна компонента — сила тяги реактивного двигателя), и вектор $u(t)$ называется *вектором управления* данной системы.

Следовательно, функция f, вообще говоря, является функцией как состояния x, так и применяемых к системе управлений и времени. Если переменное t не участвует явно в функции f, то говорят, что речь идет об *инвариантной системе*. Если функция f линейна относительно вектора $[x, u]$, то говорят, что рассматриваемая система *линейна*. В противном случае она называется *нелинейной*.

Тогда можно поставить задачи различных типов.

Задача 1. Речь идет о задаче управления на фиксированном промежутке времени $[t_0, t_1]$.

Заданы начальное состояние $x(t_0)$ и конечное состояние $x(t_1)$. Нужно найти управление $u(t)$ на интервале $[t_0, t_1]$, позволяющее привести систему из состояния $x(t_0)$ в состояние $x(t_1)$ и минимизирующее критерий качества вида

$$J(u)=\int_{t_0}^{t_1} g(x, u, t)\,dt.$$

Задача 2. Вторая задача, очень близкая к предыдущей, состоит в том, что задано начальное состояние $x(t_0)$, но конечное состояние не задано. Однако начальный момент t_0 и конечный момент t_1 известны. Речь идет тогда об определении управления $u(t)$, минимизирующего вышеприведенный критерий J. Технические приемы решения задач 1 и 2 очень похожи.

Задача 3. Начальное состояние $x(t_0)$ и конечное состояние $x(t_1)$ заданы, но момент t_1 достижения конечного состояния не задан. Задача быстродействия состоит тогда в том, чтобы найти управление $u(t)$, проводящее данную систему из начального состояния в конечное за кратчайшее время.

1.3. Системы, описываемые эллиптическими уравнениями с частными производными. Пусть в $\mathbf{R}^n$ задана открытая область Ω с границей Γ. Предположим, что Γ определена достаточно регулярной гиперповерхностью (непрерывной и с непрерывной нормалью, по крайней мере кусочно). Обозначим через $L^2(\Omega)$ векторное (гиль-

бертово) пространство функций на Ω с суммируемым квадратом, т. е.

$$f \in L^2(\Omega) \Leftrightarrow \int_\Omega [f(x)]^2\, dx < +\infty$$

(интеграл понимается в смысле Лебега).

Поставим себе задачу найти функцию u, определенную в Ω и удовлетворяющую уравнению с частными производными

$$\begin{aligned} -\Delta u &= f \quad \text{в} \quad \Omega, \\ u &= 0 \quad \text{на} \quad \Gamma, \end{aligned}$$

где функция $f \in L^2(\Omega)$ задана и где $\Delta u = \sum_i \frac{\partial^2 u}{\partial x_i^2}$ есть лапласиан от u (*задача Дирихле*).

Эта задача встречается, например, в электростатике, когда мы отыскиваем потенциал, порожденный объемным распределением зарядов f внутри заземленной замкнутой металлической поверхности Γ (потенциал на Γ равен нулю).

Мы сейчас увидим, что эта задача допускает вариационную формулировку, т. е. эквивалентна задаче минимизации некоторого функционала.

Введем для этого векторное пространство (*Соболева*)

$$H^1(\Omega) = \left\{ v \mid v \in L^2(\Omega);\ \frac{\partial v}{\partial x_i} \in L^2(\Omega),\quad i = 1, \ldots, n \right\}.$$

Более общим образом, для целого $p \geqslant 1$ пространство $H^p(\Omega)$ есть векторное пространство функций с суммируемым квадратом на Ω, частные производные которых порядков $1, 2, \ldots, p$ также являются функциями с суммируемым квадратом на Ω (производные понимаются здесь в смысле обобщенных функций).

С другой стороны, обозначим через $H_0^1(\Omega)$ подпространство таких функций u из $H^1(\Omega)$, что $u = 0$ на Γ.

Напомним *формулу Грина*

$$\left.\begin{aligned} u &\in H^2(\Omega) \\ v &\in H^1(\Omega) \end{aligned}\right\} \Rightarrow -\int_\Omega (\Delta u)\, v\, dx = \int_\Omega (\operatorname{grad} u \operatorname{grad} v)\, dx - \int_\Gamma \frac{\partial u}{\partial \nu}\, v\, d\sigma,$$

где ν — единичный вектор, нормальный к Γ (направленный во внешность Ω), $\partial u/\partial \nu = \nu \operatorname{grad} u$ (скалярное произведение) есть нормальная производная функции u на Γ, а $d\sigma$ есть лебегова мера на Γ (частные производные берутся в смысле обобщенных функций).

Умножая уравнение $-\Delta u = f$ на некоторую функцию $v \in H_0^1(\Omega)$ («пробную» функцию) и интегрируя по Ω, получаем

$$-\int_\Omega (\Delta u)\, v\, dx = \int_\Omega f v\, dx.$$

Если предполагать, что $u \in H^2(\Omega)$, то, применяя формулу Грина, получим

$$\int_\Omega (\operatorname{grad} u \operatorname{grad} v)\, dx - \int_\Gamma \frac{\partial u}{\partial \nu} v\, d\sigma = \int_\Omega fv\, dx.$$

Так как v равно нулю на Γ (по определению $H_0^1(\Omega)$), то мы видим, что функция u необходимо удовлетворяет соотношению

$$\int_\Omega (\operatorname{grad} u \operatorname{grad} v)\, dx = \int_\Omega fv\, dx \qquad \forall v \in H_0^1(\Omega). \tag{1}$$

Следовательно, исходная задача может быть заменена следующей задачей: пусть дана функция $f \in L^2(\Omega)$; найти достаточно регулярную функцию $u \in H_0^1(\Omega)$ (достаточна, например, регулярность $u \in H^2(\Omega)$), удовлетворяющую соотношению (1).

Говорят, что мы представили данную задачу в *вариационной форме*.

Действительно, мы убедимся ниже, что в этой форме поставленная задача эквивалентна поиску в пространстве $V = H_0^1(\Omega)$ минимума квадратичного функционала

$$J(v) = \frac{1}{2}\int_\Omega \|\operatorname{grad} v\|^2\, dx - \int_\Omega fv\, dx.$$

Более общим образом, очень многие задачи, касающиеся уравнений с частными производными, могут быть представлены в следующей вариационной форме.

Пусть даны:

a) векторное (гильбертово) пространство V (над $\mathbf{R}$), норма которого обозначается $\|\cdot\|_V$;

b) непрерывная билинейная форма A, сопоставляющая $u \in V$, $v \in V$ число $A(u, v) \in \mathbf{R}$ (предполагается, что A положительно определена, т. е. $\forall v \in V$: $A(v, v) \geqslant 0$ и $A(v, v) = 0 \Rightarrow v = 0$);

c) непрерывная линейная форма L, сопоставляющая $v \in V$ число $L(v) \in \mathbf{R}$ (таким образом, L — элемент пространства V^*, двойственного к V).

Нужно найти такой элемент $u \in V$, что

$$A(u, v) = L(v) \qquad \forall v \in V. \tag{P_V}$$

Заметим, что вариационная формулировка задачи Дирихле есть частный случай задачи (P_V), в котором

$$V \equiv H_0^1(\Omega),$$

$$A(u, v) = \int_\Omega (\operatorname{grad} u \operatorname{grad} v)\, dx,$$

$$L(v) = \int_\Omega fv\, dx.$$

Покажем, что если билинейная форма A симметрична, т. е. если

$$A(u, v) = A(v, u) \qquad \forall u \in V, \quad \forall v \in V,$$

то задача (P_V) эквивалентна поиску минимума в V функционала

$$J(v) = \frac{1}{2} A(v, v) - L(v).$$

Действительно, пусть u — решение задачи (P_V), и рассмотрим некоторый элемент $w \in V$. Можем записать:

$$J(u + w) = \frac{1}{2} A(u + w, u + w) - L(u + w) =$$

$$= \frac{1}{2} A(u, u) + \frac{1}{2} A(w, w) + A(u, w) - L(u) - L(w).$$

Так как u — решение задачи (P_V), то

$$J(u + w) = \frac{1}{2} A(u, u) - L(u) + \frac{1}{2} A(w, w) = J(u) + \frac{1}{2} A(w, w).$$

Поскольку A положительно определена, то отсюда получаем, что

$$J(u + w) \geqslant J(u) \qquad \forall w \in V,$$

что и показывает, что u есть минимум функционала J на V.

Обратно, если элемент u минимизирует функционал $J(v)$ на V, то для любого $v \in V$ и любого $\lambda \in \mathbf{R}$ имеем

$$J(u + \lambda v) \geqslant J(u),$$

что можно записать в виде

$$\frac{1}{2} A(u, u) + \lambda A(u, v) + \frac{\lambda^2}{2} A(v, v) - L(u) - \lambda L(v) \geqslant \frac{1}{2} A(u, u) - L(u),$$

и отсюда для любого фиксированного $v \in V$ получаем, что

$$\frac{\lambda^2}{2} A(v, v) + \lambda [A(u, v) - L(v)] \geqslant 0 \qquad \forall \lambda \in \mathbf{R}.$$

Если бы выполнялось соотношение $A(u, v) - L(v) \neq 0$, то последнее неравенство не могло бы удовлетворяться для всех λ. Следовательно, $A(u, v) = L(v)$ и u — действительно решение задачи (P_V).

Таким образом, в примере с задачей Дирихле задача (P_V) равносильна минимизации функционала

$$J(v) = \frac{1}{2} \int_\Omega \| \operatorname{grad} v \|^2 \, dx - \int_\Omega fv \, dx.$$

Существование и единственность решения задачи (P_V) могут быть доказаны (в этом состоит *теорема Лакса — Милграма*) в предположении (более сильном, чем положительная определенность), называемом эллиптичностью (или коэрцитивностью):

Форма A называется *эллиптической* (или *коэрцитивной*), если существует такая постоянная $\alpha > 0$, что

$$A(v, v) \geqslant \alpha \| v \|_V^2 \qquad \forall v \in V$$

(по этой причине задачи с частными производными, которые могут быть представлены в виде задачи (P_v) с билинейной формой A, удовлетворяющей условию эллиптичности, называются *эллиптическими* задачами).

Другой классический пример эллиптической задачи состоит в следующем. Если положить

$$V \equiv H^1(\Omega),$$

$$A(u, v) = \int_\Omega (\operatorname{grad} u \operatorname{grad} v)\, dx + \int_\Omega uv\, dx,$$

$$L(v) = \int_\Omega fv\, dx,$$

то можно показать (снова с помощью формулы Грина), что полученная задача (P_v) может быть интерпретирована как задача поиска решения u уравнения с частными производными

$$-\Delta u + u = f \text{ в } \Omega \qquad (f \in L^2(\Omega) \text{ задана}),$$

$$\partial u/\partial \nu = 0 \text{ на } \Gamma,$$

где Ω — снова открытая ограниченная область в $\mathbf{R}^n$, а ее граница Γ достаточно регулярна.

Эта задача называется *задачей Неймана*.

Существует много других примеров систем уравнений с частными производными, допускающих вариационную формулировку, которые могут также быть преобразованы в задачи поиска минимума функционала в некотором бесконечном векторном пространстве: задачи упругости, гидромеханики и т. д.

§ 2. Банаховы и гильбертовы пространства

Приведенные выше примеры, и особенно примеры из п. 1.3, показывают, что задачи бесконечномерной оптимизации могут быть и осмыслены, и решены лишь при некоторых предположениях о структуре векторных пространств, в которых эти задачи поставлены.

Поэтому нам нужно ввести в рассмотрение банаховы и гильбертовы (являющиеся частным случаем банаховых) пространства и изучить их общие свойства перед тем, как мы будем иметь возможность поставить основной вопрос: в чем состоит существенное сходство и различие между задачами конечномерной и бесконечномерной оптимизации?

Ответ на этот вопрос облегчается тем, что все вводимые в этой главе понятия могут быть применены к частному случаю конечномерных пространств ($\mathbf{R}^n$ — частный случай гильбертова пространства).

Из этой постоянно приводимой параллели между двумя рассматриваемыми классами задач мы извлечем несколько фундаментальных различий (в бесконечной размерности нужно различать

два понятия сходимости — сильную и слабую), по главным образом будем убеждаться в их очень большом сходстве (алгоритмы решения остаются практически такими же).

2.1. Нормированные векторные пространства. Мы ограничимся исключительно случаем векторных пространств над $\mathbf{R}$, хотя большая часть определений и формулируемых свойств без труда обобщается на векторные пространства над $\mathbf{C}$ ($\mathbf{C}$ — поле комплексных чисел).

Понятие векторного пространства носит исключительно алгебраический характер. Для того чтобы иметь возможность изучать в этих пространствах такие задачи, как сходимость, нужно добавить к структуре векторного пространства топологические понятия (понятия открытого, замкнутого, компактного множества и т. д.), что можно осуществить введением понятия расстояния между элементами векторного пространства.

Определение 1. *Нормированное* векторное пространство над $\mathbf{R}$ есть векторное пространство V, снабженное отображением $V \to \mathbf{R}$, которое каждому элементу $v \in V$ сопоставляет вещественное число $\|v\|_V$, называемое нормой элемента v (обозначаемое просто $\|v\|$, если нет опасности путаницы) и обладающее следующими свойствами:

i) $\|v\| \geqslant 0 \quad \forall v \in V$,

$$\|v\| = 0 \Leftrightarrow v = 0;$$

ii) $\|v + w\| \leqslant \|v\| + \|w\| \quad \forall v \in V, \ \forall w \in V$ (неравенство треугольника);

iii) $\|\lambda v\| = |\lambda| \|v\| \quad \forall v \in V, \ \forall \lambda \in \mathbf{R}$.

Расстояние. В нормированном векторном пространстве V расстояние между двумя элементами v и w определяется как норма их разности.

Сильная сходимость. Исходя из этого определения расстояния, можно ввести обычные топологические понятия (открытые множества, замкнутые множества, окрестности, см. гл. 1, § 2), а также понятие сильной сходимости (эта сходимость называется сильной в противоположность другому типу сходимости, который мы изучим ниже).

Если V — векторное пространство, снабженное нормой $\|\cdot\|$, то говорят, что последовательность $\{v^k\}$ ($v^k \in V$) *сходится* (*сильно*) к $v \in V$ при $k \to \infty$, если выполняется соотношение

$$\lim_{k \to \infty} \|v^k - v\| = 0.$$

2.2. Банаховы пространства. *Последовательностью Коши* в V (относительно сильной топологии) называется такая последовательность $\{v^k\}$, что $\forall \varepsilon > 0$ существует такой $k(\varepsilon)$, для которого выполняется утверждение

$$\forall l \geqslant k(\varepsilon), \quad \forall m \geqslant k(\varepsilon) \Rightarrow \|v^l - v^m\| < \varepsilon.$$

Определение 2. *Банахово* пространство V есть нормированное и *полное* в сильной топологии (связанной с нормой) вектор-

ное пространство, т. е. такое нормированное пространство, что любая последовательность Коши в V сходится (сильно) к элементу пространства V.

Сразу же приведем примеры.

Пример 1. Пространство $\mathbb{R}^n$, снабженное нормой

$$\|x\| = \left[\sum_{i=1}^{n} x_i^2\right]^{1/2},$$

есть нормированное и полное векторное пространство, следовательно, это — банахово пространство.

Пример 2. В $C[a, b]$ — в пространстве непрерывных вещественных функций на вещественном интервале $[a, b]$ — положим

$$\|u\|_{\mathrm{II}} = \max_{t \in [a,b]} |u(t)|,$$

это — норма в $C[a, b]$.

Можно показать, кроме того, что относительно этой нормы пространство $C[a, b]$ полно; следовательно, это — банахово пространство.

Напротив, если снабдить $C[a, b]$ нормой

$$\|u\| = \left[\int_a^b [u(t)]^2\, dt\right]^{1/2},$$

то можно показать, что $C[a, b]$ не полно.

Пример 3. Пусть p — такое вещественное число, что $1 \leqslant p < +\infty$.

Рассмотрим векторное пространство l^p (бесконечных) последовательностей вещественных чисел

$$u = (u_0, u_1, \ldots, u_k, \ldots),$$

удовлетворяющих условию

$$\sum_{k=0}^{\infty} |u_k|^p < +\infty.$$

Можно показать, что

$$\|u\| = \left[\sum_{k=0}^{\infty} |u_k|^p\right]^{1/p}$$

является нормой и что пространство l^p, снабженное этой нормой, полно. Следовательно, это — банахово пространство.

Пример 4. Пусть Ω — открытая область в $\mathbb{R}^n$. Для числа p, удовлетворяющего условию $1 \leqslant p < +\infty$, обозначим через $L^p(\Omega)$ множество таких измеримых функций $u(x) = u(x_1, \ldots, x_n)$, определенных на Ω, что

$$\int_{\Omega} |u(x)|^p\, dx < +\infty$$

(интеграл понимается в смысле Лебега).

Можно показать, что

$$\|u\| = \left[\int_\Omega |u(x)|^p\,dx\right]^{1/p}$$

есть норма и что пространство $L^p(\Omega)$, снабженное этой нормой, полно; следовательно, оно является банаховым пространством.

Замечание. Гильбертовы пространства, которые мы изучаем ниже, являются частным случаем банаховых пространств; важно напомнить, что все свойства банаховых пространств являются также свойствами гильбертовых пространств.

2.3. Сопряженное пространство к нормированному векторному пространству. Пусть V и Y — два нормированных пространства над $\mathbf{R}$. Напомним, что отображение A из V в Y называется *линейным*, если

$$\left.\begin{array}{l}\forall u \in V, \quad \forall v \in V\\ \forall \lambda \in \mathbf{R}, \quad \forall \mu \in \mathbf{R}\end{array}\right\} \Rightarrow A(\lambda u + \mu v) = \lambda A(u) + \mu A(v);$$

A называется *непрерывным* отображением, если

$$v^n \to v \text{ в } V \ (n \to \infty) \Rightarrow A(v^n) \to A(v) \text{ в } Y$$

(сильная сходимость).

Заметим, что в бесконечномерном пространстве линейное отображение не обязательно непрерывно.

Множество непрерывных линейных отображений из V в Y есть векторное пространство, которое мы будем обозначать $\mathscr{L}(V, Y)$.

Легко видеть, что линейное отображение A из V в Y непрерывно тогда и только тогда, когда существует такая постоянная M $(0 < M < +\infty)$, что

$$\|A(v)\|_Y \leqslant M\|v\|_V \qquad \forall v \in V.$$

Действительно, пусть $\varepsilon > 0$. Вследствие непрерывности A в нуле существует такое $\eta > 0$, что

$$\|v\|_V \leqslant \eta \Rightarrow \|A(v)\|_Y < \varepsilon.$$

Для любого $w \in V$ $(w \neq 0)$ имеем

$$\left\|\frac{\eta w}{\|w\|_V}\right\|_V \leqslant \eta,$$

откуда следует, что

$$\left\|A\left(\frac{\eta w}{\|w\|_V}\right)\right\|_Y < \varepsilon.$$

Отсюда вследствие линейности A мы можем записать

$$\|A(w)\|_Y < \frac{\varepsilon}{\eta}\|w\|_V \qquad \forall w \in V, \quad w \neq 0,$$

что и доказывает искомое свойство с $M = \varepsilon/\eta$.

Благодаря этому свойству мы можем связать с каждым непрерывным линейным отображением A: $V \to Y$ (т. е. с каждым

элементом пространства $\mathscr{L}(V, Y)$) вещественное число

$$\|A\|_{\mathscr{L}(V,Y)} = \sup_{\substack{v\in V\\ v\neq 0}} \frac{\|A(v)\|_Y}{\|v\|_V},$$

и легко проверить, что $\|\cdot\|_{\mathscr{L}(V,Y)}$ есть норма в векторном пространстве $\mathscr{L}(V, Y)$.

Кроме того, можно доказать следующее предложение.

Предложение 1. *Пусть V — нормированное векторное пространство, Y — банахово пространство. Тогда $\mathscr{L}(V, Y)$ — банахово пространство.*

Определение 3 (сопряженное к нормированному векторному пространству).

Пусть V — нормированное векторное пространство. *Сопряженным* к пространству V называется пространство непрерывных линейных форм на V, т. е. пространство $\mathscr{L}(V, \mathbf{R})$ непрерывных линейных отображений V в $\mathbf{R}$.

Пространство, сопряженное к нормированному векторному пространству V,— это пространство обозначается V^* — есть банахово пространство (это следует из предложения 1).

Замечание. Если $\mathscr{L}(V, \mathbf{R})$ — *пространство линейных отображений* V в $\mathbf{R}$, непрерывных в сильной топологии, то введенное выше сопряженное пространство должно называться *сильным топологическим сопряженным* к V. Однако для упрощения и в связи с отсутствием двусмысленности мы говорим просто, что V^* — *сопряженное* к V.

Для $L\in V^*$ значение L в точке $v\in V$ обозначается $L(v)$, а норма L в V^* определяется (см. общее определение нормы в $\mathscr{L}(V, Y)$ выше) равенством

$$\|L\|_{V^*} = \sup_{\substack{v\in V\\ v\neq 0}} \frac{|L(v)|}{\|v\|_V}.$$

Приведем некоторые примеры.

Пример 1. Если $V = \mathbf{R}^n$, то хорошо известно, что сопряженное пространство V^* изоморфно $\mathbf{R}^n$.

Пример 2. Для вещественного p $(1\leqslant p<+\infty)$ сопряженное к $V = l^p$ есть $V^* = l^{p'}$, где p и p' связаны соотношением $(1/p)+(1/p')=1$.

Пример 3. Пусть Ω — открытая область в $\mathbf{R}^n$ и пусть $1\leqslant p<+\infty$. Тогда пространство, сопряженное к $V = L^p(\Omega)$, есть $V^* = L^{p'}(\Omega)$, причем $(1/p)+(1/p')=1$.

2.4. Второе сопряженное. Рефлексивные банаховы пространства. Так как пространство V^*, сопряженное к нормированному векторному пространству, есть банахово пространство и, следовательно, нормированное векторное пространство, то можно определить пространство V^{**}, сопряженное к V^*. Очевидно, что V^{**} есть банахово пространство; оно называется *вторым сопряженным* к V (всегда — в смысле сильной топологии).

Пусть $u \in V$ и пусть $\bar{u}$ — такой элемент V^{**}, что $\bar{u}(L) = L(u)$ для всех $L \in V^*$.

Определение 4. Банахово пространство V называется *рефлексивным*, если $u \to \bar{u}$, $u \in V$ есть отображение V на V^{**}.

Примеры. Банаховы пространства $\mathbf{R}^n$ рефлексивны. При $p > 1$ банаховы пространства l^p, $L^p(\Omega)$ рефлексивны.

Гильбертовы пространства, как мы увидим ниже, суть также рефлексивные банаховы пространства.

Заметим, что если V — рефлексивное банахово пространство, то и V^* есть рефлексивное банахово пространство.

Сейчас мы приступим к изучению понятия, которое приводит к глубоким отличиям бесконечномерных задач от конечномерных: это понятие слабой сходимости. В этой связи мы увидим чрезвычайно важную роль, которую играет свойство рефлексивности.

2.5. Слабая сходимость.

Определение 5. Пусть V — нормированное векторное пространство, V^* — его сопряженное пространство. Последовательность $\{v^k\}$ $(v^k \in V)$ называется *сходящейся слабо* (или еще: в смысле слабой топологии) к элементу $v \in V$, если выполняется условие

$$\lim_{k \to \infty} L(v^k) = L(v) \qquad \forall L \in V^*.$$

Обозначение: $v^k \to v$ (слабо).

Предложение 2. *Если* $v^k \to v$ (*сильно*), *т. е. если* $\lim_{k \to \infty} \| v^k - v \|_V = 0$, *то* $v^k \to v$ (*слабо*).

Доказательство. Действительно, возьмем некоторый функционал $L \in V^*$. По определению нормы в V^* имеем

$$| L(v^k - v) | \leqslant \| L \|_{V^*} \| v^k - v \|_V,$$

следовательно, если $v^k \to v$ (сильно), то $L(v^k - v) \to 0$, и вследствие линейности L имеем $L(v^k) \to L(v)$.

Следовательно, $v^k \to v$ (слабо).

Существенным является то обстоятельство, что утверждение, обратное этому предложению, выполняется в конечномерных пространствах (в $\mathbf{R}^n$), но, вообще говоря, не выполняется в бесконечномерных пространствах.

Изучим сначала случай конечномерных пространств.

Предположим, что $v^k \to v$ (слабо) в $\mathbf{R}^n$. Это означает, что для любого вектора w из $(\mathbf{R}^n)^* \equiv \mathbf{R}^n$ имеем

$$w \cdot v^k \to w \cdot v$$

($\cdot$ означает скалярное произведение в $\mathbf{R}^n$).

Беря последовательно в качестве w векторы канонического базиса в $\mathbf{R}^n$, мы видим тогда, что $v^k \to v$ покомпонентно, следовательно, $v^k \to v$ (сильно).

Изучим теперь бесконечномерный случай и приведем примеры того, что из $v^k \to v$ (слабо) не следует, что $v^k \to v$ (сильно).

Пример 1. Рассмотрим случай $V = l^2$, т. е. векторное пространство бесконечных последовательностей $v = (v_0, v_1, \ldots, v_i, \ldots)$

элементов R, снабженное нормой

$$\|v\| = \left[\sum_{i=0}^{\infty} |v_i|^2\right]^{1/2}.$$

и рассмотрим последовательность $\{v^k\}$ с общим членом $v^k = (0, 0, \ldots, 1, 0, \ldots)$ с 1 на k-м месте.

Поскольку пространство l^2 рефлексивно, то элемент w пространства $(l^2)^*$ также может рассматриваться как бесконечная последовательность $w = (w_0, w_1, \ldots, w_i, \ldots)$, причем

$$w(v^k) = \sum_{i=0}^{\infty} w_i v_i^k = w_k.$$

Следовательно, $w(v^k) \to 0$ при $k \to \infty$, и, таким образом $v^k \to 0$ (слабо).

Однако поскольку норма каждого v^k равна 1, то v^k не сходится сильно к 0.

Пример 2. Рассмотрим множество $C[a, b]$ непрерывных вещественных функций на вещественном интервале $[a, b]$, снабженное нормой

$$\|u\| = \max_{t \in [a,b]} |u(t)|;$$

$C[a, b]$ есть банахово пространство (см. п. 2.2, пример 2).

Сильная сходимость u^k к u в этом пространстве определяется условием

$$u^k \to u \quad (\text{сильно}) \leftrightarrow \max_{t \in [a,b]} |u^k(t) - u(t)| \to 0, \quad k \to \infty.$$

Сильная сходимость в $C[a, b]$ эквивалентна поэтому равномерной сходимости соответствующей последовательности функций.

С другой стороны, можно показать, что слабая сходимость u^k к u в $C[a, b]$ эквивалентна следующей паре условий:

i) u^k сходится к u в смысле простой сходимости функций, т. е. сходится поточечно:

$$u^k(t) \to u(t) \quad \forall t;$$

ii) $\exists \alpha > 0$, удовлетворяющее условию $\forall k$,

$$\forall t \in [a, b]: \ |u^k(t)| \leqslant \alpha.$$

Тогда может случиться, что последовательность u^k равномерно ограничена и сходится к некоторой непрерывной же функции поточечно, но неравномерно. Тогда мы видим, что

$$u^k \in C[a, b], \quad u \in C[a, b] \quad \text{и} \quad u^k \to u \quad (\text{слабо}),$$

но u^k не стремится к u сильно. И в этом примере из слабой сходимости не следует сильная.

Подведем итоги: нормированные векторные пространства (и, в частности, банаховы и гильбертовы пространства) могут быть снабжены двумя топологиями: сильной и слабой.

Эти два понятия совпадают в конечномерном случае, но, вообще говоря, различны в бесконечномерном случае.

2.6. Теорема о слабой компактности для рефлексивных банаховых пространств.

Определение 6 (слабая компактность). Пусть V — нормированное векторное пространство.

Подмножество $K \subset V$ называется *слабо компактным*, если из любой бесконечной последовательности $\{v^k\}$ элементов множества K можно выделить подпоследовательность $\{v^{k_i}\}$, слабо сходящуюся к K.

Тогда можно доказать следующее утверждение.

Теорема 1. *Если V — рефлексивное банахово пространство, то из любой ограниченной последовательности элементов пространства V можно выделить подпоследовательность, слабо сходящуюся к некоторому элементу пространства V.*

Этот результат называется также *теоремой о слабой компактности*, так как он немедленно дает

Следствие. *В рефлексивном банаховом пространстве слабо замкнутые ограниченные множества слабо компактны.*

Доказательство. Если K ограничено, то любая (бесконечная) последовательность элементов K допускает выделение подпоследовательности, слабо сходящейся в V. Если K слабо замкнуто, то предел этой подпоследовательности принадлежит K.

Это свойство рефлексивных банаховых пространств исключительно важно, поскольку, как мы увидим, именно оно позволяет установить в бесконечномерных векторных пространствах

— существование (и, возможно, единственность) решений;

— сходимость разрешающих алгоритмов.

Это свойство можно применить, в частности, к гильбертовым пространствам, которые, как мы сейчас увидим, являются частным случаем рефлексивных банаховых пространств.

2.7. Скалярное произведение. Гильбертовы пространства. Пусть V — векторное пространство над $\mathbb{R}$.

Скалярное произведение на V есть симметрическая билинейная форма (называемая также *эрмитовой формой*), являющаяся при этом положительно определенной.

Обозначим скалярное произведение двух элементов $u \in V$ и $v \in V$ через $\langle u, v\rangle$. Тогда имеем

$$\langle u_1 + u_2, v\rangle = \langle u_1, v\rangle + \langle u_2, v\rangle,$$

$$\langle u, v_1 + v_2\rangle = \langle u, v_1\rangle + \langle u, v_2\rangle,$$

$$\langle \lambda u, v\rangle = \langle u, \lambda v\rangle = \lambda\langle u, v\rangle$$

(билинейность),

$$\langle u, v\rangle = \langle v, u\rangle$$

(симметрия),

$$\langle v, v\rangle \geqslant 0,$$

$$\langle v, v\rangle = 0 \Rightarrow v = 0$$

(положительная определенность).

Отметим *неравенство Коши — Буняковского*:

$$|\langle u, v\rangle|^2 \leqslant \langle u, u\rangle\langle v, v\rangle \quad \forall u \in V, \ \forall v \in V.$$

Легко видеть, что векторное пространство V над $\mathbf{R}$, снабженное скалярным произведением $\langle\cdot, \cdot\rangle$, является также нормированным векторным пространством относительно нормы

$$\|v\| = [\langle v, v\rangle]^{1/2}.$$

Действительно, благодаря положительной определенности скалярного произведения имеем

$$\forall v \in V: \ \|v\| \geqslant 0,$$
$$\|v\| = 0 \Rightarrow v = 0.$$

С другой стороны, имеем также $\forall v \in V$, $\forall \lambda \in \mathbf{R}$:

$$\|\lambda v\| = |\langle \lambda v, \lambda v\rangle|^{1/2} = |\lambda|\langle v, v\rangle^{1/2} = |\lambda|\|v\|.$$

Покажем, наконец, справедливость неравенства треугольника. Имеем $\forall u \in V$, $\forall v \in V$:

$$\|u + v\|^2 = \langle u + v, u + v\rangle = \|u\|^2 + 2\langle u, v\rangle + \|v\|^2,$$

откуда, используя неравенство Коши — Буняковского, получаем, что

$$\|u + v\|^2 \leqslant \|u\|^2 + \|v\|^2 + 2\|u\|\|v\| = [\|u\| + \|v\|]^2.$$

Следовательно, определенная выше функция $\|\cdot\|$ действительно является нормой на V (нормой, связанной со скалярным произведением).

Тогда получаем

Определение 7 (гильбертово пространство). Векторное пространство V, снабженное скалярным произведением $\langle\cdot, \cdot\rangle$, называется *гильбертовым пространством*, если V полно относительно топологии, определяемой нормой $\|v\| = [\langle v, v\rangle]^{1/2}$.

Следовательно, гильбертово пространство является банаховым пространством, в котором норма определяется с помощью скалярного произведения.

Пример 1. Пространство $\mathbf{R}^n$, снабженное скалярным произведением

$$\langle u, v\rangle = \sum_{i=1}^{n} u_i v_i,$$

где

$$u = (u_1, \ldots, u_n), \quad v = (v_1, \ldots, v_n).$$

Пример 2. Пространство l^2, снабженное скалярным произведением

$$\langle u, v\rangle = \sum_{i=0}^{\infty} u_i v_i.$$

Пример 3. Пространство $L^2(\Omega)$ (где Ω — открытая область в $\mathbf{R}^n$), снабженное скалярным произведением

$$\int_{\Omega} uv\, dx$$

(интеграл в смысле Лебега).

Пример 4. В пространстве $C[a, b]$ непрерывных вещественных функций на вещественном интервале $[a, b]$ можно, как и в примере 3 выше, определить скалярное произведение

$$\langle u, v\rangle = \int_a^b uv\,dx.$$

Однако $C[a, b]$ не полно относительно топологии, связанной с нормой, определяемой этим скалярным произведением (см. пример 2 п. 2.2): следовательно, это — не гильбертово пространство.

Пример 5. Пространство

$$H^1(\Omega) = \left\{v \mid v \in L^2(\Omega);\ \frac{\partial v}{\partial x_i} \in L^2(\Omega),\quad 1 \leqslant i \leqslant n\right\}$$

(где производные понимаются в смысле обобщенных функций), снабженное скалярным произведением

$$\langle u, v\rangle = \int_\Omega \left(uv + \sum_{i=1}^n \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_i}\right) dx.$$

Можно обобщить это построение на случай пространств $H^p(\Omega)$ для любого целого $p \geqslant 1$ (это — *пространства Соболева*).

Поскольку гильбертово пространство V является также и банаховым пространством и, следовательно, нормированным векторным пространством, то можно определить его сопряженное пространство V^* и его второе сопряженное пространство V^{**}.

Важный вопрос — вопрос рефлексивности. Следующая теорема (называемая *теоремой о представлении линейного функционала*) позволяет ответить на этот вопрос.

Теорема 2 (Рисс). *Пусть V — гильбертово пространство и пусть $L \in V^*$ — непрерывная линейная форма на V. Тогда существует такой однозначно определенный элемент $u_L \in V$, что*

$$L(v) = \langle u_L, v\rangle \qquad \forall v \in V,$$

и при этом

$$\|L\|_{V^*} = \|u_L\|_V.$$

Обратно, любому элементу $u \in V$ можно сопоставить непрерывную линейную форму L_u на V, определенную равенством

$$L_u(v) = \langle u, v\rangle \qquad \forall v \in V.$$

Из этой теоремы сразу следует, что можно отождествить V^* с V, и тогда V^{**} будет отождествлено с V с помощью отображения $u \to \bar{u}$ (см. определение 4, п. 2.4). Таким образом, мы можем сформулировать утверждение фундаментальной важности.

Свойство 1. *Любое гильбертово пространство рефлексивно.*

Следовательно, в гильбертовых пространствах мы можем использовать теорему о слабой компактности (теорему 1 п. 2.6).

Заметим также, что вследствие теоремы Рисса определение слабой сходимости в гильбертовом пространстве V, снабженном скалярным произведением $\langle \cdot, \cdot \rangle$, принимает вид $v^k \to v$ (слабо) $\Leftrightarrow \forall u \in V$: $\langle u, v^k \rangle \to \langle u, v \rangle$.

§ 3. Оптимизация функционалов. Существование минимума. Необходимые условия оптимальности

Пусть дан функционал на банаховом (или гильбертовом) пространстве V, т. е. задано отображение J: $V \to \mathbf{R}$. Мы будем интересоваться минимизацией J на V или на его подмножестве $U \subset V$. Сначала мы изучим условия существования решения этой задачи, вводя понятия сильной и слабой непрерывности и используя теоремы компактности.

Затем мы распространим на функционалы классические понятия дифференцируемости, чтобы вывести из них необходимые условия оптимальности. Наконец, мы дадим примеры приложений к вариационному исчислению и к оптимальному управлению.

3.1. Существование решения. Теорема Вейерштрасса.

Определение 8 (сильная непрерывность и слабая непрерывность).

Функционал J называется *сильно непрерывным* (или просто *непрерывным*, если нет опасности путаницы), если

$$v^k \to v \quad (\text{сильно}) \Rightarrow J(v^k) \to J(v) \quad \text{в } \mathbf{R}.$$

Аналогично, J называется *слабо непрерывным*, если

$$v^k \to v \quad (\text{слабо}) \Rightarrow J(v^k) \to J(v) \quad \text{в } \mathbf{R}.$$

Ясно, что J слабо непрерывен $\Rightarrow$ J сильно непрерывен. (Действительно, если $J(v^k) \to J(v)$ для $v^k \to v$ (слабо), то это тем более верно для $v^k \to v$ (сильно).)

Напомним, что множество $U \subset V$ сильно (соответственно слабо) компактно, если из любой бесконечной последовательности $\{v^k\}$ элементов множества U можно выделить подпоследовательность, которая сильно (соответственно слабо) сходится в U.

Следующая теорема дает достаточные условия того, чтобы задача минимизации

$$\min_{v \in U \subset V} \{J(v)\} \tag{P}$$

имела оптимальное решение в U.

Теорема 3 (Вейерштрасс). *Если подмножество $U \subset V$ сильно (соответственно слабо) компактно и если J сильно (соответственно слабо) непрерывен на U, то задача (P) имеет оптимальное решение в U.*

Доказательство. Пусть $m = \inf\limits_{u \in U} \{J(v)\}$ (m может принимать априори и значение $-\infty$). Это означает, что существует такая последовательность $\{v^k\}_{k \in \mathbf{N}}$ ($v^k \in U$), что $J(v^k) \to m$ ($k \to \infty$).

Так как U сильно (соответственно слабо) компактно, то существует подпоследовательность $\{v^l\}_{l\in L}$ $(L\subset \mathbf{N})$, сильно (соответственно слабо) сходящаяся к $u\in U$.

Так как J сильно (соответственно слабо) непрерывен, то $J(v^l)\to J(u)$, и мы можем записать равенство

$$m=\lim_{k\to\infty} J(v^k)=\lim_{\substack{l\to\infty\\ l\in L}} J(v^l)=J(u).$$

Так как $J(u)>-\infty$, то $m>-\infty$ и $J(u)=m$; следовательно, U — оптимальное решение задачи (P).

Легко видеть, что слабый вариант этой теоремы обобщает теорему 1 гл. 1.

В бесконечномерном случае всегда используется вариант этой теоремы, относящийся к слабой сходимости, так как условие сильной компактности множества U, вообще говоря, гораздо более ограничительно, и его трудно проверять.

С другой стороны, если V — гильбертово (или, в более общем случае, рефлексивное банахово) пространство, то для проверки слабой компактности множества U достаточно — в силу теоремы о слабой компактности (теорема 1, п. 2.6) — удостовериться в том, что U ограничено и слабо замкнуто.

3.2. Полунепрерывность и усиление теоремы Вейерштрасса. Понятие полунепрерывности, которое мы сейчас введем, позволит дать очень часто используемое в приложениях обобщение теоремы 3.

Определение 9. Функционал J называется *сильно* (соответственно *слабо*) *полунепрерывным снизу*, если выполнено следующее условие:

$\forall v\in V$ $\forall\{v^k\}$ — такой последовательности элементов пространства V, что $v^k\to v$ (сильно) (соответственно $v^k\to v$ (слабо)),— выполняется неравенство

$$\liminf J(v^k)\geqslant J(v). \tag{2}$$

Аналогично определяется *полунепрерывность сверху* (*сильная* или *слабая*); в ее определении формула (2) заменяется неравенством

$$\limsup J(v^k)\leqslant J(v). \tag{3}$$

Ясно, что если J полунепрерывен одновременно сверху и снизу, то J непрерывен.

Теорема 3 обобщается тогда сразу же следующим образом.

Теорема 3′. *Пусть $U\subset V$ — сильно (соответственно слабо) компактное множество, J — сильно (соответственно слабо) полунепрерывный снизу функционал на J. Тогда задача (P) имеет оптимальное решение в U.*

Доказательство. Оно протекает шаг за шагом, как в доказательстве теоремы 3. Единственное отличие состоит в том, что теперь $\lim\limits_{\substack{l\to\infty\\ l\in L}} J(v^l)\geqslant J(u)$; следовательно, $m\geqslant J(u)$. Это показывает снова, что $u\in U$ — оптимальное решение задачи (P).

Слабо полунепрерывные снизу функционалы (сокращение: *сл. п. н. сн.*) встречаются достаточно часто, как показывает следующий результат (который мы приводим здесь без доказательства).

Предложение 3. *Если функционал J выпуклый и сильно полунепрерывный снизу, то J слабо полунепрерывен снизу (сл. п. н. сн.).*

Пример. Функционал $J(u)=\|u\|$ сильно непрерывный и выпуклый. Следовательно, он — сл. п. н. сн.

Предложение 7 из п. 3.4 дает другое достаточное условие (в котором также участвует свойство выпуклости) слабой полунепрерывности снизу функционалов на банаховом пространстве.

Из теоремы 3′ вытекает тогда очень важное следствие.

Следствие 1. *Пусть V — гильбертово пространство, $U\subset V$ — сильно замкнутое выпуклое множество, J — сильно полунепрерывный снизу выпуклый функционал.*

Тогда если U ограничено, то J имеет по крайней мере один минимум в U.

Доказательство. Примем следующее свойство без доказательства: если $U\subset V$ — сильно замкнутое выпуклое множество, то U слабо замкнуто (это следует из теоремы Хана — Банаха).

Вследствие предложения 3 функционал J слабо полунепрерывен снизу.

С другой стороны, U ограничено и слабо замкнуто $\Rightarrow U$ слабо компактно. Это завершает доказательство следствия 1.

Замечание. В формулировке только что приведенного следствия 1 условие «U ограничено» может быть заменено условием, что $J(u)\to\infty$ при $\|u\|\to\infty$.

3.3. Производная в смысле Гато. Градиент функционала. Пусть V — нормированное векторное пространство и J — функционал на V.

Определение 10 (G-дифференцируемость). Функционал J называется *имеющим производную по направлению* (или *дифференцируемым в смысле Гато*) в точке $v\in V$ по направлению $\varphi\in V$, если выражение

$$\frac{J(v+\theta\varphi)-J(v)}{\theta}$$

имеет предел при $\theta\to 0$ (в $\mathbf{R}$). Этот предел обозначается $\delta J(v,\varphi)$.

Если $\forall\varphi\in V$ предел $\delta J(v,\varphi)$ существует*), то функционал J называется *дифференцируемым в смысле Гато* (*G-дифференцируемым*) *в точке $v\in V$.*

Определение 11 (градиент). Пусть V — гильбертово пространство, снабженное скалярным произведением $\langle\cdot,\cdot\rangle$. Если функционал J является G-дифференцируемым в точке $v\in V$ и если $\delta J(v,\varphi)$ является непрерывной линейной формой по φ, то тогда (вследствие теоремы о представлении Рисса, см. п. 2.7, теорема 2)

*) Дифференцируемость по Гато часто включает требование линейности выражения $\delta J(v,\varphi)$ по φ, входящее ниже в определение 11.— *Примеч. пер.*

существует такой элемент $J'(v) \in V$, что

$$\delta J(v, \varphi) = \langle J'(v), \varphi \rangle \qquad \forall \varphi \in V.$$

Этот элемент называется *градиентом функционала* J в точке v.

Пример 1. Если $J(v) = J(v_1, \ldots, v_n)$ — дифференцируемая в обычном смысле функция из $\mathbf{R}^n$ в $\mathbf{R}$, то

$$J'(v) = \left(\frac{\partial J}{\partial v_1}, \frac{\partial J}{\partial v_2}, \ldots, \frac{\partial J}{\partial v_n}\right)^T.$$

Действительно, мы и на самом деле получаем $\forall \varphi \in \mathbf{R}^n$:

$$\delta J(v, \varphi) = [J'(v)]^T \varphi$$

(скалярное произведение в $\mathbf{R}^n$).

Пример 2. Пусть $J(v)$ — функционал на $L^2[a, b]$ со значениями в $\mathbf{R}$, определенный соотношением

$$J(v) = \int_a^b [v(x)]^2 \, dx \qquad \forall v \in L^2[a, b].$$

Тогда имеем $\forall \varphi \in L^2[a, b]$:

$$\frac{J(v + 0\varphi) - J(v)}{0} = \int_a^b \varphi [2v + 0\varphi] \, dx \to 2 \int_a^b v\varphi \, dx,$$

и, следовательно, $\delta J(v, \varphi)$ линеен и непрерывен относительно φ, поскольку

$$\delta J(v, \varphi) = \int_a^b 2v\varphi \, dx = \langle 2v, \varphi \rangle.$$

Следовательно, $J'(v) = 2v$.

Пример 3. Пусть теперь $J(v)$ — функционал из $H^1[a, b]$ в $\mathbf{R}$, определенный формулой

$$J(v) = \int_a^b v^2 \, dx \qquad \forall v \in H^1[a, b].$$

Снова имеем

$$\delta J(v, \varphi) = 2 \int_a^b v\varphi \, dx,$$

что, безусловно, является непрерывной линейной формой от φ.

Но поскольку скалярное произведение в $H^1[a, b]$ определено формулой

$$\langle v, w \rangle = \int_a^b \left(vw + \frac{dv}{dx} \frac{dw}{dx}\right) dx$$

(см. пример 5 п. 2.7), то градиент J больше не равен $2v$, как было в предыдущем примере 2.

Этот пример показывает, насколько существенно уточнение того, в каком именно пространстве определены функционалы, для которых мы хотим определить градиент.

Ниже мы найдем другие примеры вычисления производных — в рамках вариационного исчисления (п. 3.6) или оптимального управления (п. 3.7).

Используя G-дифференцируемость, мы можем следующим образом обобщить формулу конечных приращений.

Предложение 4. *Если функционал J является G-дифференцируемым в точке $v+\alpha\varphi$ для любого $\alpha\in[0,1]$ в направлении φ, то существует такое $\theta\in(0,1)$, что*

$$J(v+\varphi)=J(v)+\delta J(v+\theta\varphi,\varphi).$$

Доказательство. Рассмотрим вещественную функцию вещественного переменного α, определенную равенством $f(\alpha)=J(v+\alpha\varphi)$. Из условий предложения следует, что эта функция дифференцируема на $[0,1]$ по α, и ее производная по α равна

$$df/d\alpha=\delta J(v+\alpha\varphi,\varphi).$$

Применяя к f формулу конечных приращений, видим, что существует такое $\theta\in(0,1)$, что

$$f(1)=f(0)+\frac{df}{d\alpha}(\theta),$$

что равносильно равенству

$$J(v+\varphi)=J(v)+\delta J(v+\theta\varphi,\varphi).$$

Определение 12 (двукратная G-дифференцируемость, гессиан.) Функционал $J\colon V\to\mathbf{R}$, дифференцируемый по Гато в каждой точке $v+\theta\psi$, $|\theta|\leqslant\delta$ $(\delta>0)$ в направлении φ $(v,\psi,\varphi\in V)$, называется *дважды дифференцируемым в смысле Гато в точке v в направлениях φ и ψ*, если отношение

$$\frac{\delta J(v+\theta\psi,\varphi)-\delta J(v,\varphi)}{\theta}$$

имеет предел в $\mathbf{R}$ при $\theta\to 0$. Этот предел обозначается $\delta^2J(v,\varphi,\psi)$. Если $\delta^2J(v,\varphi,\psi)$ существует $\forall\varphi\in V$, $\forall\psi\in V$, то функционал J называется *дважды G-дифференцируемым в точке $v\in V$*.

Если, кроме того, V — гильбертово пространство и для данного элемента $v\in V$ отображение $(\varphi,\psi)\to\delta^2J(v,\varphi,\psi)$ непрерывно и линейно по φ и ψ, то существует такой линейный оператор $\mathscr{H}(v)$ из V в V, что

$$\delta^2J(v,\varphi,\psi)=\langle\mathscr{H}(v)\psi,\varphi\rangle;$$

оператор $\mathscr{H}(v)$ называется *гессианом* функционала J в точке v.

С помощью этих новых понятий можно обобщить следующим образом формулу Тейлора.

Предложение 5. *Если функционал J дважды G-дифференцируем в точке $v+\alpha\varphi$ при всех $\alpha\in[0,1]$ в направлениях φ и $\psi=\varphi$,*

то существует такое $\theta \in (0, 1)$, *что*

$$J(v+\varphi) = J(v) + \delta J(v, \varphi) + \frac{1}{2}\delta^2 J(v + \theta\varphi, \varphi, \varphi).$$

Доказательство аналогично доказательству предложения 4.

Определение 13. Вектор $\varphi \in V$ называется *направлением спуска* для функционала J в точке $v \in V$, если существует такое $t > 0$, что

$$J(v + \theta\varphi) < J(v)$$

для всех $\theta \in (0, t)$.

Используя формулу Тейлора (предложение 5), мы видим, что достаточное условие того, что φ является направлением спуска для функционала J в точке v (в предположении двукратной G-дифференцируемости J в некоторой окрестности $O(v)$ точки v и, скажем, справедливости соотношения вида $\|\delta^2 J(v+\theta\varphi, \varphi, \varphi)\| \leqslant C\|\varphi\|^2$ для $v + \theta\varphi \in O(v)$), является условие $\langle J'(v), \varphi\rangle < 0$ *).

В частности, $\varphi = -J'(v)$ есть направление спуска для J в v.

Все предыдущее обобщает свойства, хорошо известные для обычных функций (на $\mathbf{R}^n$).

3.4. G-дифференцируемые выпуклые функционалы. Коэрцитивные выпуклые функционалы.

Определение 14. Пусть $J\colon V \to \mathbf{R}$ — функционал на нормированном векторном пространстве V. Функционал J называется *α-выпуклым* (иначе: *коэрцитивным* или *эллиптическим*), если существует такое вещественное число $\alpha > 0$, что $\forall u \in V$, $\forall v \in V$, $\forall \theta \in [0, 1]$:

$$J[(1-\theta)u + \theta v] \leqslant (1-\theta)J(u) + \theta J(v) - \frac{\alpha}{2}\theta(1-\theta)\|u - v\|^2.$$

Замечание. В предыдущем определении случай $\alpha = 0$ соответствует обычным выпуклым функциям. Поэтому следующий результат применим к обычным выпуклым функциям; для этого достаточно положить $\alpha = 0$.

Предложение 6. *Пусть* $J\colon V \to \mathbf{R}$ — *функционал, являющийся G-дифференцируемым в каждой точке пространства V. Следующие условия эквивалентны*:

i) J *является α-выпуклым*;

ii) $\forall u \in V$, $\forall v \in V$:

$$J(v) \geqslant J(u) + \delta J(u, v-u) + \frac{\alpha}{2}\|v-u\|^2;$$

iii) $\forall u \in V$, $\forall v \in V$:

$$\delta J(u, u-v) - \delta J(v, u-v) \geqslant \alpha\|u-v\|^2.$$

Доказательство.

(i) $\Rightarrow$ (ii).

---

*) С помощью определений 10 и 11 легко убедиться, что направление $\varphi \in V$, удовлетворяющее условию $\langle J'(v), \varphi\rangle < 0$, является направлением спуска в более слабых предположениях — дифференцируемости функционала J по Гато в точке v и существования градиента в точке v.— *Примеч. пер.*

Действительно, из (i) следует $\forall \theta \in [0, 1]$:

$$J(u + \theta(v - u)) \leqslant J(u) + \theta(J(v) - J(u)) - \frac{\alpha}{2}\theta(1 - \theta)\|u - v\|^2,$$

следовательно, при $\theta > 0$

$$\frac{J(u + \theta(v - u)) - J(u)}{\theta} \leqslant J(v) - J(u) - \frac{\alpha}{2}(1 - \theta)\|u - v\|^2,$$

так как J является G-дифференцируемым, то, устремляя θ к 0 справа, получаем, что

$$\delta J(u, v - u) \leqslant J(v) - J(u) - \frac{\alpha}{2}\|u - v\|^2,$$

откуда следует (ii).

(ii) $\Rightarrow$ (i).

Действительно, если выполнено условие (ii), то для любого $\theta \in [0, 1]$ можно записать, полагая $w = u + \theta(v - u)$,

$$J(u) \geqslant J(w) + \delta J(w, u - w) + \frac{\alpha}{2}\|u - w\|^2,$$

$$J(v) \geqslant J(w) + \delta J(w, v - w) + \frac{\alpha}{2}\|v - w\|^2.$$

Так как $u - w = -\theta(v - u)$ и $v - w = (1 - \theta)(v - u)$, то мы видим, что

$$J(u) \geqslant J(w) - \theta\delta J(w, v - u) + \frac{\alpha}{2}\theta^2\|v - u\|^2,$$

$$J(v) \geqslant J(w) + (1 - \theta)\delta J(w, v - u) + \frac{\alpha}{2}(1 - \theta)^2\|v - u\|^2.$$

Умножая первое неравенство на $(1 - \theta)$, второе — на θ и складывая, получим, что

$$(1 - \theta)J(u) + \theta J(v) \geqslant J(w) + \frac{\alpha}{2}\theta(1 - \theta)\|v - u\|^2,$$

что и доказывает, что функционал J является α-выпуклым.

(ii) $\Rightarrow$ (iii).

Действительно, имеем

$$J(v) \geqslant J(u) + \delta J(u, v - u) + \frac{\alpha}{2}\|v - u\|^2,$$

$$J(u) \geqslant J(v) + \delta J(v, u - v) + \frac{\alpha}{2}\|u - v\|^2,$$

отсюда, складывая, получаем, что

$$0 \geqslant \delta J(u, v - u) + \delta J(v, u - v) + \alpha\|u - v\|^2,$$

откуда следует (iii).

(iii) $\Rightarrow$ (ii).

Выберем произвольные u и v и рассмотрим функцию вещественного переменного θ:

$$\psi(\theta) = J(u + \theta(v - u)).$$

Имеем

$$\psi'(\theta_0) = \frac{d\psi}{d\theta}(\theta_0) = \delta J\,(u + \theta_0(v-u), v-u),$$

откуда для любого θ получаем, что

$$\psi'(\theta) - \psi'(0) = \delta J(u + \theta(v-u),\ v-u) - \delta J(u,\ v-u).$$

Полагая $w = u + \theta(v-u)$, имеем для $\theta > 0$

$$\psi'(\theta) - \psi'(0) = \frac{1}{\theta}\,[\delta J\,(w, w-u) - \delta J\,(u, w-u)],$$

откуда, используя (iii), получаем неравенство

$$\psi'(\theta) - \psi'(0) \geqslant \frac{1}{\theta}\,\alpha \| w-u \|^2 = \theta\alpha \| v-u \|^2.$$

Интегрируя по θ, получаем отсюда, что

$$\psi(\theta) \geqslant \psi(0) + \psi'(0)\,\theta + \frac{\alpha}{2} \| v-u \|^2 \theta^2,$$

что при $\theta = 1$ дает соотношение (ii).

Существенное свойство G-дифференцируемых выпуклых функций состоит в следующем.

Предложение 7. *Если функционал J на банаховом пространстве V является выпуклым и G-дифференцируемым и если $\delta J(v, \varphi)$ — непрерывная линейная форма по φ, то J слабо полунепрерывен снизу (сл. п. н. сн.).*

Доказательство. Предположим, что $v^k \to v$ слабо в V при $k \to \infty$, т. е. для любой непрерывной линейной формы $L \in V^*$ имеем $L(v^k - v) \to 0$ при $k \to \infty$. Согласно предложению 6 (в котором полагаем $\alpha = 0$), имеем

$$J(v^k) \geqslant J(v) + \delta J(v, v^k - v).$$

Так как отображение $\varphi \to \delta J(v, \varphi)$ линейно и непрерывно (и поэтому принадлежит V^*), то заключаем отсюда, что $\delta J(v, v^k - v) \to 0$ при $k \to \infty$. Отсюда и из предыдущего неравенства следует, что

$$\liminf_{k \to \infty} J(v^k) \geqslant J(v)$$

и, таким образом, J — сл. п. н. сн.

Заметим, что предложение 7 применимо, в частности, к выпуклым функционалам на гильбертовом пространстве, которые имеют градиент в каждой точке.

3.5. Необходимые условия оптимальности. В случае оптимума без ограничений мы можем сформулировать следующее утверждение.

Теорема 4 (необходимое условие первого порядка). *Пусть J — функционал на банаховом пространстве V, являющийся G-дифференцируемым в точке $v^0 \in V$. Для того чтобы v^0 была точкой оптимума функционала J, необходимо, чтобы выполнялось соотношение*

$$\delta J(v^0, \varphi) = 0 \qquad \forall \varphi \in V.$$

Доказательство. Если v^0 — точка (локального или глобального) минимума функционала J, то для любого $\varphi \in V$ функция $J(v^0+\theta\varphi)$ вещественного переменного θ должна иметь минимум в точке $\theta=0$. Необходимое условие этого состоит в выполнении соотношения

$$\left[\frac{d}{d\theta} J(v^0+\theta\varphi)\right]_{\theta=0} = \delta J(v^0, \varphi) = 0.$$

Точка v^0, удовлетворяющая необходимому условию первого порядка

$$\delta J(v^0, \varphi) = 0 \quad \forall\varphi,$$

называется стационарной точкой.

Теорема 5 (необходимое условие второго порядка). *Пусть J — дважды G-дифференцируемый в точке $v^0 \in V$ функционал на банаховом пространстве V. Для того чтобы v^0 была точкой минимума функционала J, необходимо, чтобы для любого $\varphi \in V$ выполнялись соотношения*

$$\delta J(v^0, \varphi) = 0,$$
$$\delta^2 J(v^0, \varphi, \varphi) \geqslant 0.$$

Доказательство. Оно сразу вытекает из необходимых условий второго порядка для вещественной функции $J(v^0+\theta\varphi)$ вещественного переменного θ для произвольного φ в V.

Интересное обобщение теоремы 4 на случай задач с ограничениями состоит в следующем.

Теорема 6. *Пусть J — функционал на банаховом пространстве V; пусть J является G-дифференцируемым в точке v^0 и пусть $U \subset V$ — выпуклое множество, содержащее v^0. Для того чтобы точка $v^0 \in U$ была точкой минимума функционала J на U, необходимо, чтобы выполнялось условие*

$$\delta J(v^0, v-v^0) \geqslant 0 \qquad \forall v \in U.$$

Доказательство. Пусть $v \in U$ произвольно. Так как U выпукло, $v^0 \in U$ и $v \in U$, то

$$v^0 + \theta(v-v^0) \in U \qquad \forall\theta \in [0, 1].$$

Следовательно, для того чтобы v^0 была точкой минимума для J, необходимо, чтобы для каждого $v \in U$

$$\left[\frac{d}{d0} J(v^0 + \theta(v-v^0))\right]_{\theta=0} = \delta J(v^0, v-v^0) \geqslant 0.$$

Замечание. В случае без ограничений необходимые условия оптимальности выражаются в форме вариационных уравнений вида $\delta J(v^0, \varphi) = 0$ $(\forall\varphi \in V)$, тогда как в случае с ограничениями условия оптимальности выражаются в форме вариационных неравенств

$$\delta J(v^0, v-v^0) \geqslant 0 \qquad \forall v \in U.$$

Несмотря на чрезвычайную простоту их получения, предыдущие результаты составляют основания вариационного исчисления.

Действительно, как мы сейчас увидим, они позволяют вывести необходимые условия оптимальности, известные под названием уравнений Эйлера — Лагранжа.

3.6. Приложения к вариационному исчислению. Уравнения Эйлера — Лагранжа. Вернемся к примеру 1 п. 1.1.

Задача состоит в минимизации интеграла

$$J(u) = \int_a^b f(u(t), \dot{u}(t), t)\, dt, \qquad \dot{u} = \frac{du}{dt},$$

на множестве $V = C^1[a, b]$ непрерывных функций с непрерывной первой производной на вещественном интервале $[a, b]$. Наложим, кроме того, условия $u(a) = \alpha$ и $u(b) = \beta$ (α и β даны).

Предполагая, что f — непрерывная функция всех своих аргументов, непрерывно дифференцируемая по u и $\dot{u}$, вычислим G-дифференциал функционала J в направлении вектора φ, где $\varphi \in C^1[a, b]$ и $\varphi(a) = \varphi(b) = 0$. Имеем

$$\delta J(u, \varphi) = \frac{d}{d\theta}\left[\int_a^b f(u + \theta\varphi, \dot{u} + 0\dot{\varphi}, t)\, dt\right]_{\theta=0},$$

или, с учетом сделанных выше предположений,

$$\delta J(u, \varphi) = \int_a^b \frac{\partial f}{\partial u}(u, \dot{u}, t)\, \varphi(t)\, dt + \int_a^b \frac{\partial f}{\partial \dot{u}}(u, \dot{u}, t)\, \dot{\varphi}(t)\, dt.$$

Мы видим, что $\delta J(u, \varphi)$ — непрерывная линейная функция φ. Однако так как $V = C^1[a, b]$ не является гильбертовым пространством, то определить градиент функционала J в этом пространстве мы не можем.

Вследствие теоремы 4, необходимое условие того, что точка $\bar{u}$ является точкой оптимума функционала J, состоит в выполнении условий

$$\delta J(\bar{u}, \varphi) = 0 \quad \forall \varphi \in C^1[a, b], \quad \varphi(a) = \varphi(b) = 0.$$

Иначе говоря, для любой $\varphi \in C^1[a, b]$ с $\varphi(a) = \varphi(b) = 0$ должно выполняться соотношение

$$\int_a^b [g(t)\varphi(t) + h(t)\dot{\varphi}(t)]\, dt = 0,$$

где

$$g(t) = \frac{\partial f}{\partial u}(u, \dot{u}, t), \qquad h(t) = \frac{\partial f}{\partial \dot{u}}(u, \dot{u}, t).$$

Формулируемый ниже результат позволяет непосредственно выразить необходимое условие оптимальности в теоремах самой функции f и ее частных производных.

Предложение 8. *Если $y(t)$ и $h(t)$ — непрерывные функции на $[a, b]$ и если для любой $\varphi \in C^1[a, b]$ с $\varphi(a) = \varphi(b) = 0$*

выполняется соотношение

$$\int_a^b [g(t)\varphi(t) + h(t)\dot{\varphi}(t)]\,dt = 0, \tag{4}$$

то производная $dh(t)/dt$ *существует, причем*

$$g(t) - \frac{dh(t)}{dt} \equiv 0 \qquad \forall t \in [a, b].$$

Доказательство.

a) Покажем сначала, что если $q(t)$ — непрерывная функция на $[a, b]$ и если для любой функции $\varphi \in C^1[a, b]$ с $\varphi(a) = \varphi(b) = 0$ выполняется соотношение

$$\int_a^b q(t)\dot{\varphi}(t)\,dt = 0,$$

то q — постоянная функция на $[a, b]$.

Определим для доказательства этого утверждения постоянную c так, чтобы выполнялось равенство $\int_a^b [q(t) - c]\,dt = 0$. Тогда, полагая

$$\varphi(t) = \int_a^t [q(\tau) - c]\,d\tau,$$

видим, что вследствие соотношения $\dot{\varphi}(t) = q(t) - c$ должно выполняться равенство

$$0 = \int_a^b q(t)[q(t) - c]\,dt = \int_a^b [q(t) - c]^2\,dt + \int_a^b c\,[q(t) - c]\,dt.$$

Но

$$\int_a^b c\,[q(t) - c]\,dt = c\int_a^b \dot{\varphi}(t)\,dt = c\,[\varphi(b) - \varphi(a)] = 0.$$

Отсюда следует, что необходимо должно выполняться соотношение

$$\int_a^b [q(t) - c]^2\,dt = 0$$

и так как $q(t)$ непрерывна, то должно выполняться равенство $q(t) - c = 0$ на $[a, b]$, откуда и следует доказываемое утверждение.

b) Докажем теперь предложение 8.

Положим

$$G(t) = \int_a^t g(\tau)\,d\tau.$$

Интегрируя по частям, получаем, что

$$0 = [G(t)\varphi(t)]_a^b = \int_a^b G(t)\dot{\varphi}(t)\,dt + \int_a^b g(t)\varphi(t)\,dt.$$

Следовательно, $\forall \varphi \in C^1[a, b]$, $\varphi(a) = \varphi(b) = 0$, формула (4) принимает вид

$$\int_a^b [-G(t) + h(t)]\,\dot{\varphi}(t)\,dt = 0.$$

Согласно части а) доказательства, тогда должно выполняться соотношение

$$q(t) = -G(t) + h(t) = c \qquad \text{(где } c \text{ — постоянная) на } [a, b].$$

Следовательно, $h(t) = G(t) + c$.

Но так как $G(t)$ дифференцируема по построению, то и $h(t)$ дифференцируема, причем

$$\dot{h}(t) = g(t) \qquad \forall t \in [a, b],$$

что завершает доказательство предложения 8.

Вследствие предложения 8 из условия $\delta J(\bar{u}, \varphi) = 0$ следует, что

$$\frac{\partial f}{\partial u}(\bar{u}, \dot{\bar{u}}, t) - \frac{d}{dt}\left[\frac{\partial f}{\partial \dot{u}}(\bar{u}, \dot{\bar{u}}, t)\right] = 0.$$

Это необходимое условие оптимальности функционала $J(u) = \int_a^b f(u, \dot{u}, t)\,dt$ и называется *уравнением Эйлера — Лагранжа.*

Можно воспользоваться этим условием, в частности, в задаче о траектории светового луча в среде с постоянным показателем преломления $n(t)$ (см. п. 1.1). Тогда нужно минимизировать интеграл

$$J(u) = \int_a^b \sqrt{1 + (\dot{u})^2}\,dt.$$

Здесь имеем

$$f(u, \dot{u}, t) = \sqrt{1 + (\dot{u})^2},$$

следовательно,

$$\frac{\partial f}{\partial u} = 0, \qquad \frac{\partial f}{\partial \dot{u}} = \frac{-\dot{u}}{\sqrt{1 + (\dot{u})^2}}.$$

Тогда уравнение Эйлера — Лагранжа записывается в виде

$$\frac{d}{dt}\left[\frac{\partial f}{\partial \dot{u}}\right] = 0,$$

что означает, что $\partial f/\partial \dot{u}$ постоянна и потому $\dot{u}$ постоянна. Мы получаем новое доказательство того факта, что траектория светового

луча в однородной среде (с постоянным показателем преломления) есть прямая линия.

3.7. Приложения к оптимальному управлению. Принцип максимума Понтрягина. Рассмотрим задачу оптимального управления, введенную в п. 1.2 (задача 2) в общем виде:

найти функцию $u(t)$ на $[t_0, t_1]$, минимизирующую функционал

$$J(u) = \int_{t_0}^{t_1} g(x(t), u(t), t)\, dt,$$

где $x(t)$ (вектор состояния) зависит от $u(t)$ посредством уравнения состояния:

$$\dot{x}(t) = f(x(t), u(t), t),$$
$$x(t_0) = \xi_0 \quad \text{(данный вектор)}.$$

Обозначим через n размерность вектора состояния $x(t)$, а через m — размерность вектора управления $u(t)$.

Вычислим G-дифференциал функционала J в точке $\bar{u}$, т. е.

$$\delta J(\bar{u}, \varphi) = \lim_{\theta \to 0_+} \frac{J(\bar{u} + \theta\varphi) - J(\bar{u})}{\theta}.$$

Для этого предположим, что отображение, которое сопоставляет элементу u пространства $V \equiv (L^2[t_0, t_1])^m$ решение $x(u(t), t)$ уравнения состояния, (сильно) непрерывно как отображение пространства $(L^2[t_0, t_1])^m$ в $H^1[t_0, t_1]^n$ и G-дифференцируемо по u, т. е. $\forall \varphi \in V$ предел

$$\lim_{\theta \to 0} \frac{x(u + \theta\varphi) - x(u)}{\theta}$$

существует; обозначим его $y(u, t)$ (величина $y(u, t)$ зависит и от u, и от φ). Имеем $y(u(t_0), t_0) = 0$.

Заметим, что $x(u, t)$ удовлетворяет интегральному уравнению

$$x(\bar{u}, t) = x(t_0) + \int_{t_0}^{t} f(x(\bar{u}, t), \bar{u}(t), t)\, dt.$$

Следовательно,

$$x(\bar{u} + \theta\varphi, t) = x(t_0) + \int_{t_0}^{t} f(x(\bar{u} + \theta\varphi, t), \bar{u} + \theta\varphi, t)\, dt,$$

$$\frac{x(\bar{u} + \theta\varphi, t) - x(\bar{u}, t)}{\theta} = \int_{t_0}^{t} \frac{f(x(\bar{u} + \theta\varphi, t), \bar{u} + \theta\varphi, t) - f(x(\bar{u}, t), \bar{u}, t)}{\theta}\, dt.$$

Если $\theta \to 0$ справа, то левая часть стремится к $y(\bar{u}, t)$.

Если мы предположим, что f непрерывно дифференцируема по x и u, то при $\theta \to 0$ справа имеем, кроме того,

$$\frac{f(x(\bar{u}+\theta\varphi), \bar{u}+\theta\varphi, t) - f(x(\bar{u}, t), \bar{u}, t)}{\theta} \to$$

$$\to \frac{\partial f}{\partial x}(x(\bar{u}, t), \bar{u}, t)\, y(\bar{u}, t) + \frac{\partial f}{\partial u}(x(\bar{u}, t), \bar{u}, t)\, \varphi.$$

Следовательно, $y(\bar{u}, t)$ удовлетворяет интегральному уравнению

$$y = \int_{t_0}^{t} \left[\frac{\partial f}{\partial x} y + \frac{\partial f}{\partial u} \varphi\right] dt,$$

где $\partial f/\partial x$ и $\partial f/\partial u$ вычисляются при $x = x(\bar{u}, t)$ и $u = \bar{u}$.

Отсюда следует, что y удовлетворяет системе дифференциальных уравнений

$$\begin{gathered} \dot{y} = \frac{\partial f}{\partial x} y + \frac{\partial f}{\partial u} \varphi, \\ y(t_0) = 0. \end{gathered} \tag{I}$$

Вычислим теперь величину

$$\frac{J(\bar{u}+\theta\varphi) - J(\bar{u})}{\theta} = \int_{t_0}^{t_1} \frac{g(x(\bar{u}+\theta\varphi, t), \bar{u}+\theta\varphi, t) - g(x(\bar{u}, t), \bar{u}, t)}{\theta}\, dt.$$

Если мы предположим, что g непрерывно дифференцируема по переменным x и u, то мы получим, заставляя θ стремиться к нулю справа, что

$$\delta J(\bar{u}, \varphi) = \int_{t_0}^{t_1} \left[\frac{\partial g}{\partial x} y + \frac{\partial g}{\partial u} \varphi\right] dt,$$

где y — решение системы (I) — есть функция переменного φ.

Чтобы выразить $\delta J(\bar{u}, \varphi)$ однозначно как функцию φ, введем вектор $p(t)$, называемый *сопряженным вектором состояния* и определяемый как решение линейной системы дифференциальных уравнений

$$\begin{gathered} -\frac{dp}{dt} = -\dot{p} = \left(\frac{\partial f}{\partial x}\right)^T p + \left(\frac{\partial g}{\partial x}\right)^T, \\ p(t_1) = 0. \end{gathered} \tag{II}$$

Система (II) называется *сопряженной системой*.

Заменяя $\partial g/\partial x$ выражением $-\dot{p}^T - p^T(\partial f/\partial x)$ в выражении $\delta J(\bar{u}, \varphi)$, мы видим, что

$$\delta J(\bar{u}, \varphi) = \int_{t_0}^{t_1} \left[-\dot{p}^T y - p^T \left(\frac{\partial f}{\partial x}\right) y + \frac{\partial g}{\partial u} \varphi\right] dt.$$

Интегрируя по частям, получаем, что

$$\int_{t_0}^{t_1} -\dot{p}^T y\,dt = \int_{t_0}^{t_1} p^T \dot{y}\,dt - [p^T y]_{t_0}^{t_1}.$$

Так как $p(t_1)=0$ и $y(t_0)=0$, то $[p^T y]_{t_0}^{t_1}=0$, откуда

$$\int_{t_0}^{t_1} -\dot{p}^T y\,dt = \int_{t_0}^{t_1} p^T \dot{y}\,dt = \int_{t_0}^{t_1} \left[p^T \left(\frac{\partial f}{\partial x}\right) y + p^T \frac{\partial f}{\partial u}\,\varphi\right] dt.$$

Следовательно,

$$\delta J(\bar{u}, \varphi) = \int_{t_0}^{t_1} \left[p^T \frac{\partial f}{\partial u} + \frac{\partial g}{\partial u}\right] \varphi\,dt.$$

Мы видим, что в пространстве $V=(L^2[t_0, t_1])^n$ функционал $\delta J(\bar{u}, \varphi)$ непрерывен и линеен относительно φ, причем $p^T \frac{\partial f}{\partial u} + \frac{\partial g}{\partial u}$ есть градиент функционала J в точке $\bar{u}$.

Применение теоремы 4 приводит к следующему необходимому условию оптимальности.

Для того чтобы управление $\bar{u}(t)$ было оптимальным при сделанных выше предложениях, необходимо, чтобы существовал вектор $p(t)$ (сопряженный вектор), удовлетворяющий системе дифференциальных уравнений

$$-\frac{dp}{dt} = \left(\frac{\partial f}{\partial x}\right)^T p + \left(\frac{\partial g}{\partial x}\right)^T,$$
$$p(t_1) = 0,$$

для которого справедливо соотношение

$$p^T \frac{\partial f}{\partial u}(x(\bar{u}, t), \bar{u}(t), t) + \frac{\partial g}{\partial u}(x(\bar{u}, t), \bar{u}(t), t) = 0.$$

Если при данном $p(t)$ определить *гамильтониан системы* формулой

$$H(x, u, p, t) = g(x, u, t) + p^T f(x, u, t),$$

то мы видим, что высказанное выше условие равносильно условию, что для оптимального управления $\bar{u}(t)$ и для соответствующей траектории $x(\bar{u}(t), t)$ выполняется соотношение $\frac{\partial H}{\partial u} = 0$ для всех моментов времени t.

Иначе говоря, оптимальное управление есть необходимо такое управление, которое делает гамильтониан стационарным в каждый момент времени. Этот результат, имеющий исключительно большое значение как с теоретической точки зрения, так и с точки зрения приложений, известен под названием *принципа максимума Понтря-*

гина. Очевидно, что его справедливость может быть установлена в более слабых предположениях, чем те, которые были здесь использованы для упрощения изложения. Можно сослаться на [28, 21, 22, 25].

3.8. Приложение к проектированию на замкнутое выпуклое множество в гильбертовом пространстве. Пусть V — гильбертово пространство и $K \subset V$ — сильно замкнутое выпуклое подмножество.

Пусть $u_0 \in V$ произволен; рассмотрим задачу оптимизации с ограничениями:

$$\min J(u), \quad J(u) = \|u - u_0\|^2,$$
$$u \in K.$$

1) Существование и единственность.

Функционал J выпуклый и в каждой точке имеет градиент, равный

$$J'(u) = 2(u - u_0),$$

следовательно, согласно предложению 7 (п. 3.4), функционал J слабо полунепрерывен снизу и, согласно следствию 1 (п. 3.2), имеет в K по крайней мере один минимум. Так как, кроме того, J — строго выпуклый функционал, то соответствующая точка минимума u^* определена однозначно. Она называется *проекцией элемента u_0 на K.*

2) Характеризация проекции.

Поскольку существование и единственность уже доказаны, то теорема 6 позволяет дать следующую характеризацию проекции u^*:

$$\langle J'(u^*), u - u^* \rangle \geqslant 0 \qquad \forall u \in K.$$

Следовательно, мы можем сформулировать это утверждение в следующем виде.

Свойство 2. *Проекция u^* элемента u_0 на (сильно замкнутое выпуклое) множество K характеризуется вариационным неравенством*

$$\langle u^* - u_0, u - u^* \rangle \geqslant 0 \qquad \forall u \in K.$$

Укажем, наконец, важное свойство проекции.

Свойство 3. *Если u_0^* и u_1^* — проекции на K двух элементов u_0 и u_1, то*

$$\|u_0^* - u_1^*\| \leqslant \|u_0 - u_1\|,$$

иначе говоря, проектирование является сжатием.

Доказательство. Для любого $u \in K$ имеем $\langle u_0^* - u_0, u - u_0^* \rangle \geqslant 0$; следовательно, полагая $u = u_1^*$, получим

$$\langle u_0^* - u_0, u_1^* - u_0^* \rangle \geqslant 0;$$

аналогично, для любого $u \in K$ имеем

$$\langle u_1^* - u_1, u - u_1^* \rangle \geqslant 0;$$

следовательно, полагая $u = u_0^*$, получаем, что

$$\langle u_1^* - u_1, u_0^* - u_1^* \rangle \geqslant 0.$$

Складывая, видим, что

$$\langle (u_0^* - u_1^*) + (u_1 - u_0), (u_1^* - u_0^*) \rangle \geqslant 0,$$

откуда вытекает неравенство

$$\| u_1^* - u_0^* \|^2 \leqslant \langle (u_1 - u_0), (u_1^* - u_0^*) \rangle$$

и, вследствие неравенства Коши — Буняковского, также неравенство

$$\| u_1^* - u_0^* \|^2 \leqslant \| u_1 - u_0 \| \| u_1^* - u_0^* \|.$$

Если $\| u_1^* - u_0^* \| \neq 0$, то действительно получаем

$$\| u_1^* - u_0^* \| \leqslant \| u_1 - u_0 \|.$$

Если же $\| u_1^* - u_0 \| = 0$, то это же неравенство остается справедливым, поскольку $\| u_1 - u_0 \| \geqslant 0$.

§ 4. Алгоритмы бесконечномерной оптимизации

Цель этого параграфа состоит главным образом в том, чтобы описать основные методы бесконечномерной оптимизации и показать их исключительно большое формальное подобие методам, относящимся к конечномерному случаю.

Однако мы видим, что существуют и существенные различия, которые проистекают из того факта, что топологические свойства векторных пространств, рассматриваемых в том и в другом случае, существенно различны (см. теорему о слабой компактности из п. 2.6). Это приводит к различиям в природе делаемых предположений и в технике доказательств.

4.1. Оптимизация без ограничений: метод наискорейшего спуска. Рассмотрим здесь случай, когда V — гильбертово пространство (скалярное произведение обозначается $\langle \cdot, \cdot \rangle$), а функционал J имеет в каждой точке u градиент, который мы обозначим $J'(u)$. Тогда имеем $\delta J(u, \varphi) = \langle J'(u), \varphi \rangle$.

Как и в конечномерном случае, метод наискорейшего спуска состоит в том, чтобы, исходя из точки u^k, переместиться затем в направлении $d^k = -J'(u^k)/\|J'(u^k)\|$. Тогда точка u^{k+1} определяется равенством $u^{k+1} = u^k + \theta_k d^k$, где θ_k минимизирует функцию $g(\theta) = J(u^k + \theta d^k)$ на множестве $\theta \geqslant 0$.

Изучим условия, при выполнении которых последовательность $\{u^k\}$, порожденная алгоритмом наискорейшего спуска, сходится к оптимальному решению задачи $\min\limits_{v \in V} \{J(u)\}$.

Заметим сначала, что в любой такой точке u^k, что $J'(u^k) \neq 0$, направление $d^k = -J'(u^k)/\|J'(u^k)\|$ есть направление спуска, так как

$$\frac{dg}{d\theta}(0) = -\left\langle J'(u^k), \frac{J'(u^k)}{\|J'(u^k)\|} \right\rangle = -\|J'(u^k)\| < 0.$$

Следовательно, если $g(\theta)$ непрерывно дифференцируема, то $g(\theta)<g(0)$ при достаточно малом $\theta>0$. В любом случае имеем

$$J(u^{k+1})\leqslant J(u^k) \quad \forall k.$$

Определение 15. Градиент $J'(u)$ называется *равномерно непрерывным по* u, если для любого $\varepsilon>0$ существует такое $\eta(\varepsilon)>0$, что

$$\|u-v\|\leqslant\eta(\varepsilon)\Rightarrow\forall\varphi,\quad \|\varphi\|=1:\ |\langle J'(u)-J'(v),\ \varphi\rangle|\leqslant\varepsilon. \qquad (5)$$

Предположим, кроме того, что в предыдущем определении функция $\eta(\varepsilon)$ является *сходящейся*; иначе говоря, предположим, что

$$\eta(\varepsilon)\to 0_+ \Leftrightarrow \varepsilon\to 0_+.$$

Это предположение не слишком ограничительно. Чтобы гарантировать его выполнение, достаточно, например, предположить, что градиент J' удовлетворяет условию Липшица вида

$$\|J'(u)-J'(v)\|\leqslant\beta\|u-v\| \qquad \forall u,\quad \forall v.$$

Введем обозначения

$$d^k=-\frac{J'(u^k)}{\|J'(u^k)\|},$$

$$g_k(\theta)=J\left(u^k+\theta d^k\right),$$

$$g_k'(\theta)=\frac{dg_k(\theta)}{d\theta}=\left\langle J'\left(u^k+\theta d^k\right),d^k\right\rangle,$$

$$\Delta J_k(\theta)=J\left(u^k\right)-J\left(u^k+\theta d^k\right),$$

где индекс k напоминает, что все величины относятся к текущей точке u^k.

Заметим, что в предыдущих обозначениях из предположения равномерной непрерывности J следует (поскольку $\|d^k\|=1$), что

$$\theta\leqslant\eta(\varepsilon)\Rightarrow\left|g_k'(\theta)-g_k'(0)\right|\leqslant\varepsilon \qquad \forall\varepsilon>0.$$

Лемма 1. *Предположим, что J' равномерно непрерывно. Определим для любого k величину θ_k соотношениями*

$$\theta_k>0,$$

$$g_k'(\theta_k)=0,$$

$$J\left(u^k+\theta_k d^k\right)\leqslant J\left(u^k+0d^k\right) \qquad \forall 0\in[0,\theta_k].$$

Тогда

$$\lim_{k\to\infty}J\left(u^k\right)-J\left(u^{k+1}\right)=0\Rightarrow\lim_{k\to\infty}\left|\left\langle J'\left(u^k\right),d^k\right\rangle\right|=0.$$

Доказательство. Заметим сначала, что

$$\theta_k>\eta\left(\left|cg_k'(0)\right|\right) \qquad \forall c\in(0,1).$$

Действительно, в противоположном случае мы имели бы (по определению функции $\eta(\varepsilon)$) неравенство

$$|\langle J'(u^k+\theta_k d^k)-J'(u^k), d^k\rangle| \leqslant |cg_k'(0)|,$$

которое может быть записано в виде

$$|g_k'(\theta_k)-g_k'(0)| \leqslant |cg_k'(0)|.$$

Его можно переписать еще и в виде

$$|g_k'(0)| \leqslant |cg_k'(0)|.$$

Так как $c<1$, то мы получили бы противоречие.

С другой стороны, по определению θ_k имеем для любого $\theta \in [0, \theta_k]$

$$J(u^k+\theta_k d^k) \leqslant J(u^k+\theta d^k),$$

откуда

$$J(u^k)-J(u^k+\theta d^k) \leqslant J(u^k)-J(u^k+\theta_k d^k).$$

Это можно записать также в виде

$$\Delta J_k(0) \leqslant \Delta J_k(\theta_k) \qquad \forall \theta \in [0, \theta_k].$$

В частности, выбирая $0=\bar{\theta}_k=\eta(|cg_k'(0)|)<\theta_k$, имеем

$$\Delta J_k(\bar{\theta}_k) \leqslant \Delta J_k(0_k).$$

Применяя формулу конечных приращений, можем записать равенство

$$\Delta J_k(0) = g_k(0)-g_k(\theta)=\theta g_k'(\hat{\theta}), \qquad \hat{\theta} \in (0, \theta).$$

Для $0=\bar{\theta}_k$ имеем

$$\Delta J_k(\bar{\theta}_k) = \bar{\theta}_k' g_k'(\hat{\theta}_k), \qquad \hat{\theta}_k \in (0, \bar{\theta}_k).$$

Но, поскольку $\bar{\theta}_k=\eta(|cg_k'(0)|)$, то получаем

$$\hat{\theta}_k < \eta(|cg_k'(0)|),$$

отсюда следует, что

$$|g_k'(\hat{\theta}_k)-g_k'(0)| < |cg_k'(0)|,$$

откуда

$$g_k'(\hat{\theta}_k) \geqslant (1-c)|g_k'(0)|$$

и, следовательно,

$$\Delta J_k(\bar{\theta}_k) \geqslant \bar{\theta}_k(1-c)|g_k'(0)|,$$

или еще, если заменить $\bar{\theta}_k$ на $\eta(|cg_k'(0)|)$,

$$\Delta J_k(\theta_k) \geqslant \Delta J_k(\bar{\theta}_k) \geqslant (1-c)|g_k'(0)|\,\eta(|cg_k'(0)|).$$

Следовательно, если предположить, что

$$J(u^k)-J(u^{k+1})=\Delta J_k(\theta_k) \to 0, \qquad k \to \infty,$$

то отсюда выводим, что

$$\alpha_k \eta(\alpha_k) \to 0, \qquad k \to \infty,$$

где положено

$$\alpha_k = c\,|\,g_k'(0)\,| = c\,|\,\langle J'(u^k), d^k\rangle\,|.$$

Покажем, что отсюда следует, что необходимо $\alpha_k \to 0$. Если α_k не стремится к нулю при $k \to \infty$, то из последовательности $\{\alpha_k\}$ можно выделить такую подпоследовательность $\{\alpha_l\}_{l \in L}$, что $\alpha_l \geqslant \bar{\alpha} > 0$ $(\forall l \in L)$. Но тогда из соотношения $\alpha_l \eta(\alpha_l) \to 0$ следует, что $\eta(\alpha_l) \to 0$, откуда $\alpha_l \to 0$ (так как по условию функция η сходящаяся). Отсюда получается противоречие.

Следовательно, $\Delta J_k(0_k) \to 0$ при $k \to \infty \Rightarrow |\,g_k'(0)\,| \to 0$, иначе говоря,

$$|\langle J'(u^k),\ d^k\rangle| \to 0,$$

что завершает доказательство.

Тогда можно сформулировать следующую теорему.

Теорема 7. *Пусть J — функционал, который нужно минимизировать на гильбертовом пространстве V. Пусть $\{u^k\}$ — последовательность, порожденная алгоритмом наискорейшего спуска.*

Предположим, что J ограничен снизу и что J имеет градиент в каждой точке, причем этот градиент удовлетворяет условиям леммы 1 (J' равномерно непрерывен).

Тогда:

i) *Если отображение $u \to \|J'(u)\|$ слабо непрерывно, то в любой точке u, являющейся слабой точкой накопления последовательности $\{u^k\}$, выполняется условие $J'(u) = 0$.*

Существует по крайней мере одна такая точка, если $J(v) \to +\infty$ при $\|v\| \to +\infty$.

ii) *Если функционал J выпуклый и если $J(v) \to \infty$ при $\|v\| \to \infty$, то последовательность $\{u^k\}$ имеет по крайней мере одну слабую точку накопления, и в этой точке функционал J имеет минимум.*

Доказательство. Так как J ограничен снизу, а последовательность $J(u^k)$ убывает, то

$$\lim_{k \to \infty} J(u^k) - J(u^{k+1}) = 0.$$

Тогда, согласно лемме 1, получаем отсюда, что

$$\lim_{k \to \infty} |\langle J'(u^k), d^k\rangle| = 0,$$

и так как d^k определяется формулой

$$d^k = -\frac{J'(u^k)}{\|J'(u^k)\|},$$

то отсюда следует, что

$$\lim_{k \to \infty} \|J'(u^k)\| = 0.$$

Заметим, с другой стороны, что из предположения, что $J(v) \to +\infty$ при $\|v\| \to +\infty$, следует — поскольку последовательность $J(u^k)$ не возрастает,— что все u^k содержатся в общем ограниченном множестве. При этих условиях, вследствие теоремы о слабой компактности (п. 2.6), последовательность $\{u^k\}$ имеет слабую точку прикосновения u (можно выделить некоторую подпоследовательность, слабо сходящуюся к u).

Докажем теперь утверждение (i). Пусть u — слабая точка прикосновения последовательности $\{u^k\}$. Таким образом, можно выделить из последовательности $\{u^k\}$ такую подпоследовательность $\{u^l\}_{l \in L}$ ($L \subset \mathbf{N}$), что $u^l \to u$ (слабо) при $l \to \infty$ ($l \in L$).

Так как функционал $u \to \|J'(u)\|$ слабо непрерывен, то получаем отсюда

$$\lim_{\substack{l \to \infty \\ l \in L}} \|J'(u^l)\| = \|J'(u)\| = 0.$$

С другой стороны, если $J(v) \to +\infty$ при $\|v\| \to \infty$, то существует по крайней мере одна такая слабая точка прикосновения.

Докажем теперь утверждение (ii). По предположению, существует по крайней мере одна слабая точка прикосновения; обозначим ее u^*. Таким образом, имеем $u^l \to u^*$ (слабо) при $l \to \infty$, $l \in L \subset \mathbf{N}$. Но функционал J выпуклый, поэтому для любого $v \in V$ имеем

$$J(v) \geqslant J(u^l) + \langle J'(u^l),\ v - u^l \rangle.$$

Так как $\|J'(u^l)\| \to 0$ и $\|v - u^l\|$ — ограниченная последовательность, то

$$\langle J'(u^l),\ v - u^l \rangle \to 0, \qquad l \to \infty, \quad l \in L.$$

С другой стороны, так как J — выпуклый и слабо полунепрерывный снизу (см. п. 3.4, предложение 7) функционал, то отсюда получаем

$$\lim_{\substack{l \to \infty \\ l \in L}} J(u^l) \geqslant J(u^*).$$

Отсюда, таким образом, следует, что

$$J(v) \geqslant J(u^*) \qquad \forall v \in V.$$

Если теперь функционал J является строго выпуклым, то минимум u^* функционала J единствен, следовательно, последовательность $\{u^k\}$ имеет единственную слабую точку прикосновения.

4.2. Приложение методов наискорейшего спуска к решению задач оптимального управления. В п. 3.7 мы видели, что для задачи оптимального управления:

найти на $[t_0, t_1]$ управление $u(t)$ минимизирующее функционал

$$J(u) = \int_{t_0}^{t_1} g(x(t), u(t), t)\, dt$$

при условиях:

$$\dot{x}(t)=f(x(t),\ u(t),\ t), \qquad x(t_0)=\xi_0 \quad \text{(данный вектор)},$$

можно определить градиент функционала $u \to J(u)$ в произвольной точке u, и этот градиент может быть вычислен следующим образом.

a) Нужно определить траекторию, соответствующую управлению u, т. е. проинтегрировать уравнение состояния

$$\dot{x}(t)=f(x(t),\ u(t),\ t)$$

на промежутке от t_0 до t_1, исходя из точки $x(t_0)=\xi_0$.

b) Нужно проинтегрировать сопряженную систему

$$-\dot{p}(t)=\left(\frac{\partial f}{\partial x}\right)^T p+\left(\frac{\partial g}{\partial x}\right)^T$$

от t_1 до t_0 при начальных условиях $p(t_1)=0$.

Тогда градиент J в точке u определяется формулой

$$J'(u)=p^T\frac{\partial f}{\partial u}+\frac{\partial g}{\partial u},$$

где частные производные вычисляются на траектории $x(u(t),\ t)$, соответствующей управлению u.

Далее ищется оптимум J в направлении $-J'(u)$, и процедура повторяется, исходя из нового полученного таким образом управления $u(t)$.

Этот технический прием является одним из наиболее часто применяемых методов решения задач оптимального управления.

4.3. Оптимизация с ограничениями: градиент с фиксированным шагом в сочетании с проектированием. Расположимся раз навсегда в гильбертовом пространстве V (со скалярным произведением $\langle\cdot,\ \cdot\rangle$).

Мы ищем решение задачи оптимизации при наличии ограничений

$$\min\ J(u),$$

$$u\in U\subset V,$$

где J — выпуклый, дифференцируемый (и имеющий градиент) функционал, а U — сильно замкнутое выпуклое подмножество V.

Предположим, кроме того, что либо U ограничено, либо $J(u)\to+\infty$ при $\|u\|\to+\infty$.

Известно, что при этих условиях предыдущая задача имеет оптимальное решение $u^*\in U$ (см. следствие 1 п. 3.2).

Если, кроме того, функционал J является строго выпуклым, то рассматриваемая задача имеет единственное решение, и это решение характеризуется вариационным неравенством

$$\langle J'(u^*),\ u-u^*\rangle\geqslant 0 \qquad \forall u\in U,$$

где $J'(u^*)$ обозначает градиент J в u^* (см. теорему 6 п. 3.5).

Для любого $u \in V$, $u \notin U$ обозначим через $\mathrm{Pr}_U(u)$ проекцию точки u на выпуклое множество U и рассмотрим следующий алгоритм, в котором $\theta > 0$ есть фиксированный шаг перемещения:

$$u^{k+1} = \mathrm{Pr}_U [u^k - \theta J'(u^k)].$$

Теорема 8. *Если J является α-выпуклым ($\alpha > 0$) и если отображение $u \to J'(u)$ удовлетворяет условию Липшица:*

$$\|J'(u) - J'(v)\| \leqslant \beta \|u - v\| \qquad \forall u, \quad \forall v,$$

то для любого значения $\theta \in (0, 2\alpha\beta^{-2})$ алгоритм градиента с проектированием сходится к точке минимума u^* функционала J на U.

Доказательство. Заметим сначала, что точка u^* характеризуется вариационным неравенством

$$\langle J'(u^*), v - u^* \rangle \geqslant 0 \qquad \forall v \in U,$$

которое можно переписать в виде

$$\langle u^* - (u^* - \theta J'(u^*)), v - u^* \rangle \geqslant 0 \qquad \forall v \in U, \quad \forall \theta > 0,$$

и, следовательно (см. свойство 2 п. 3.8),

$$u^* = \mathrm{Pr}_U (u^* - \theta J'(u^*)).$$

Следовательно, мы можем записать соотношение

$$u^{k+1} - u^* = \mathrm{Pr}_U [u^k - \theta J'(u^k)] - \mathrm{Pr}_U [u^* - \theta J'(u^*)].$$

Так как оператор проектирования является сжатием (см. свойство 3 п. 3.8), то

$$\|u^{k+1} - u^*\|^2 \leqslant \|u^k - u^* - \theta (J'(u^k) - J'(u^*))\|^2,$$

где правую часть можно тогда переписать в виде

$$\|u^k - u^*\|^2 + \theta^2 \|J'(u^k) - J'(u^*)\|^2 - 2\theta \langle J'(u^k) - J'(u^*), u^k - u^* \rangle;$$

так как J является α-выпуклым, то мы имеем

$$\langle J'(u^k) - J'(u^*), u^k - u^* \rangle \geqslant \alpha \|u^k - u^*\|^2,$$

и так как $J'(u)$ липшицев, то

$$\|J'(u^k) - J'(u^*)\|^2 \leqslant \beta^2 \|u^k - u^*\|^2.$$

Следовательно, мы получаем оценку

$$\|u^{k+1} - u^*\|^2 \leqslant (1 - 2\alpha\theta + \theta^2\beta^2) \|u^k - u^*\|^2,$$

отсюда вытекает, что для $\theta \in (0, 2\alpha\beta^{-2})$ имеем

$$\|u^{k+1} - u^*\| \leqslant \gamma \|u^k - u^*\|$$

с $\gamma < 1$, что и доказывает сходимость алгоритма.

Замечание 1. Предыдущий метод можно равным образом применить и к оптимизации без ограничений: тогда достаточно взять $U = V$. Мы получим метод *градиента с фиксированным шагом*. Условия на θ, обеспечивающие сходимость, те же, что и в случае при наличии ограничений.

Замечание 2. Для того чтобы алгоритм градиента с фиксированным шагом представлял практический интерес, нужно, чтобы было легко провести вычисление проекции произвольной точки на множество U.

4.4. Оптимизация с ограничениями: методы штрафов. Пусть V, как и в предыдущем пункте, есть гильбертово пространство и пусть мы ищем решение задачи

$$\min J(u), \quad u \in U \subset V. \tag{P}$$

Функция $u \to H(u)$ называется *функцией штрафа* относительно подмножества U, если

$$H(u) = 0 \Leftrightarrow u \in U,$$

$$H(u) \geqslant 0 \quad \text{для всех } u,$$

H слабо полунепрерывна снизу.

Как и в конечномерном случае (см. гл. 6 § 1), идея состоит в том, чтобы заменить задачу (P) последовательностью задач без ограничений:

$$\min J_r(u) = J(u) + rH(u), \tag{P_r}$$

где $r > 0$ — параметр (*коэффициент штрафа*), который мы устремляем к $+\infty$.

Предположим в дальнейшем в этом разделе, что

J слабо полунепрерывен снизу,

$J(u) \to \infty$ при $\|u\| \to +\infty$,

J ограничен снизу.

При этих условиях для любого значения $r > 0$ задача (P_r) имеет оптимальное решение $\bar{u}_r$. Следующий результат изучает сходимость последовательности $\bar{u}_r$ к оптимальному решению u^* задачи (P) при $r \to \infty$. Он обобщает теорему 1 гл. 6.

Теорема 9. *Предположим, что J слабо полунепрерывен снизу (сл. п. н. сн.), ограничен снизу и что $J(u) \to +\infty$ при $\|u\| \to +\infty$. Предположим, что подмножество U слабо замкнуто. Тогда любая слабая точка прикосновения последовательности $\{\bar{u}_r\}$ есть оптимальное решение задачи (P).*

Доказательство. Для любого значения $r > 0$ и любого $u_0 \in U$ имеем

$$J_r(\bar{u}_r) = J(\bar{u}_r) + rH(\bar{u}_r) \leqslant J(u_0) + rH(u_0),$$

и так как $H(u_0) = 0$ для $u_0 \in U$, то мы видим, что

$$J(\bar{u}_r) \leqslant J(u_0) \qquad \forall u_0 \in U.$$

Следовательно, величина $J(\bar{u}_r)$ остается ограниченной фиксированным числом, не зависящим от r. Так как $J(u) \to +\infty$ при $\|u\| \to \infty$, то отсюда следует, что вся последовательность $\{\bar{u}_r\}$ содержится в некотором ограниченном множестве.

Используя теорему о слабой компактности (п. 2.6), мы видим, что из этой последовательности можно выделить подпоследователь-

пость $\{\bar{u}_{r'}\}$, слабо сходящуюся к $u^* \in V$. Так как U слабо замкнуто, то $u^* \in U$.

С другой стороны, так как для любого r' имеем

$$J(\bar{u}_{r'}) + r'H(\bar{u}_{r'}) \leqslant J(u_0)$$

и мы можем переписать это соотношение в виде

$$H(\bar{u}_{r'}) \leqslant \frac{1}{r'}[J(u_0) - J(\bar{u}_{r'})],$$

то, поскольку J ограничен снизу, и, таким образом, существует такое число m, что

$$J(u) \geqslant m \quad \forall u,$$

то отсюда следует соотношение

$$H(\bar{u}_{r'}) \leqslant \frac{1}{r'}[J(u_0) - m] \quad \forall r'$$

и, следовательно, $H(\bar{u}_{r'}) \to 0$ при $r' \to +\infty$.

Так как H сл. п. н. сн., то теперь можно записать — вследствие того, что $\bar{u}_{r'} \to u^*$ (слабо), — что

$$\lim_{r' \to +\infty} H(\bar{u}_{r'}) \geqslant H(u^*),$$

откуда $H(u^*) \leqslant 0$.

Так как по определению функции штрафа должно быть также $H(u^*) \geqslant 0$, то отсюда получаем, что $H(u^*) = 0$ и $u^* \in U$.

Наконец, так как

$$J(\bar{u}_{r'}) \leqslant J(u_0) \qquad \forall u_0 \in U,$$

то, используя тот факт, что функционал J сл. п. п. сп., имеем

$$J(u^*) \leqslant \lim_{r' \to +\infty} J(\bar{u}_{r'}) \leqslant J(u_0)$$

— и это для любого $u_0 \in U$. Отсюда следует, что u^* — оптимальное решение задачи (P).

4.5. Оптимизация с ограничениями: методы двойственности. Теория седловых точек и теория двойственности, развитые в гл. 6, обобщаются без существенных трудностей на случай бесконечномерных пространств.

Предположим, что задача оптимизации при наличии ограничений задана в следующей форме:

$$\begin{aligned} J(v) &\to \min, \\ F(v) &= 0, \\ v &\in V, \end{aligned} \qquad (P)$$

где F — отображение V в Y.

Предположим, что V и Y — нормированные векторные пространства. Следовательно, мы можем определить пространство Y^*, сопряженное к Y.

Таким образом, любой элемент $\lambda \in Y^*$ является непрерывной линейной формой на Y, и результат применения отображения $\lambda \in V^*$ к элементу $y \in Y$ будет обозначаться λy (λy есть вещественное число).

Тогда можно определить *функцию Лагранжа* задачи (P) формулой

$$L(v, \lambda) = J(v) + \lambda F(v), \qquad \lambda \in Y^*.$$

Понятие седловой точки определяется, как и в конечномерном случае: точка $(\bar{v}, \bar{\lambda})$ является *седловой точкой* тогда и только тогда, когда

$$L(\bar{v}, \lambda) \leqslant L(\bar{v}, \bar{\lambda}) \leqslant L(v, \bar{\lambda}) \qquad \forall v, \quad \forall \lambda.$$

Можно доказать — аналогично конечномерному случаю,— что если точка $(\bar{v}, \bar{\lambda})$ является седловой точкой функции Лагранжа, то $\bar{v}$ есть точка (глобального) оптимума задачи (P).

И на этот раз существование седловой точки можно гарантировать в выпуклом случае (иначе говоря, здесь — в случае, если J — выпуклый функционал, а F — линейное отображение, поскольку ограничения суть ограничения в форме равенств).

Можно также определить *дуальную функцию* для любого $\lambda \in Y^*$

$$w(\lambda) = \min_{v \in V} \{L(u, \lambda)\}$$

(следовательно, это — функционал на Y^*), и седловая точка — если она существует — может быть определена с помощью решения *дуальной задачи*

$$\begin{gathered} \max w(\lambda), \\ \lambda \in Y^*. \end{gathered} \tag{D}$$

Решение дуальной задачи (которая является задачей без ограничений) может быть проведено, например, с помощью метода наискорейшего спуска (п. 4.1) или метода градиента с фиксированным шагом (п. 4.3).

Рассмотрим в качестве примера задачу оптимального управления вида

$$\min J(x, u) = \int_{t_0}^{t_1} g(x(t), u(t), t)\, dt,$$

$$\dot{x}(t) = f(x(t), u(t), t), \tag{6}$$

$$x(t_0) = \xi_0. \tag{7}$$

Предположим, что мы ищем u в $(L^2[t_0, t_1])^m$ и x в $(H^1[t_0, t_1])^n$.

Уравнение состояния (6) может быть, таким образом, переписано в виде

$$0 = F(x, u) = -\dot{x}(t) + f(x(t), u(t), t),$$

где F — отображение $V = (H^1[t_0, t_1])^n \times (L^2[t_0, t_1])^m$ в $Y = (L^2[t_0, t_1])^n$.

Если функционал J является выпуклым по (x, u) и если f линейна по (x, u), то седловая точка существует.

Свяжем с каждым уравнением состояния

$$\dot{x}_i(t) = f_i(x(t), u(t), t)$$

функцию (множитель Лагранжа) $\lambda_i(t) \in L^2[t_0, t_1]$.

Образуем из этих функций вектор $\lambda(t)$, принадлежащий Y^*.

Функция Лагранжа, связанная с предыдущей задачей, записывается в виде

$$L(x, u, \lambda, t) = J(x, u) + \int_{t_0}^{t_1} \lambda^T(t) [-\dot{x} + f(x, u, t)]\, dt.$$

Вычисление дуальной функции $w(\lambda)$ для фиксированного λ сводится, таким образом, к решению задачи вариационного исчисления

$$\min_{(x,u)} \int_{t_0}^{t_1} \left(g(x, u, t) + \lambda^T(t) [-\dot{x} + f(x, u, t)]\right) dt,$$

$$x(t_0) = \xi_0.$$

Выпишем тогда уравнения Эйлера — Лагранжа, связанные с этой задачей, предполагая, что требуемые для этого условия регулярности выполнены (см. п. 3.6). Получим

$$\frac{\partial g}{\partial x} + \lambda^T \frac{\partial f}{\partial x} - \frac{d}{dt}\left[-\lambda^T(t)\right] = 0,$$

$$\frac{\partial g}{\partial u} + \lambda^T \frac{\partial f}{\partial u} = 0.$$

Первое из этих уравнений есть не что иное, как дифференциальное уравнение, определяющее сопряженное состояние; второе уравнение выражает стационарность в точке u функционала $g(x, u, t) + \lambda^T f(x, u, t)$, т. е. гамильтониана системы, в каждый момент времени (см. п. 3.7).

Пусть $(x_\lambda(t), u_\lambda(t))$ — решение предыдущей задачи вариационного исчисления. Тогда получаем

$$w(\lambda) = L(x_\lambda(t), u_\lambda(t), \lambda(t), t).$$

Можно доказать, как в случае конечномерных пространств, что $w(\lambda)$ — вогнутый функционал от λ и что

$$F(\dot{x}_\lambda, u_\lambda) = -\dot{x}_\lambda(t) + f(x_\lambda, u_\lambda, t)$$

есть субградиент w в точке λ.

Решение дуальной задачи можно, таким образом, осуществить методом наискорейшего спуска или методом градиента с фиксированным шагом. Следовательно, новое значение λ определяется на каждой следующей итерации формулой типа

$$\lambda'(t) = \lambda(t) + \theta [-\dot{x}_\lambda(t) + f(x_\lambda(t), u_\lambda(t), t)].$$

4.6. Приближение бесконечномерных задач конечномерными задачами. Метод Галеркина. Естественная и очень широко используемая идея решения задачи оптимизации в бесконечномерном пространстве V состоит в ограничении этой задачи на конечномерное подпространство V_n размерности n.

Здесь мы ограничимся изучением общего принципа этого метода и не будем входить в подробности его работы. Действительно, в соответствии с природой изучаемой задачи особенности применения этого метода могут принимать весьма различные формы: методы конечных разностей, методы конечных элементов, методы Ритца — Галеркина. Подчеркнем, с другой стороны, что теоретическое оправдание этих методов основано на свойстве сепарабельности рассматриваемых гильбертовых пространств, которое позволяет ввести понятие полной ортонормированной системы, являющееся распространением понятия базиса на бесконечномерный случай.

Определение 16 (плотное подмножество).

Пусть V — банахово пространство, D — подмножество V. Множество D называется *плотным* в V, если для любого элемента $v \in V$ существует элемент $u \in D$, сколь угодно близкий к v. Иначе говоря:

$$\|u - v\| < \varepsilon \qquad \forall v \in V, \quad \forall \varepsilon > 0, \quad \exists u \in D.$$

Определение 17 (тотальное семейство). Счетное семейство элементов пространства V вида $\xi = \{\xi_1, \ldots, \xi_k, \ldots\}$ называется *тотальным*, если подпространство V_ξ, порожденное этим семейством, плотно в V (иначе говоря, если $\overline{V_\xi} = V$, где $\overline{V_\xi}$ означает замыкание V_ξ в смысле сильной топологии).

Определение 18. Банахово пространство V называется *сепарабельным*, если оно содержит не более чем счетное тотальное семейство.

Для важных в приложениях гильбертовых пространств можно доказать следующее свойство.

Свойство 4. *Гильбертовы пространства в примерах* 1, 2, 3, 5 *п.* 2.7 *сепарабельны, т. е. содержат не более чем счетное тотальное семейство.*

Определение 19. Пусть V — гильбертово пространство. Семейство $\xi = \{\xi_i | i \in I\}$ элементов пространства V называется *ортонормированным*, если для любых $i \in I$, $j \in I$ выполняются соотношения

$$\langle \xi_i, \xi_j \rangle = 0, \qquad i \neq j,$$
$$\langle \xi_i, \xi_j \rangle = 1, \qquad i = j.$$

В гильбертовом пространстве, исходя из любого счетного семейства элементов

$$\xi = \{\xi_1, \ldots, \xi_n, \ldots\},$$

каждая конечная подсистема которого линейно независима, можно построить ортонормированное семейство $e = \{e_1, \ldots, e_n, \ldots\}$ с помощью *процесса ортогонализации Грама — Шмидта*:

$$e_1 = \frac{\xi_1}{\|\xi_1\|}, \qquad e_n = \frac{\xi_n - \sum_{i=1}^{n-1} \langle \xi_n, e_i \rangle e_i}{\left\| \xi_n - \sum_{i=1}^{n-1} \langle \xi_n, e_i \rangle e_i \right\|}.$$

Из предыдущего вытекает следующее фундаментальное свойство сепарабельных гильбертовых пространств.

Свойство 5. *Любое сепарабельное гильбертово пространство V содержит счетное тотальное ортонормированное семейство*

$$e = \{e_1, \ldots, e_n, \ldots\}.$$

Для произвольного вектора $u \in V$ проекция вектора u на подпространство V_n, порожденное n векторами $\{e_1, \ldots, e_n\}$, определяется формулой

$$u_n = \sum_{i=1}^{n} \langle u, e_i \rangle e_i,$$

и при этом имеем $u_n \to u$ (сильно) при $n \to \infty$.

Рассмотрим теперь приближение бесконечномерной задачи вида

$$\begin{gathered} \min J(v), \\ v \in U \subset V \end{gathered} \tag{P}$$

(где V — гильбертово пространство) конечномерными задачами.

Рассмотрим случай, когда задача (P) имеет единственное оптимальное решение $\bar{u}$, и с этой целью введем следующие предположения:

J — α-выпуклый, дифференцируемый, имеющий градиент и сильно непрерывный функционал;

U — выпуклое сильно замкнутое множество.

Действительно, поскольку функционал J выпуклый, дифференцируемый и имеющий градиент, то он слабо полунепрерывен снизу (согласно предложению 7 п. 3.4). С другой стороны, поскольку J является α-выпуклым, то он удовлетворяет условию $J(u) \to +\infty$ при $\|u\| \to +\infty$, и, вследствие теоремы 3′ п. 3.2, задача (P) имеет оптимальное решение u'. Так как J — строго выпуклый функционал, то это решение единственно.

Для произвольного $n \in \mathbb{N}$ обозначим через V_n подпространство пространства V размерности n, порожденное набором $\{e_1, \ldots, e_n\}$, и предположим, что мы заменили задачу (P) задачей (P_n) в пространстве V_n:

$$\begin{gathered} \min J(v), \\ v \in U_n \subset V_n, \end{gathered} \tag{P_n}$$

где U_n — замкнутое выпуклое подмножество V_n.

Вследствие сделанных ранее предположений задача (P_n) имеет единственное оптимальное решение, которое мы обозначим $\bar{u}_n$. Следующий результат показывает, что при некоторых предположениях $\bar{u}_n \to \bar{u}$ при $n \to \infty$.

Теорема 10. *Предположим, что семейство замкнутых выпуклых подмножеств U_n позволяет приблизить U в следующем смысле:*

1) *для любого $u \in U$ можно найти такую последовательность $\{u_n\}$, что $u_n \in U_n$ для всех n и $u_n \to u$ (сильно);*

2) *если* $u_n \in U_n$ *для любого* n *и если последовательность* $\{u_n\}$ *сходится слабо в* V *к* $u^* \in V$, *то* $u^* \in U$.

Тогда в сделанных выше предположениях относительно функционала J *имеем*

$$\lim_{n\to\infty} \|\bar{u}_n - \bar{u}\| = 0,$$

где $\bar{u}$ *есть оптимальное решение задачи* (P).

Доказательство. Так как $\bar{u} \in U$, то (по построению подмножеств U_n) существует такая последовательность $u_n \in U_n$, что

$$\lim_{n\to\infty} \|u_n - \bar{u}\| = 0.$$

Так как J сильно непрерывен, то

$$\lim_{n\to\infty} J(u_n) = J(\bar{u}).$$

С другой стороны, так как $\bar{u}_n$ есть оптимум для задачи (P_n), то для любого n имеем

$$J(\bar{u}_n) \leqslant J(u_n).$$

Отсюда следует, что существует некоторая постоянная c, удовлетворяющая условию

$$J(\bar{u}_n) \leqslant c \quad \forall n.$$

Так как $J(u) \to +\infty$ при $\|u\| \to +\infty$, то вся последовательность $\{\bar{u}_n\}$ содержится в ограниченном подмножестве V.

Следовательно, мы можем (по теореме о слабой компактности) выделить подпоследовательность $\{\bar{u}_l\}$ $(l \in L \subset \mathbf{N})$, слабо сходящуюся к элементу $u^* \in V$.

Но тогда из условий

$$\bar{u}_l \to u^* \quad \text{в} \quad V \quad \text{(слабо)},$$
$$\bar{u}_l \in U_l,$$

следует, что $u^* \in U$. С другой стороны, так как функционал J слабо полунепрерывен снизу, то

$$J(u^*) \leqslant \lim_{\substack{l\to\infty \\ l\in L}} J(\bar{u}_l).$$

Отсюда следует, что

$$J(u^*) \leqslant J(\bar{u}), \quad u^* \in U, \ \bar{u} \in U.$$

Так как $\bar{u}$ есть единственное оптимальное решение задачи (P), то отсюда с необходимостью следует, что

$$u^* = \bar{u},$$

следовательно, $\bar{u}$ есть слабый предел любой слабо сходящейся подпоследовательности, выделенной из последовательности $\{\bar{u}_n\}$.

Поскольку J, по предположению, является α-выпуклым, то мы можем теперь записать неравенство

$$J(\bar{u}_n) \geqslant J(\bar{u}) + \langle J'(\bar{u}), \bar{u}_n - \bar{u}\rangle + \frac{\alpha}{2}\|\bar{u}_n - \bar{u}\|^2 \quad \forall n,$$

где $J'(u)$ — градиент J в точке $\bar{u}$.

Но, так как $J(u_n) \geqslant J(\bar{u}_n)$ для всех n, то

$$\frac{\alpha}{2}\|\bar{u}_n - \bar{u}\|^2 \leqslant J(u_n) - J(\bar{u}) - \langle J'(\bar{u}), \bar{u}_n - \bar{u}\rangle \leqslant J(u_n) - J(\bar{u}).$$

Правая часть стремится к нулю, так как $J(u_n) \to J(\bar{u})$. Отсюда заключаем, что $\lim\limits_{n\to\infty}\|\bar{u}_n - \bar{u}\| = 0$.

Вопрос о выборе размерности и природы подпространств для наилучшего приближения данной бесконечномерной задачи является одним из наиболее важных и наиболее деликатных в численном анализе. За его более глубоким изучением можно отослать к [36, 18, 34, 29, 7].

СПИСОК ЛИТЕРАТУРЫ

1. Abadie J. (1970), Application of the G.R.G. Algorithm to Optimal Control Problems, in: *Integer and Nonlinear Programming*, (J. Abadie ed.), North Holland, p. 191-213.
2. Auslender A. (1976), *Optimisation. Méthodes numériques*, Masson, Paris.
3. Boudarel R., Delmas J., Guichet P. (1969), *Commande optimale des processus*, Dunod, Paris.
4. Bryson A. E., Ho Y. C. (1969), *Applied Optimal Control*, Blaisdell, New York.
5. Canon M. D., Cullum C. D., Polak E. (1970), *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill, New York.
6*. Cea J. (1971), *Optimisation : théorie et algorithmes*, Dunod.
7. Ciarlet P. (1976), *Numerical Analysis of the Finite Element Method*, Presses Université de Montréal.
8*. Courant R., Hilbert D. (1953), *Methods of Mathematical Physics*, Interscience.
9. Curtain R. F., Pritchard A. J. (1977), *Functional analysis in modern applied mathematics*, Academic Press, New York.
10. Dreyfus S. E. (1965), *Dynamic programming and the calculus of variations*, Academic Press, New York.
11*. Ekeland I., Temam R. (1974), *Analyse convexe et problèmes variationnels*, Dunod-Gauthier-Villars, Paris.
12. Fortin M., Glowinski R. (1982), *Résolution numérique de problèmes aux limites par des méthodes de lagrangiens augmentés*, Collection Méthodes Mathématiques de l'Informatique, Dunod, Paris.
13*. Glowinski R., Lions J. L., Tremolieres R. (1976), *Analyse numérigue des inéquations variationnelles*, Dunod, Paris.
14. Guignard M. (1969), Generalized Kühn-Tucker Conditions for Mathematical Programming Problems in a Banach Space, *S.I.A.M. Journal of Control* 7, p. 239-241.
15. Hestenes M. R. (1966), *Calculus of Variations and Optimal Control Theory*, Wiley, New York.
16. Hwang C. L., Fan L. T. (1967), A Discrete Version of Pontryagin's Maximum, Principle, *Operations Research* 15, p. 139-146.

17. Lasdon L. S., Mitter S., Waren A. (1967), The Method of Conjugate Gradient for Optimal Control Problems, *IEEE Trans. Automatic Control*, AC-12, 2, p. 132-138.
18. Laurent P. J. (1972), *Approximation et Optimisation*, Herman, Paris.
19*. Lions J. L. (1968), *Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles*, Dunod-Gauthier-Villars, Paris.
20. Lions J. L., Faurre P. (1980), *Cours d'analyse numérique, notes d'optimisation*, Cours de l'Ecole Polytechnique, Paris.
21. Luenberger D. G. (1966), A Generalized Maximum Principle in: *Recent Advances in Optimization Techniques*, (A. Lavi, T. P. Vogl eds.), Wiley, New York.
22. Luenberger D. G. (1969), *Optimization by vector space methods*, Wiley, New York.
23. Luenberger D. G. (1972), Mathematical Programming and Control Theory: Trends of Interplay, in: *Perspectives of Optimization*, (T. M. Geoffrion ed.), Addison Wesley, p. 102-133.
24. Mesarovic M. D., Macko D., Takahara Y. (1970), *Theory of Multilevel Hierarchical Systems*, Academic Press, New York.
25. Michel P. (1977), Une démonstration élémentaire du Principle du Maximum de Pontriaguine, *Bull. Math. Economiques*, n° 14, p. 9-17.
26. Pervozvanskiy A. A. (1967), Relationship Between the Basic Theorems of Mathematical Programming and the Maximum Principle, *Engineering Cybernetics* 6, 11.
27. Polak E. (1973), An Historical Survey of Computational Methods in Optimal Control, *S.I.A.M. Review*, 15, p. 553-584.
28*. Pontryagin L. S., Boltyanskii V. G., Gamkrelidze R. V., Mishchenko E. F. (1962), *The Mathematical Theory of Optimal Processes*, Wiley Interscience, New York.
29. Raviart P., Faurre P. (1976), *Cours d'analyse numérique*, Ecole Polytechnique, Paris.
30. Ritter K. (1967), Duality for Nonlinear Programming in Banach Space, *S.I.A.M. J. Appl. Math* 15, p. 294-302.
31*. Rockaffelar R. T. (1970), *Convex analysis*, Princeton University Press, Princeton, New Jersey.
32. Rubio J. E. (1980), Solution of Nonlinear Optimal Control Problems in Hilbert Spaces by Means of Linear Programming Techniques, *Journal Optimization Theory and Appl.* 30, 4, p. 643-661.
33. Saguez C. (1979), *Optimisation et contrôle optimal des systèmes*, Cours de l'Ecole Centrale des Arts et Manufactures, Chatenay-Malabry, France.
34*. Strang G., Fix G. J. (1973), *An Analysis of the Finite Element Method*, Prentice Hall, USA.
35. Varaiya O. (1967), Nonlinear Programming in Banach Spaces, *S.I.A.M. J. Appl. Math.* 15, p. 284-293.
36*. Zienkiewicz O. C. (1971), *The Finite Element Method in Engineering Science*, McGraw-Hill, Londres.

Приложение 1

ОТДЕЛЕНИЕ ВЫПУКЛЫХ МНОЖЕСТВ. ТЕОРЕМА ФАРКАША И МИНКОВСКОГО. ТЕОРЕМА ГОРДАНА

Мы напомним в § 1 некоторые основные результаты, касающиеся отделения выпуклых множеств.

Отсюда мы выведем в § 2 две теоремы, используемые в математическом программировании: теорему Фаркаша и Минковского (дающую необходимые и достаточные условия существования неотрицательного решения системы линейных неравенств) и одно из ее непосредственных следствий — теорему Гордана.

§ 1. Отделение выпуклых множеств

Теорема 1. *Пусть $S \subset \mathbf{R}^n$ — непустое замкнутое выпуклое множество и пусть $y \notin S$. Тогда существует такой ненулевой вектор $a \in \mathbf{R}^n$ и такое число α, что*

$$a^T y > \alpha, \quad a^T x \leqslant \alpha \qquad \forall x \in S$$

(говорят, что гиперплоскость, определенная уравнением $a^T x = \alpha$, разделяет y и S).

Доказательство. Пусть $\bar{x}$ — проекция y на S. Имеем (см. гл. 10, п. 3.8)

$$(x - \bar{x})^T (y - \bar{x}) \leqslant 0 \qquad \forall x \in S. \tag{1}$$

Из соотношения

$$\|y - \bar{x}\|^2 = (y - \bar{x})^T (y - \bar{x}) = y^T (y - \bar{x}) - \bar{x}^T (y - \bar{x})$$

можно вывести с учетом соотношения (1), что

$$\|y - \bar{x}\|^2 \leqslant (y - x)^T (y - \bar{x}) \qquad \forall x \in S.$$

Выбирая тогда $a = y - \bar{x} (\neq 0)$, получаем, что

$$a^T y \geqslant a^T x + \|y - \bar{x}\|^2 \qquad \forall x \in S,$$

и если взять $\alpha = \sup\limits_{x \in S} \{a^T x\}$, то теорема доказана.

Теорема 2. *Пусть $S \subset \mathbf{R}^n$ — выпуклое множество и пусть $\bar{x}$ — точка границы множества S. Тогда S допускает в точке $\bar{x}$ опорную гиперплоскость, т. е. существует такой ненулевой вектор $a \in \mathbf{R}^n$, что*

$$a^T (x - \bar{x}) \leqslant 0 \qquad \forall x \in \mathrm{cl}(S).$$

Доказательство. Так как $\bar{x}$ принадлежит границе множества S, то существует такая последовательность $\{y^k\}$, что $y^k \notin \mathrm{cl}\,(S)$ для любого k и $y^k \to \bar{x}$.

Согласно теореме 1, с каждым y^k можно связать такой ненулевой вектор a^k, что

$$(a^k)^T(x - y^k) < 0 \qquad \forall x \in \mathrm{cl}\,(S). \tag{2}$$

Заметим теперь, что так как $a^k \neq 0$, то всегда можно предполагать, что $\|a^k\| = 1$ в вышеприведенной формуле. Тогда последовательность $\{a^k\}$ целиком содержится в компактном множестве (замкнутом шаре радиуса 1), и поэтому из нее можно выделить подпоследовательность $\{a^l\}_{l \in L}$ $(L \subset \mathbb{N})$, сходящуюся к такому вектору a, что $\|a\| = 1$.

Рассматривая эту подпоследовательность и переходя к пределу в (2) при фиксированном x, получаем, что

$$a^T(x - \bar{x}) \leqslant 0 \qquad \forall x \in \mathrm{cl}(S),$$

откуда и следует теорема.

Из предыдущих теорем 1 и 2 получаем

Следствие 1. *Пусть $S \subset \mathbb{R}^n$ — непустое выпуклое множество и пусть $\bar{x} \notin S$. Тогда существует такой ненулевой вектор $a \in \mathbb{R}^n$, что*

$$a^T(x - \bar{x}) \leqslant 0 \qquad \forall x \in \mathrm{cl}(S).$$

Доказательство. Если $\bar{x} \notin \mathrm{cl}(S)$, то следствие вытекает из теоремы 1. Если $\bar{x}$ принадлежит границе множества S, то следствие вытекает из теоремы 2.

Следующий результат показывает, что два дизъюнктных выпуклых множества всегда могут быть отделены гиперплоскостью.

Теорема 3. *Пусть S_1 и S_2 — два дизъюнктных $(S_1 \cap S_2 = \varnothing)$ непустых выпуклых множества. Тогда существует разделяющая их гиперплоскость, иначе говоря, существует такой ненулевой вектор $a \in \mathbb{R}^n$, что*

$$a^T x_1 \leqslant a^T x_2 \qquad \forall x_1 \in S_1, \quad \forall x_2 \in S_2.$$

Доказательство. Рассмотрим множество

$$S = \{x \mid x = x_1 - x_2,\ x_1 \in S_1,\ x_2 \in S_2\}.$$

Заметим, что множество S выпукло и что $0 \notin S$ (действительно, если бы $0 \in S$, то было бы $S_1 \cap S_2 \neq \varnothing$).

Применяя следствие 1, мы получаем тогда существование такого $a \neq 0$, что

$$a^T x \leqslant 0 \qquad \forall x \in S,$$

откуда следует, что

$$a^T(x_1 - x_2) \leqslant 0 \qquad \forall x_1 \in S_1, \qquad \forall x_2 \in S_2,$$

и теорема доказана.

Отсюда можно вывести

Следствие 2. *Пусть S_1 и S_2 — такие два выпуклых множества, что $S_1 \neq \varnothing$ и $\mathrm{int}\,(S_2) \neq \varnothing$. Тогда если $S_1 \cap \mathrm{int}\,(S_2) = \varnothing$, то существует такой ненулевой вектор $a \in \mathbb{R}^n$, что*

$$\forall x_1 \in S_1, \quad \forall x_2 \in S_2 \Rightarrow a^T x_1 \leqslant a^T x_2.$$

§ 2. Теорема Фаркаша и Минковского. Теорема Гордана

Теорема Фаркаша и Минковского касается существования неотрицательных решений систем линейных уравнений вида

$$\begin{aligned} Ax &= b, \\ x &\geqslant 0, \end{aligned} \tag{3}$$

где A — матрица $m \times n$ с вещественными коэффициентами, $x \in \mathbf{R}^n$, $b \in \mathbf{R}^n$.

Вот ее формулировка.

Теорема 4 (Фаркаш и Минковский). *Необходимое и достаточное условие того, что система* (3) *имеет решение, состоит в следующем:*

для любого такого $u = (u_1, \ldots, u_m)$, *что* $uA \geqslant 0$, *имеем* $ub \geqslant 0$.

Доказательство. Условие необходимо.

Действительно, если система (3) имеет решение $\bar{x} \geqslant 0$, то для любого такого u, что $uA \geqslant 0$, имеем $ub = uA\bar{x} \geqslant 0$, что показывает, что условие (4) выполнено.

Условие достаточно. Покажем, что если система (3) не имеет решений, то (4) не выполнено.

Множество $S = \{y \mid y = Ax,\ x \geqslant 0\}$ замкнуто и выпукло, а вектор b удовлетворяет условию $b \notin S$. Вследствие теоремы 1 (§ 1 настоящего приложения 1) существует такой ненулевой вектор $a \in \mathbf{R}^n$ и такое число α, что $a^T b > \alpha$ и $a^T y \leqslant \alpha$ для любого $y \in S$. Так как $0 \in S$, то должно выполняться соотношение $a^T Ax \leqslant \alpha$ для всех $x \geqslant 0$, что означает, что $a^T A \leqslant 0$.

Полагая $u = -a^T$, мы видим, что он очевидным образом представляет собой m-вектор-строку, не удовлетворяющую условию (4).

Замечание. Теорема Фаркаша и Минковского имеет несколько вариантов, и все они могут быть очень легко сведены к предыдущей формулировке.

а) Если вместо системы уравнений мы рассматриваем системы линейных неравенств вида

$$\begin{aligned} Ax &\leqslant b, \\ x &\geqslant 0, \end{aligned} \tag{3'}$$

то нужно присоединить условие неотрицательности, относящееся к вектору u, и условие (4) теоремы принимает вид

$$u \geqslant 0, \qquad uA \geqslant 0 \Rightarrow ub \geqslant 0. \tag{4'}$$

b) Если удалить теперь условие неотрицательности вектора x и искать решение системы вида

$$\begin{aligned} Ax &\leqslant b, \\ x &\quad \text{любого знака}, \end{aligned} \tag{3''}$$

то аналог условия (4′) принимает вид

$$u \geqslant 0, \quad uA = 0 \Rightarrow ub \geqslant 0. \tag{4''}$$

Следствие 3 (теорема Гордана). *Пусть* A *— матрица* $m \times n$, $x \in \mathbf{R}^n$, $u^T \in \mathbf{R}^n$. *Тогда имеет место одно и только одно из следующих двух условий:*

1) *существует такой вектор* x, *что* $Ax < 0$;

2) *существует такой ненулевой вектор* u, *что* $uA = 0$ *и* $u \geqslant 0$.

Доказательство. Обозначим через $\mathbf{1}$ m-вектор, все компоненты которого равны 1. Условие 1) эквивалентно условию, что система

$$Ax \leqslant -\mathbf{1}$$

имеет решение $x \in \mathbf{R}^n$ (без ограничений на знак x).

Вследствие теоремы Фаркаша и Минковского это условие эквивалентно условию

$$u \geqslant 0, \qquad uA = 0 \Rightarrow -u \cdot \mathbf{1} \geqslant 0,$$

откуда и следует, что не существует ненулевого вектора u, удовлетворяющего условиям $uA = 0$ и $u \geqslant 0$.

Обратно, если существует такой $u \neq 0$, что $uA = 0$ и $u \geqslant 0$, то условие теоремы Фаркаша и Минковского не выполнено и, следовательно, система $Ax < 0$ не имеет решения.

СПИСОК ЛИТЕРАТУРЫ

1. Farkas J. (1901), Über die Theorie der einfachen Ungleichungen, *Journal für die reine und angewandte Mathematik* 124, p. 1, 27.
2. Minkowski H. (1896), *Geometrie der Zahlen*, Teubner, Leipzig & Berlin, 2e éd. 1910.

Приложение 2
СУЩЕСТВОВАНИЕ СЕДЛОВЫХ ТОЧЕК В ВЫПУКЛОМ МАТЕМАТИЧЕСКОМ ПРОГРАММИРОВАНИИ

Мы дадим здесь прямое доказательство теоремы 5 гл. 5, которая утверждает существование седловой точки для задачи выпуклого программирования вида

$$f(x) \to \min,$$
$$g_i(x) \leqslant 0, \quad i \in I = \{1, \ldots, m\},$$
$$x \in S.$$

Напомним формулировку теоремы.

Теорема. *Пусть $S \subset \mathbf{R}^n$ — замкнутое выпуклое множество, пусть f, g_i $(i \in I)$ — выпуклые функции и пусть множество решений задачи (P) имеет непустую внутренность, т. е. пусть множество $\{x \mid g_i(x) < 0 \quad \forall i \in I\}$ непусто. Тогда задача (P) имеет седловую точку $(\bar{x}, \bar{\lambda})$.*

Доказательство. Пусть $\bar{x}$ — оптимальное решение задачи (P). Рассмотрим множества

$$A = \{(y_0, y) \mid \exists x \in S: \ g(x) \leqslant y, \quad f(x) \leqslant y_0\},$$
$$B = \{(y_0, y) \mid y_0 \leqslant f(\bar{x}), \ y_i \leqslant 0 \qquad \forall i = 1, \ldots, m\}.$$

Множество A образовано всеми точками, расположенными над и на графике функции

$$\Phi(y) = \min \{f(x) \mid g(x) \leqslant y, \quad x \in S\};$$

согласно предположению, функция Φ выпукла, и, следовательно, множество A выпукло. Так как $f(\bar{x})$ есть минимум, то пересечение $A \cap \operatorname{int} B$ пусто, и так как

$\operatorname{int} B$ непусто, то существует гиперплоскость, разделяющая A и B (см. приложение 1, следствие 2 из теоремы 3), уравнение которой имеет вид

$$v_0 y_0 + vy = \text{const}, \qquad v = (v_1, \ldots, v_m).$$

v_0 — число.

Следовательно, имеем $\forall \begin{pmatrix} y_0 \\ y \end{pmatrix} \in A$, $\forall \begin{pmatrix} z_0 \\ z \end{pmatrix} \in B$:

$$v_0 y_0 + vy \geqslant v_0 z_0 + vz \tag{1}$$

и при этом необходимо имеем $(v_0, v) \geqslant 0$ (и по крайней мере одна компонента строго положительна).

Действительно, если бы хотя бы одна компонента была отрицательной, то можно было бы придать соответствующей компоненте z достаточно большое значение, для которого соотношение (1) не выполнялось бы.

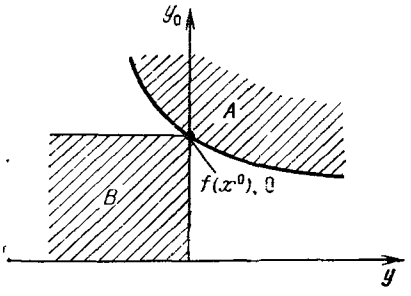


Рис. 1

Поскольку $[f(\bar{x}), 0] \in B$ и $[f(x), g(x)] \in A$ $\forall x \in S$, то мы имеем

$$v_0 f(x) + vg(x) \geqslant v_0 f(\bar{x}) \qquad \forall x \in S. \tag{2}$$

Если бы v_0 было равно нулю, то выполнялось бы соотношение

$$vg(x) \geqslant 0 \qquad \forall x \in S,$$

что невозможно (так как существует такое $x \in S$, что $g_i(x) < 0$). Следовательно, $v_0 > 0$.

Положим тогда $\bar{\lambda} = v/v_0$; очевидно, что $\bar{\lambda} \geqslant 0$, и из (2) вытекает, что

$$f(x) + \bar{\lambda} g(x) \geqslant f(\bar{x}) \qquad \forall x \in S; \tag{3}$$

полагая в (3) $x = \bar{x}$, мы видим, что

$$\bar{\lambda} g(\bar{x}) \geqslant 0.$$

Но одновременно имеем $\bar{\lambda} g(\bar{x}) \leqslant 0$ (поскольку $\bar{\lambda} \geqslant 0$ и $g(\bar{x}) \leqslant 0$). Отсюда следует, что $\bar{\lambda} g(\bar{x}) = 0$.

Добавляя $0 = \bar{\lambda} g(\bar{x})$ к правой части неравенства (3), видим, что

$$f(x) + \bar{\lambda} g(x) \geqslant f(\bar{x}) + \bar{\lambda} g(\bar{x}) \qquad \forall x \in S.$$

Следовательно,

$$L(x, \bar{\lambda}) \geqslant L(\bar{x}, \bar{\lambda}) \qquad \forall x \in S$$

(так что $\bar{x}$ минимизирует $L(x, \bar{\lambda})$ на S).

Вследствие теоремы 2 из п. 2.1 гл. 5, точка $(\bar{x}, \bar{\lambda})$ есть седловая точка для $L(x, \lambda)$.

Предыдущая теорема показывает, что для того, чтобы элемент $\bar{x}$ был точкой глобального оптимума задачи выпуклого программирования (P), необходимо и достаточно, чтобы существовало такое $\bar{\lambda}$, что точка $(\bar{x}, \bar{\lambda})$ является седловой точкой функции Лагранжа.

Приложение 3

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ В ЦЕЛЫХ ЧИСЛАХ

§ 1. Постановка задачи

Пусть $A = (a_i^j)$ — $m \times n$-матрица с целыми коэффициентами, имеющая ранг r; пусть $b = (b_i)$ $(i = 1, \ldots, m)$ — m-вектор (столбец) с целыми компонентами; пусть $x = (x_j)$ $(j = 1, \ldots, n)$ — вектор с целыми компонентами.

Поставим себе задачу решить (в целых числах) систему

$$\begin{aligned} Ax &= b, \\ x &\in \mathbf{Z}^n. \end{aligned} \tag{I}$$

Мы сейчас введем важнейшие математические инструменты, позволяющие решать системы вида (I): приведенную форму и нормальную форму Смита.

Затем мы покажем, что общее решение системы (I) можно представить в виде

$$x = x^0 + Wy,$$

где: W — нижняя треугольная $n \times (n-r)$-матрица, y — произвольный $(n-r)$-вектор с целыми компонентами, x^0 — частное решение задачи (I).

Однако нужно сразу же отметить, что все нижеследующее не позволяет решить систему (I) с учетом, кроме того, ограничений положительности $x \geqslant 0$. Этот последний тип задач на самом деле имеет отношение к методам целочисленного линейного программирования (см. гл. 7).

§ 2. Определения

Назовем *унимодулярной матрицей* квадратную матрицу с целыми элементами и определителем, равным $+1$ или -1.

Важное свойство матриц этого типа состоит в следующем. Если Q — унимодулярная матрица $(q \times q)$ и если y и z — два вектора (размерности q), связанные соотношением

$$y = Qz,$$

то вектор z является целочисленным тогда и только тогда, когда вектор y является целочисленным (ясно, что если z целочислен, то и y целочислен. Обратно, если y целочислен, то и $z = Q^{-1}y$ целочислен, так как Q^{-1} — тоже матрица с целыми элементами).

С другой стороны, запомним, что произведение произвольного числа унимодулярных матриц есть снова унимодулярная матрица.

Мы назовем *матрицей подстановки* квадратную матрицу вида

$$U_{(ij)} = \begin{matrix} \\ \\ i \\ \\ j \\ \\ \end{matrix} \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 0 & & 1 & & \\ & & & 1 & & & \\ & & 1 & & 0 & & \\ & & & & & 1 & \end{bmatrix}$$

Если A — произвольная квадратная матрица того же порядка, что и $U_{(i,j)}$, то умножение A на $U_{(i,j)}$ слева (спереди; иначе говоря, произведение $U_{(i,j)}A$) имеет результатом перестановку строк i и j матрицы A. Умножение A на $U_{(i,j)}$ справа (сзади; иначе говоря, $AU_{(i,j)}$) имеет результатом перестановку столбцов i и j в A.

(В каждом из этих случаев размерность матрицы $U_{(i,j)}$ должна выбираться согласованной с числом строк или столбцов матрицы A.)

Элементарной матрицей называется квадратная матрица вида

$$V(i,j,\alpha)=\begin{array}{c} \\ i \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccccc} & & & & & j & \end{array} \\ \begin{bmatrix} \ddots & & & & & \vdots & \\ & 1 & & & & \vdots & \\ \cdots & & 1 & \cdots & & -\alpha & \cdots \\ & & & 1 & & \ddots & \\ & & 0 & & \ddots & \vdots & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

Если A — произвольная матрица, то умножение A слева на $V(i, j, \alpha)$ имеет результатом вычитание из i-й строки матрицы A произведения α на j-ю строку матрицы A; умножение A на $V(i, j, \alpha)$ справа имеет результатом вычитание из j-го столбца α раз взятого i-го столбца. Для дальнейшего важно заметить, что матрицы подстановки и элементарные матрицы являются унимодулярными матрицами.

Две $(m \times n)$-матрицы A и B с целыми элементами называются *арифметически эквивалентными*, если существуют такие две унимодулярные матрицы P (порядка m) и Q (порядка n), что

$$B = PAQ.$$

Две арифметически эквивалентные матрицы имеют одинаковый ранг $r = \operatorname{rang}(A) = \operatorname{rang}(B)$. Кроме того, они имеют ряд интересных свойств. В частности, можно доказать, что для всех $h = 1, \ldots, r$ наибольшие общие делители всех миноров порядка h у обеих матриц одинаковы.

В дальнейшем мы увидим, что данной матрице с целыми элементами можно сопоставить арифметически эквивалентные ей матрицы, имеющие специальный вид (треугольный или диагональный). Затем мы увидим, как эти специальные формы используются при решении системы (1).

§ 3. Приведенные формы Эрмита

Для любой $m \times n$-матрицы A ранга r с целыми элементами существуют такие две унимодулярные матрицы P и Q, что

$$PAQ = \left[\begin{array}{c|c} L & 0 \\ \hline S & 0 \end{array}\right] = H$$

(приведенная форма *Эрмита*), где L — нижняя треугольная квадратная матри-

па порядка r и ранга r, а P — матрица подстановки. Этот результат доказывается конструктивно следующим образом).

а) Определим сначала две унимодулярные матрицы — матрицу подстановки P_1 и матрицу Q_1 — так, чтобы выполнялось соотношение

$$P_1 A Q_1 = \begin{array}{|c|c|} \hline d_1 & 0 \\ \hline \multirow{2}{*}{} & A_1 \\ \hline \end{array} = D_1$$

(в заштрихованной части элементы могут быть ненулевыми).

Перестановкой строк расположим строку, не равную нулю тождественно, на первой строке (это можно осуществить умножением слева на матрицу подстановки P_1).

Затем с помощью перестановки столбцов (чего можно достичь умножением на матрицу подстановки справа) расположим член с наименьшим (среди ненулевых членов строки) абсолютным значением на первом месте. Обозначим этот член a_1^1.

Умножая нашу матрицу справа на подходящие элементарные матрицы, можно заменить каждый член первой строки (отличный от a_1^1) его остатком (при делении на a_1^1 с остатком).

Если все остатки нулевые, то получим форму матрицы D_1 с $d_1 = a_1^1$.

В противном случае повторим с полученной матрицей все то, что мы делали выше со столбцами матрицы A. В результате конечного числа операций мы необходимо получим форму D_1, так как каждый из ненулевых членов первой строки, отличный от a_1^1, каждый раз заменяется членом, строго меньшим его по абсолютному значению. Единственные операции, осуществляемые над строками, суть перестановки (умножение слева на матрицу подстановки P_1).

b) Если форма D_1 уже получена и если $A_1 = 0$, то матрица A имеет ранг $r = 1$; тогда $H = D_1$ и есть приведенная форма Эрмита матрицы A.

Если же A_1 не равна нулю, то можно применить к A_1 преобразования, определенные в а), и получить матрицу вида

$$P_2 P_1 A Q_1 Q_2 = \begin{array}{|c|c|c|} \hline d_1 & 0 & 0 \\ \hline & d_2 & 0 \\ \hline & & A_2 \\ \hline \end{array} = D_2,$$

где P_2 — матрица подстановки.

Таким образом, мы получаем последовательность матриц A_1, A_2, ..., размерности которых на каждом шаге уменьшаются на единицу. Если $r = \operatorname{rang}(A)$, то мы, таким образом, необходимо получаем через r шагов матрицу $A_r = 0$ и, тем самым, приведенную форму Эрмита в виде

$$PAQ = \begin{pmatrix} d_1 & & & 0 & 0 \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_r & \\ & & & & 0 \end{pmatrix}$$

где

$$P = P_r P_{r-1} \dots P_2 P_1 \quad \text{(матрица подстановки)}$$

и

$$Q = Q_1 \dots Q_r.$$

Заметим, что можно совершенно не вмешивать в это дело матрицу подстановки P, т. е. не менять в предыдущем построении порядка следования строк. Отсюда мы могли бы вывести существование такой унимодулярной матрицы Q, что либо

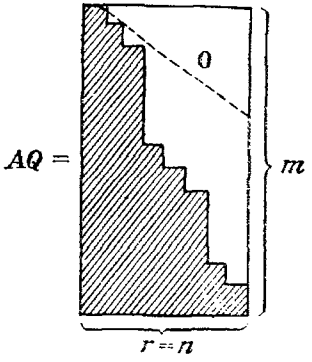
$$AQ = \begin{matrix} \\ m \end{matrix} \overbrace{\begin{pmatrix} & & \\ & & 0 \\ & & \\ \underbrace{\qquad}_{r} & & \end{pmatrix}}^{n} = W, \quad m \leqslant n,$$

либо

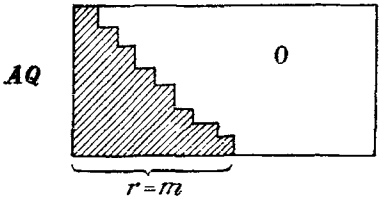
$$AQ = \begin{matrix} \\ m \end{matrix} \begin{pmatrix} & & \\ & & 0 \\ & & \\ \underbrace{\qquad}_{r} & & \end{pmatrix} = W, \quad m \geqslant n.$$

Мы видим, что W — треугольная (в широком смысле) матрица, т. е. все ее ненулевые элементы расположены не выше главной диагонали, выходящей из (1, 1) (она на рисунках изображена пунктиром).

В частности, если A — матрица полного ранга и если $r = n \leqslant m$, то мы получаем матрицу с конфигурацией вида



Если же A — полного ранга и $r = m \leqslant n$, то получаем матрицу с конфигурацией вида



(заметим здесь, что на главной диагонали необходимо должны стоять ненулевые элементы).

В заключение заметим, что данная матрица A может иметь несколько приведенных форм Эрмита, а матрицы P и Q не определены однозначно.

§ 4. Приведенные формы Смита

Для любой $(m \times n)$-матрицы A ранга r с целыми элементами существуют такие унимодулярные матрицы P и Q, что

$$PAQ = \left(\begin{array}{cc} \begin{matrix} d_1 & & \\ & d_2 & 0 \\ & \ddots & \\ 0 & & d_r \end{matrix} & \begin{matrix} 0 \\ \\ \\ \end{matrix} \\ \begin{matrix} 0 & \cdots & 0 \end{matrix} & 0 \end{array}\right) = D$$

(*приведенная форма Смита*).

Этот результат доказывается конструктивно, и способ его доказательства совершенно аналогичен способу, который в предыдущем параграфе был использован для приведенной формы Эрмита.

а) Перестановками строк и столбцов поместим элемент с наименьшим (среди ненулевых элементов) модулем в левый верхний угол (1, 1). Пусть этот элемент есть a_1^1. Заменим тогда каждый элемент первой строки, отличный от a_1^1, остатком от деления этого элемента на a_1^1 (это сводится к умножению справа на элементарные матрицы), а затем каждый элемент первого столбца (отличный от a_1^1) — остатком от его деления на a_1^1 (это сводится к умножению слева на элементарные матрицы).

Повторим все предыдущее с новой, только что полученной матрицей — до тех пор, пока все остатки и в первой строке, и в первом столбце не окажутся нулями.

Таким образом, за конечное число таких операций мы необходимо получим две такие унимодулярные матрицы P и Q, что

$$P_1AQ_1 = \left|\begin{array}{c|c} d_1 & 0 \\ \hline 0 & A_1 \end{array}\right| = D_1.$$

b) Если $A_1 = 0$, то матрица A имеет ранг $r = 1$, и матрица D_1 является приведенной формой Смита матрицы A.

Если $A_1 \neq 0$, тогда можно заново применить к A_1 все преобразования, определенные в а), и определить такие матрицы P_2 и Q_2, что

$$P_2P_1AQ_1Q_2 = \left|\begin{array}{c|c|c} d_1 & 0 & 0 \\ \hline 0 & d_2 & 0 \\ \hline 0 & 0 & A_2 \end{array}\right| = D_2.$$

Если $r = \operatorname{rang}(A)$, то через r шагов необходимо получим $A_r = 0$, и приведенная форма Смита получается таким образом сама собой в виде

$$PAQ = \left|\begin{array}{ccc|c} d_1 & & & \\ & d_2 & 0 & 0 \\ & \ddots & & \\ & 0 & d_r & \\ \hline \multicolumn{3}{c|}{0} & 0 \end{array}\right| = D$$

с унимодулярными P и Q.

Как и для приведенных форм Эрмита, приведенная форма Смита не единственна; то же можно сказать о матрицах P и Q.

Заметим, что, в отличие от случая приведенной формы Эрмита, здесь матрица P уже не обязательно является матрицей подстановки.

§ 5. Нормальная форма Смита

Для любой $(m \times n)$-матрицы A с целыми коэффициентами существуют такие две унимодулярные матрицы P и Q, что

$$PAQ = \left(\begin{array}{c|c} \begin{matrix} \varepsilon_1 & & 0 \\ & \varepsilon_2 & \\ & & \ddots \\ 0 & & \varepsilon_r \end{matrix} & \begin{matrix} 0 \\ \\ \\ \\ \end{matrix} \\ \hline 0 & 0 \end{array}\right) = S,$$

где ε_i является делителем числа ε_{i+1} для всех $i = 1, \ldots, r-1$ (*нормальная форма Смита*).

Кроме того, этим условием матрица S определена однозначно.

И в этом случае доказательство нашего результата конструктивно и совершенно аналогично предыдущим доказательствам.

Так как при получении приведенной формы Смита мы уже определили (на шаге а)) две такие унимодулярные матрицы P_1 и Q_1, что

$$P_1AQ_1 = \left(\begin{array}{c|c} d_1 & 0 \\ \hline 0 & A_1 \end{array}\right) = D_1,$$

то мы ими можем воспользоваться. Если d_1 является делителем всех элементов матрицы A_1, то $\varepsilon_1 = d_1$ и можно продолжать операции над подматрицей A_1.

Если же d_1 не является делителем одного из элементов матрицы A_1, то пусть для определенности номер строки, в которой этот элемент расположен, равен i. Тогда мы можем прибавить строку i к строке 1 (умножением на матрицу $V(1, i, -1)$ слева) и заново провести шаг а) для полученной таким образом новой матрицы. В результате появится новый элемент d_1, который будет тогда строго меньше предыдущего, и за конечное число шагов мы обязательно получим такие P_1 и Q_1, что d_1 является делителем всех ненулевых элементов матрицы A_1. Тогда будем иметь $d_1 = a_1$, и все нужно повторить для подматрицы A_1.

Если $\operatorname{rang}(A) = r$, то за r таких шагов мы получим проведенную форму Смита S.

Единственность натуральных чисел ε_i доказывается с помощью наблюдения, что $\Delta_h = \varepsilon_1 \times \ldots \times \varepsilon_h$ есть наибольший общий делитель всех миноров порядка h матрицы S (мы напоминаем, что для любого h наибольшие общие делители всех миноров порядка h у этих двух матриц одни и те же).

§ 6. Пример вычисления нормальной формы Смита

Доказательства, приведенные в § 3—5, позволяют легко построить алгоритмы получения приведенной или нормальной формы данной матрицы A.

Рассмотрим пример, показывающий, как работает алгоритм получения нормальной формы Смита.

Пусть дана 3×3-матрица

$$\begin{bmatrix} 3 & 6 & -12 \\ 0 & 4 & -5 \\ -9 & -6 & 18 \end{bmatrix}.$$

Ненулевой элемент с наименьшим абсолютным значением уже находится в положении (1, 1).

Умножая на шаге (A) данную матрицу на элементарную матрицу справа, мы получаем нули в местах (1, 2) и (1, 3).

Затем умножением слева на элементарную матрицу мы на шаге (B) получаем нули в местах (2, 1) и (3, 1).

На этом этапе элемент 3 в (1, 1) не является делителем всех элементов подматрицы, образованной элементами (2, 2), (2, 3), (3, 2) и (3, 3). Следовательно, прибавляем к первой строке вторую (которая содержит элемент, не делящийся на 3). Это осуществляется на шаге (C) умножением слева на элементарную матрицу.

На шаге (D) умножение справа на элементарную матрицу позволяет заменить элементы 4 и —5 их остатками от деления на 3.

Теперь, на шаге (E), ненулевой элемент с наименьшим абсолютным значением равен 1. Чтобы поместить его в (1, 1), умножаем справа на матрицу подстановки. На шаге (F) умножение справа на элементарную матрицу позволяет получить нули в положениях (1, 2) и (1, 3). Затем в (G) умножение слева на элементарную матрицу позволяет получить нули в первом столбце, кроме элемента в (1, 1).

Умножение справа на элементарную матрицу в (H) позволяет поместить ненулевой элемент оставшейся подматрицы с наименьшим модулем на место (2, 2).

Умножение справа на элементарную матрицу в (I), а затем умножение слева на элементарную матрицу в (J) позволяет, наконец, получить нормальную форму Смита исходной матрицы.

Мы можем получить матрицы P и Q, осуществляя умножение всех матриц, на которые выполнялось умножение слева, с одной стороны, и всех матриц, на которые выполнялось умножение справа, — с другой.

§ 7. Приложение к решению систем линейных уравнений в целых числах

Вернемся теперь к решению системы

$$\begin{aligned} Ax &= b, \\ x &\in \mathbf{Z}^n, \end{aligned} \tag{I}$$

где $(m \times n)$-матрица A имеет ранг r.

Матрица левого умножения (операция над строками)

Полученная матрица

Матрица правого умножения (операция над столбцами)

$$A)\quad \begin{bmatrix} 3 & 6 & -12 \\ 0 & 4 & -5 \\ -9 & -6 & 18 \end{bmatrix} \times \begin{bmatrix} 1 & -2 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$B)\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & -5 \\ -9 & 12 & -18 \end{bmatrix} =$$

$$C)\quad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & -5 \\ 0 & 12 & -18 \end{bmatrix} =$$

$$D)\quad \begin{bmatrix} 3 & 4 & -5 \\ 0 & 4 & -5 \\ 0 & 12 & -18 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & +1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$E)\quad \begin{bmatrix} 3 & 1 & -2 \\ 0 & 4 & -5 \\ 0 & 12 & -18 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$F)\quad \begin{bmatrix} 1 & 3 & -2 \\ 4 & 0 & -5 \\ 12 & 0 & -18 \end{bmatrix} \times \begin{bmatrix} 1 & -3 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$G)\quad \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -12 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 4 & -12 & 3 \\ 12 & -36 & 6 \end{bmatrix} =$$

$$H)\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & -12 & 3 \\ 0 & -36 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} =$$

$$I)\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & -12 \\ 0 & 6 & -36 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$J)\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 6 & -12 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -12 \end{bmatrix}$$

Это и есть нормальная форма Смита исходной матрицы

Пусть D — приведенная форма Смита матрицы

$$PAQ = \begin{array}{c} \\ R \\ \\ \\ \bar{R} \end{array} \overbrace{\phantom{\begin{matrix} d_1 & & \\ & \ddots & \\ & & d_r \end{matrix}}}^{R} \overbrace{\phantom{\begin{matrix} 0 \\ \\ 0 \end{matrix}}}^{\bar{R}} \begin{pmatrix} d_1 & & & 0 \\ & d_2 & \ddots & \\ & & \ddots & \\ 0 & & d_r & \\ \hline & 0 & & 0 \end{pmatrix} = D,$$

где $(m \times m)$-матрица P и $(n \times n)$-матрица Q — унимодулярные матрицы. Введем обозначения:

R — множество индексов $\{1, \ldots, r\}$;

$\bar{R}$ — дополнение R в $\{1, \ldots, n\}$;

$\bar{\bar{R}}$ — дополнение R в $\{1, \ldots, m\}$;

D_R^R — подматрица D, образованная r первыми строками и r первыми столбцами D.

Система $Ax = b$ может быть (поскольку матрица Q обратима) переписана в эквивалентном виде

$$PAQQ^{-1}x = Pb.$$

Пусть, кроме того, $DQ^{-1}x = Pb$.

Положим $y = Q^{-1}x$. (Так как матрица Q унимодулярна, то целочисленность x равносильна целочисленности y.)

Получаем

$$Dy = Pb,$$

y целочислен.

Это отношение для r первых строк переписывается в виде

$$D_R^R y_R = P_R b$$

(y_R — это r первых компонент вектора y, P_R — r первых компонент вектора P), а для оставшихся строк получаем в качестве следствия

$$0 = P_{\bar{\bar{R}}} b$$

(здесь $P_{\bar{\bar{R}}}$ — матрица из $m - r$ последних строк P).

С другой стороны, вектор $y_{\bar{R}}$ (образованный последними $n - r$ компонентами y) может быть выбран произвольно.

Отсюда следует, что необходимое и достаточное условие разрешимости системы уравнений $Ax = b$ в целых числах состоит в следующем:

$$\left(D_R^R\right)^{-1} P_R b \text{ — целочисленный вектор,}$$

$$P_{\bar{\bar{R}}} b = 0.$$

При этих условиях общее решение системы (1) имеет вид

$$x = Qy = Q^R y_R + Q^{\bar{R}} y_{\bar{R}}. \qquad (2)$$

Таким образом,

$$x = x^0 + Q^{\overline{R}} y_{\overline{R}},$$

$y_{\overline{R}}$ — произвольный целочисленный вектор,

где $x^0 = Q^R \left(D_R^R\right)^{-1} P_R b$ есть (целочисленное) частное решение системы и где $Q^{\overline{R}} y_{\overline{R}}$ ($y_{\overline{R}}$ — произвольный целочисленный вектор) есть общее решение системы уравнений с нулевой правой частью.

Заметим, что предыдущую конструкцию можно осуществить, исходя из любой приведенной формы Смита — не обязательно нормальной. Совершенно аналогичные соотношения можно получить, используя приведенную форму Эрмита.

Наконец, всегда можно выразить общее решение задачи (I) в виде

$$x = x^0 + Wz,$$

где z есть произвольный целочисленный $(n-r)$-вектор, а W — нижняя треугольная матрица.

Для этого достаточно рассмотреть в (2) приведенную форму Эрмита матрицы $Q^{\overline{R}}$.

Действительно, в § 3 мы видели, что для $(n \times (n-r))$-матрицы $Q^{\overline{R}}$ полного ранга существует такая унимодулярная $((n-r) \times (n-r))$-матрица U, что

$$Q^{\overline{R}} U = \begin{array}{|c|} \hline \\ \\ \\ \\ \hline \end{array} \quad \begin{array}{l} \\ \\ \\ \\ \end{array} \quad \begin{array}{l} 0 \\ \\ \\ \\ \end{array} = W.$$

Тогда можно записать равенство

$$Q^{\overline{R}} y_{\overline{R}} = Q^{\overline{R}} U U^{-1} y_{\overline{R}} = Wz,$$

где $z = U^{-1} y_{\overline{R}}$ — *произвольный целочисленный* $(n-r)$-вектор (так как матрица U унимодулярна, то $y_{\overline{R}}$ целочислен тогда и только тогда, когда вектор z целочислен).

СПИСОК ЛИТЕРАТУРЫ

1. Chatelet A. (1967-1968), *Arithmétique et algèbre modernes*, Presses Universitaires de France, Paris.
2. Fiorot J. C., Gondran M. (1969), Résolution des systèmes linéaires en nombres entiers, *Bulletin de la Direction des Etudes et Recherches EDF*, Série C, n° 2.
3. Kannan R., Bachem A. (1978), "Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix". Report 7895-OR, Institut für Ökonometrie und Operations Research, Univ. Bonn.

Приложение 4

ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ: ОЦЕНКИ СНИЗУ И ПРИБЛИЖЕННЫЕ РЕШЕНИЯ С ПОМОЩЬЮ ЛАГРАНЖЕВА ОСЛАБЛЕНИЯ И РЕШЕНИЯ ДВОЙСТВЕННОЙ ЗАДАЧИ

Как уже указывалось в § 2.5 гл. 7, одним из наиболее эффективных технических приемов получения хороших функций оценки для решения целочисленных задач программирования с помощью методов разветвления является метод, состоящий в решении полученной двойственной задачи с помощью лагранжева ослабления некоторого количества ограничений исходной задачи. Так как можно удовлетвориться не точным решением, а хорошей аппроксимацией оптимального значения двойственной задачи, то можно особенно рекомендовать применение алгоритмов субградиента (см. гл. 4, § 3) вследствие простоты их работы и эффективности этих алгоритмов.

Действительно, практика показывает, что для очень многих задач комбинаторной оптимизации оказывается, что

— двойственная функция хорошо обусловлена, что позволяет получить удовлетворительную скорость сходимости (см. гл. 4, § 3 относительно изучения скорости сходимости в зависимости от параметра обусловленности оптимизируемой функции);

— скачки двойственности (разности между целочисленным оптимумом и оптимумом двойственной задачи) очень малы.

Отсюда мы и получаем возможность получить во многих задачах хорошие функции оценки при относительно небольших затратах.

В этом разделе мы намереваемся дать обзор основных классов комбинаторных задач, в которых этот общий подход может быть успешно применен к нахождению точных решений задач часто весьма большого размера, которые до применения этого метода считались очень трудными.

Мы покажем также, что важность технических приемов лагранжева ослабления не ограничивается определением хороших минорант для отыскания точных решений методами разветвления, описанными в гл. 7, § 2. Мы увидим, что они часто позволяют также получить — в ходе оптимизации двойственной задачи — хорошие приближенные решения. С другой стороны, информация, полученная при решении двойственной задачи (двойственные переменные) часто позволяет преобразовать данные исходной задачи. Например, можно определить величины, которые по аналогии с задачами линейного программирования можно называть *приведенными ценами*. Так как эти приведенные цены в гораздо большей степени, чем исходные цены, позволяют учесть структуру задачи и ее ограничения, то приближенные решения, полученные с помощью эвристических методов даже элементарного характера, в применении к приведенным ценам дают часто гораздо лучшие результаты, чем все то, что удается получить с помощью даже много более сложных эвристических методов в применении к исходным ценам. Таким образом, мы приходим к понятию исходно-двойственного эвристического метода, который был успешно применен во многих задачах: задаче о коммивояжере [61], в задачах покрытия [6], в задачах автоматической классификации и локализации [64, 3].

Использование техники лагранжева ослабления и комбинаторной оптимизации было впервые осуществлено в работах [11 и 41] в применении к задаче о коммивояжере. Этот пример, одновременно очень характерный и представляющий «исторический» интерес, мы разберем в первую очередь.

§ 1. Задача о коммивояжере. Ориентированный и неориентированный случай

Пусть $G = [X, U]$ — ориентированный связный граф, где X означает множество вершин (их число $|X|$ равно N), а U — множество дуг (их число $|U|$ равно M). Здесь мы ограничимся напоминанием нескольких наиболее плодотворных в дальнейшем изложении понятий теории графов. За подробностями мы отсылаем читателя, например, к [47, 32]. Любая *дуга* $u \in U$ есть упорядоченная пара вершин $u = (i, j)$, в которой i называется *начальной* вершиной, а j — *конечной* вершиной дуги u. *Путем* называется такая последовательность дуг $\{u_1, u_2, \ldots, u_p\}$, что для всех $k = 1, \ldots, p-1$ конечная вершина дуги u_k совпадает с начальной вершиной дуги u_{k+1}. *Циклом* называется путь $\{u_1, u_2, \ldots, u_p\}$, для которого начальная вершина дуги u_1 совпадает с конечной вершиной дуги u_p.

Элементарным циклом называется цикл, удовлетворяющий следующим двум условиям:

(i) для каждой вершины j существует не более одной дуги, инцидентной этой вершине внешне (т. е. такой дуги, что j — ее начальная вершина);

(ii) для каждой вершины j существует не более одной дуги, инцидентной этой вершине внутренне (т. е. такой дуги, что j — ее конечная вершина).

Гамильтонов цикл есть элементарный цикл, проходящий по всем вершинам графа (или, что равносильно предыдущему,— это элементарный цикл, содержащий в точности N дуг).

Предположим, что каждой дуге $u = (i, j) \in U$ в графе G сопоставлено число l_{ij} (произвольного знака), представляющее собой длину пробега от i до j (в зависимости от рассматриваемой модели эти числа l_{ij} могут также представлять время пробега или стоимость перехода от i к j и т. д.). *Задача коммивояжера* (в ориентированном случае) состоит тогда в определении гамильтона цикла минимальной полной длины в графе G (длина цикла определяется при этом как сумма длин дуг, составляющих этот цикл). *Неориентированная версия* задачи о коммивояжере соответствует частному случаю, в котором для любой дуги (i, j) между двумя вершинами i и j графа, входящей в граф, дуга (j, i) также входит в граф и имеет ту же длину: $l_{ij} = l_{ji}$. В этом случае можно «забыть» об ориентации дуг, и задача о коммивояжере становится задачей поиска гамильтонова цикла минимальной полной длины в неориентированном графе. Задача о коммивояжере имеет — как в ориентированном, так и в неориентированном варианте — многочисленные применения: поездка по местам распределения, контроль за ходом выполнения заказа промышленной продукции, прокладка кабеля в электрических цепях и т. д. (см. [32, 48]).

Для этой задачи на самом деле не известен никакой полиномиальный алгоритм (т. е. такой, чтобы время вычисления было ограничено некоторой полиномиальной функцией от размерности задачи), так что эта задача принадлежит классу трудных комбинаторных задач — так называемых NP-полных задач (см. [26]).

Чтобы показать, как к этой задаче может быть применено лагранжево ослабление, мы используем постановку задачи, предложенную в [8], которая является несколько более общей, чем постановка, предложенная первоначально в [41], поскольку включает в себя одновременно и ориентированный и неориентированный случай.

Сразу же заметим, что мы можем без ограничения общности предполагать, что граф G является полным: действительно, мы получим задачу, равносильную исходной, если присвоим каждой дуге (i, j), не входившей в исходный граф, длину $l_{ij} = +\infty$. Тогда получаем, что $M = N(N-1)$.

Свяжем с каждой дугой $u = (i, j)$ графа G целочисленное переменное x_{ij}, которое может принимать не более двух значений 0 и 1 и имеет следующий смысл: $x_{ij} = 1$ тогда и только тогда, когда дуга (i, j) содержится в искомом гамильтоновом цикле; в противном случае $x_{ij} = 0$. Для всех $x = (x_{ij}) \in \in \{0, 1\}^M$ обозначим через $\overline{U}(x)$ множество $\{u | u \in U, x_u = 1\}$.

Заметим тогда, что если $x = (x_{ij}) \in \{0, 1\}^M$ является характеристическим вектором некоторого гамильтонова цикла в G, то необходимо должны выполняться следующие три условия:

(C)

$$\forall i \in X: \quad \sum_{\substack{j=\overline{1,N} \\ j \neq i}} x_{ij} = 1, \tag{C1}$$

$$\forall j \in X: \quad \sum_{\substack{i=\overline{1,N} \\ i \neq j}} x_{ij} = 1, \tag{C2}$$

$$\text{подграф } \overline{G}_{(x)} = [X, \overline{U}(x)] \text{ связен.} \tag{C3}$$

Условия (C1) и (C2) являются воспроизведением приведенных выше условий (i) и (ii), и эти условия необходимы для того, чтобы x был элементарным циклом, проходящим один и только один раз через каждую из вершин графа. Чтобы убедиться, что сами по себе условия (C1) и (C2) недостаточны, мы можем рассмотреть 0-1-вектор $x = (x_{ij})$, носителем которого является множество из p дизъюнктных циклов ($p \geqslant 2$), покрывающих все вершины графа (рис. 1). Такой вектор удовлетворяет условиям (C1) и (C2), но не соответствует гамильтонову циклу, и в этом случае мы убеждаемся, что не выполняется условие связности. Обратно, легко доказать, что любой вектор x, удовлетворяющий условиям (C1), (C2) и условию связности (C3), необходимо является гамильтоновым циклом.

В дальнейшем мы будем использовать слегка измененную форму условия (C). Заметим сначала, из условий (C1) и (C2) следует, что $\sum_{\substack{i,j \\ i \neq j}} x_{ij} = N$. С другой стороны, заметим, что для произвольно выбранной вершины — например, для вершины 1 — граф $\overline{G}(x)$ должен содержать в точности две различные дуги, инцидентные вершине 1.

Тогда условие (C3) можно заменить следующим условием:

«Подграф $\overline{G}(x) = [X, \overline{U}(x)]$ связен и содержит N дуг, среди которых в точности две различные дуги инцидентны вершине 1». (C'3)

Пусть G — связный граф с N вершинами. Подграф такого графа называется 1-деревом [41], если этот подграф связен и содержит в точности N дуг. Такой граф приведен на рис. 2; там он образован деревом (т. е. связным подграфом без циклов), покрывающим вершины от 2 до 8, к которому присоединены две различные дуги, инцидентные вершине 1. Следовательно, 1-дерево содержит в точности один-единственный цикл (так что цикломатическое число 1-дерева равно в точности единице, см., например, [32], гл. 4). Заметим,

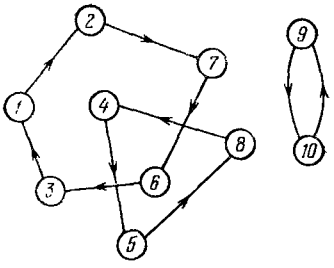


Рис. 1. В этом решении выполняются все ограничения о степени вершин (для каждой вершины существует внутренне инцидентная дуга и внешне инцидентная дуга), но это не гамильтонов цикл, так как условие связности не выполняется

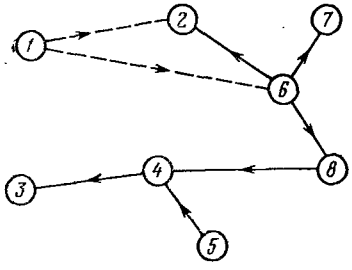


Рис. 2. Множество дуг, обозначенных сплошными стрелками, образует связный граф, не имеющий циклов (т. е. дерево), покрывающий вершины от *2* до *8*. Присоединяя две дуги, инцидентные вершине *1* (они обозначены на рисунке пунктиром), мы порождаем единственный цикл и получаем 1-дерево. Заметим, что в предыдущем определении ориентация дуг не участвует

что в понятии дерева, как и в понятии 1-дерева, ориентация дуг не играет никакой роли.

Если мы обозначим через τ_1 множество характеристических векторов 1-деревьев графа G, содержащих в точности две различные дуги с вершиной 1, то мы сможем следующим образом сформулировать нашу задачу о коммивояжере:

$$\text{(TSP)} \qquad \begin{aligned} &\sum_{i,j=1}^{N} l_{ij}x_{ij} \to \min, \\ &\sum_{\substack{j=1 \\ j\neq i}}^{N} x_{ij} = 1 \qquad \forall i \in x, \qquad (1) \\ &\sum_{\substack{i=1 \\ i\neq j}}^{N} x_{ij} = 1 \qquad \forall j \in x, \qquad (2) \\ &x \in \tau_1. \qquad (3) \end{aligned}$$

В этой форме задача хорошо приспособлена к лагранжеву ослаблению. Действительно, свяжем с ограничениями (1) дуальные переменные (множите-

ли Лагранжа) λ_i, а с ограничениями (2) — дуальные переменные μ_j (и те и другие — без ограничений на знак). Тогда функция Лагранжа может быть записана в виде

$$L(x, \lambda, \mu) = \sum_i \sum_{j \neq i} l_{ij} + \sum_i \lambda_i \left(\sum_{j \neq i} x_{ij} - 1 \right) +$$
$$+ \sum_j \mu_j \left(\sum_{i \neq j} x_{ij} - 1 \right) = - \sum_i \lambda_i - \sum_j \mu_j + \sum_i \sum_{j \neq i} (l_{ij} + \lambda_i + \mu_j) x_{ij}.$$

Для фиксированных λ и μ соответствующее значение двойственной функции $w(\lambda, \mu)$ определяется тогда с помощью формулы

$$w(\lambda, \mu) = - \sum_i \lambda_i - \sum_j \mu_j + \min_{x \in \tau_1} \left\{ \sum_i \sum_{j \neq i} (l_{ij} + \lambda_i + \mu_j) x_{ij} \right\} \qquad (4)$$

и сводится, таким образом, к определению 1-дерева минимальной полной длины, где каждая дуга $u = (i, j)$ графа G снабжена длиной $\bar{l}_{ij} = l_{ij} + \lambda_i + \mu_i$ (эти величины $\bar{l}_{ij}$ называются *приведенными длинами* и играют роль, аналогичную роли приведенных цен в линейном программировании; см. гл. 2, п. 1.6).

Заметим теперь, что 1-дерево минимальной приведенной длины можно получить следующим образом:

а) найти дерево минимальной приведенной длины, связывающее все вершины из $X \backslash \{1\}$;

б) добавить две различные дуги наименьшей приведенной длины, инцидентные вершине 1.

Мы видим, что вычисление значения двойственной функции $w(\lambda, \mu)$ для данных λ и μ по существу сводится к определению минимального дерева в графе. Для этой задачи существует много очень эффективных алгоритмов (см. [66, 74]).

Замечание. В симметричном случае, когда выполняется соотношение $l_{ij} = l_{ji}$ для всех $(i, j) \in U$, ориентация дуг не влияет на решение, и вместо того чтобы налагать условия (1) и (2) по отдельности, достаточно удостовериться, что совершенно для каждой вершины существуют в точности две инцидентные ей дуги (какова бы ни была ориентация дуг). Тогда ограничения (1) и (2) могут быть заменены единственным условием:

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ij} + \sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ji} = 2 \qquad \forall i \in X.$$

Тогда, с учетом N ограничений о степенях вершин, мы снова получаем постановку симметричной задачи о коммивояжере, изученную в [41]. В этом случае двойственная задача использует пространство двойственных переменных размерности N (вместо $2N$), и интересно заметить, что эта задача может рассматриваться — по отношению к общей двойственной задаче, определенной выше,— как частный случай этой задачи, в котором $\lambda_i = \mu_i$ для всех $i \in X$.

Обозначим через $\bar{x}(\lambda, \mu)$ характеристический вектор, соответствующий 1-дереву минимальной приведенной длины (т. е. относительно длин $\bar{l}_{ij} = l_{ij} + \lambda_i + \lambda_j$), определенный в (4) в ходе вычисления $w(\lambda, \mu)$. Тогда мы можем получить субградиент двойственной функции w в точке (λ, μ): координата

этого субградиента, соответствующая переменному λ_i, равна

$$\sum_{\substack{j=1\\ j\neq i}}^{N} \bar{x}_{ij}(\lambda, \mu) - 1 \qquad \forall i \in X,$$

а координата, соответствующая переменному μ_j, равна

$$\sum_{\substack{i=1\\ i\neq j}}^{N} \bar{x}_{ij}(\lambda, \mu) - 1 \qquad \forall j \in X.$$

Следовательно, к решению двойственной задачи

$$\text{(DTSP)} \qquad \begin{aligned} w(\lambda, \mu) &\to \max, \\ (\lambda, \mu) &\in \mathbf{R}^{2N} \end{aligned}$$

можно применить один из технических приемов оптимизации вогнутых не всюду дифференцируемых функций, описанных в гл. 4, § 3, например:

— метод сходящихся рядов (гл. 4, п. 3.6),

— метод релаксации (гл. 4, п. 3.7),

— метод ε-спуска (гл. 4, п. 3.10).

Вычислительные эксперименты, проведенные в [41] с симметричными задачами для графов, содержащих до 64 вершин, показывают, что скачки двойственности очень малы и редко превосходят 1—2 %. Тогда, оптимизируя двойственную задачу с помощью некоторой процедуры субградиента (см. гл. 4, п. 3.7, и [42]), мы можем получить очень хорошие оценки снизу, что позволяет определять точные оптимальные решения, порождая очень небольшие по объему разветвления (несколько десятков вершин) даже для графов довольно большого размера (от 50 до 100 вершин).

Вычислительные эксперименты, рассматриваемые в [8], показывают, что в несимметричном случае скачки двойственности могут оказаться существенно большими, и полученные оценки снизу дают отклонения от оптимальности до 3—4 %. Тем не менее и в этом случае рассматриваемые методы позволяют найти точное решение задач относительно большого размера.

В заключение заметим, что недавно были достигнуты существенные успехи в решении задачи о коммивояжере с помощью объединения методов лагранжева ослабления с методами сечений (см. гл. 7, § 3). Идея этого объединения состоит в присоединении к описанной выше задаче (TSP) дополнительных ограничений, соответствующих граням выпуклой оболочки целочисленных решений. Использование этих ограничений — или сечений — не исключает, таким образом, ни одного целого решения, но, с другой стороны, приводит к дополнительному — и довольно заметному — уменьшению скачков двойственности и, таким образом, к получению улучшенных функций оценки. Хотя мы и не знаем на самом деле полного описания выпуклой оболочки множества целых точек для задачи о коммивояжере (как и для большинства трудных комбинаторных задач), но некоторое число ограничений — например, соответствующих граням этой выпуклой оболочки,— может быть определено (см. [33, 34, 35]). С помощью этого подхода могут быть найдены точные решения задач очень большой размерности: 318 вершин в симметричном случае [13] и 325 вершин в несимметричном случае [5]. Эти успехи могут рассматриваться

как первые важные результаты нового направления исследований в целочисленном программировании («полиэдральная комбинаторика»), одной из целей которой является получение для каждого типа комбинаторных задач максимально возможной полноты в характеризации выпуклой оболочки целых решений. Обзоры по полиэдральной комбинаторике можно найти в [4, 67].

§ 2. Задачи локализации. Автоматическая классификация

Сначала мы введем на некотором примере достаточно общую модель для задач локализации, а затем более подробно изучим приложение техники лагранжева ослабления к задаче простой локализации. Задачи локализации возникают во многих прикладных вопросах: оптимизация по числу и по расположению множества заводов, множества складов, множества телефонных станций городской телефонной сети и т. д. Затем мы изучим приложения к вопросам автоматической классификации, т. е. разбиения n объектов на k классов таким образом, чтобы минимизировать некоторый критерий их взаимной несогласованности. Типичная задача локализации может быть поставлена следующим образом. Предприятие по производству продукции ищет способ расположения оптимального множества центров по распределению продукции (складов) с целью обслужить с наименьшей стоимостью данное множество клиентов.

Обозначим через $J = \{1, 2, \ldots, n\}$ множество клиентов, а через $I = \{1, 2, \ldots, m\}$ — множество возможных мест для расположения склада. Предположим, что известны фиксированные цены f_i помещения склада емкости Q_i в каждое возможное место $i \in I$, а также требования d_j клиентов $j \in J$. Наконец, мы предположим, что для каждого клиента $j \in J$ известны цены распределения c_{ij} требуемых d_j единиц продукции, исходя со склада, расположенного на месте $i \in I$. Обозначим через x_{ij} долю требования клиента j, предоставляемую со склада i, и определим m двузначных переменных y_i, имеющих следующий смысл: $y_i = 1$, если склад (с емкостью Q_i) расположен на месте $i \in I$, и $y_i = 0$ в противном случае. Тогда подлежащая решению задача выглядит следующим образом:

$$\text{(LOC)} \qquad \begin{aligned} &\sum_{i \in I} f_i y_i + \sum_{i \in I} c_{ij} x_{ij} \to \min, \\ &\sum_{j \in J} d_j x_{ij} \leqslant Q_i y_i \qquad \forall i \in I, \qquad (5) \\ &\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J, \qquad (6) \\ &\forall i \in I\colon\ y_i = 0 \text{ или } 1, \\ &0 \leqslant x_{ij} \leqslant 1. \end{aligned}$$

Заметим относительно этой задачи, что если переменные y_i фиксированы и если на переменные x_{ij} наложено дополнительно ограничение целочисленности, то мы приходим к обобщенной задаче об ассигнованиях, изученной ниже в § 7. Поставленная сейчас задача (LOC) может быть во многих случаях упрощена и, в частности, в случае, когда можно пренебречь ограничениями

емкости (5) (например, если априори известно, что емкость складов окажется достаточной). Тогда для любого $i \in I$ ограничение (5) можно представить в виде

$$x_{ij} \leqslant y_i \qquad \forall j \in J,$$

которое выражает тот простой факт, что клиента j можно связать с местом i только в случае, если $y_i = 1$, т. е. только если склад, расположенный в месте i. Тогда мы получаем задачу, которая называется задачей «простой локализации»:

$$\sum_{i \in I} f_i y_i + \sum_{j \in I} \sum_{j \in J} c_{ij} x_{ij} \to \min,$$

$$x_{ij} \leqslant x_i \qquad \forall i \in I, \quad \forall j \in J, \tag{5'}$$

(SL)
$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J, \tag{6}$$

$$\forall i \in I\colon\ y_i = 0 \text{ или } 1,$$

$$\forall i, j\colon\ 0 \leqq x_{ij} \leqq 1,$$

Задача простой локализации изучалась многими авторами.

Приведенная здесь формулировка приведена в [9]. В [19] предложена высокоэффективная процедура для решения непрерывной линейной задачи (получаемой ослаблением условий целочисленности для переменных y_i) с помощью двойственного метода (который не совпадает с классическим алгоритмом субградиента). Скачки двойственности весьма малы, и во многих случаях решение двойственной задачи приводит к оптимальному целочисленному решению исходной задачи. Если полученные решения не целочисленны, то нужно воспользоваться методом разветвления (типа разделения и оценки, см. гл. 7, § 2), но время вычислений чрезвычайно мало (порядка 1,5 с на IBM 360/91 для задач с $|I| = |J| = 100$).

Применение моделей локализации к автоматической классификации было предложено в [64] и в важном частном случае сводится к задаче простой локализации. Предполагается, что заданы N объектов, и для каждой пары объектов i и j предполагается заданным число $d_{ij} \geqslant 0$, называемое *расстоянием* между i и j (если эти объекты могут рассматриваться как элементы евклидова пространства), или, более общим образом, *индекс несогласованности* (по договоренности полагаем $d_{ii} = 0$).

Задача состоит в разбиении множества этих объектов на p классов (целое число p задано) и в выборе в каждом классе специального объекта, называемого *представителем* этого класса, так, чтобы сумма расстояний от объектов до их представителей была минимальна.

Мы видим, что речь идет о частном случае задачи простой локализации, в которой

$$\{\text{множество клиентов}\} = \{\text{множество мест}\}$$

есть множество из N объектов, причем $c_{ij} = d_{ij}$ для всех i, j и $f_i = 0$ для всех i.

Кроме того, так как каждый объект должен быть привязан к одному и только одному классу, то на переменные x_{ij} должны быть наложены условия целочисленности.

В несколько более общей ситуации мы можем предполагать, что множество $J = \{1, 2, \ldots, N\}$ объектов и множество $I = \{1, 2, \ldots, M\}$ возможных представителей не обязательно совпадают.

Тогда мы получаем следующую постановку задачи классификации:

$$
\text{(CP)}\quad
\begin{aligned}
&\sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} \to \min,\\
&x_{ij} \leqslant y_i \qquad \forall i \in I,\ \ \forall j \in J, \qquad (5')\\
&\sum_{i\in I} x_{ij} = 1 \qquad \forall j \in J, \qquad (6)\\
&\sum_{i\in I} y_i = p, \qquad (7)\\
&y_i = 0 \text{ или } 1 \qquad \forall i \in I,\\
&x_{ij} = 0 \text{ или } 1 \qquad \forall i \in I,\ \ \forall j \in J.
\end{aligned}
$$

Ограничения (5′) выражают тот факт, что объект j может быть связан с представителем i только в том случае, когда представитель i выбран ($y_i = 1$). Ограничения (6) в сочетании с ограничениями целости на переменные x_{ij} воспроизводят тот факт, что каждый объект должен быть связан с одним и только одним представителем. Ограничение (7) определяет число представителей (или число классов).

В [64] было предложено решать задачу (CP), применяя лагранжево ослабление по ограничениям (6). Если связать с каждым ограничением типа $\sum_{i\in I} x_{ij} = 1$ множитель Лагранжа λ_i (без ограничения на знак), то значение $w(\lambda)$ двойственной функции в точке λ может быть получено как оптимальное решение задачи

$$
(\text{RCP}_1)\quad
\begin{aligned}
&w(\lambda) = \min\left(-\sum_{j\in J}\lambda_j + \sum_{j\in J}\sum_{i\in I}(d_{ij} + \lambda_j)x_{ij}\right),\\
&x_{ij} \leqslant y_i \qquad \forall i \in I,\ \ \forall j \in J,\\
&\sum_{i\in I} y_i = p,\\
&y_i = 0 \text{ или } 1 \qquad \forall i \in I,\\
&x_{ij} = 0 \text{ или } 1 \qquad \forall i \in I,\ \ \forall j \in J.
\end{aligned}
$$

Эта задача может быть следующим образом решена для любого множества значений множителей λ_i.

Предположим, что известны те p переменных y_i, которые принимают в оптимальном решении значение, равное единице. Пусть $I_1 \subset I$ ($|I_1| = p$) — подмножество индексов этих переменных. Тогда оптимальное значение задачи (RCP_1) равно

$$
-\sum_{j\in J}\lambda_j + \sum_{i\in I_1}\sum_{j\in J}\min\{d_{ij} + \lambda_j;\ 0\}
$$

и соответствующее оптимальное решение $\bar{x}(\lambda)$ определяется следующим образом:

$$\bar{x}_{ij}(\lambda) = 1), \quad \text{если } i \in I_1 \text{ и } d_{ij} + \lambda_j < 0,$$

$$\bar{x}_{ij}(\lambda) = 0 \quad \text{в противном случае.}$$

Таким образом, для того чтобы определить оптимальное подмножество I_1, достаточно для каждого $i \in I$ вычислить величину $S(i) = \sum_{j \in J} \min\{d_{ij} + \lambda_j; 0\}$. Тогда I_1 оказывается подмножеством индексов, соответствующим p наименьшим значениям величины $S(i)$. В качестве субпродукта при решении задачи (RCP) мы получаем также и субградиент w в точке λ, j-я компонента которого определяется формулой

$$\sum_{i \in I} \bar{x}_{ij}(\lambda) - 1.$$

Решение дуальной задачи с помощью алгоритмов субградиента позволяет тогда получить чрезвычайно близкие к точным оптимальным (целочисленным) значениям оценки этих величин снизу, как показывают вычислительные эксперименты, описываемые в [64]. В огромном большинстве случаев эти скачки не превосходят 1 %, и только в редких исключительных случаях они могут принимать значения порядка 4—7 %.

Кроме того, достаточно часто этим способом получаются точные решения (соответствующие нулевым скачкам двойственности). Тем не менее, как подчеркивают в [64] сами авторы, предыдущий метод ослабления страдает тем недостатком, что в некоторых ситуациях он не приводит к построению хороших целочисленных решений исходной задачи. Таков, например, случай, когда два разных представителя k и l очень близки в том смысле, что соответствующие векторы $(d_{kj})_{j \in J}$ и $(d_{lj})_{j \in J}$ очень близки. В этом случае для любых значений λ_j всегда будет выполняться приближенное равенство $S(k) \approx S(l)$, так что в соответствии с этим процедура будет ошибочно одновременно включать k и l в множество I_1 оптимальных представителей.

Чтобы устранить эту трудность, в [3] предлагается решать двойственную задачу полученной задачи (CP) с помощью лагранжева ослабления ограничений (5′), а не ограничений (6). При таком подходе мы связываем, таким образом, с каждым ограничением типа $x_{ij} - y_i \leqslant 0$ двойственное переменное $\mu_{ij} \geqslant 0$. Для набора заданных значений μ_{ij} двойственная функция пишется тогда в виде

$$w(\mu) = \min\left(-\sum_{i \in I} y_i \sum_{j \in J} \mu_{ij} + \sum_{i \in I} \sum_{j \in J} (d_{ij} + \mu_{ij}) x_{ij}\right),$$

$$(\mathrm{RCP}_2) \quad \begin{aligned} &\sum_{i \in I} x_{ij} = 1, \\ &\sum_{i \in I} y_i = p, \\ &y_i = 0 \text{ или } 1 \qquad \forall i \in I, \\ &x_{ij} = 0 \text{ или } 1 \qquad \forall i \in I, \quad \forall j \in J. \end{aligned}$$

Оптимальное решение $(\bar{y}(\mu), \bar{x}(\mu))$ этой задачи легко получить следующим образом: (а) мы приписываем значение 1 переменным y_i, соответствую-

щим p наименьшим значениям величин $\mu_i = \sum_{j \in J} \mu_{ij}$ (таким образом, число таких переменных y_i равно p); (б) для каждого $j \in J$ имеем

$$\bar{x}_{ij}(\mu) = 0 \quad \text{при} \quad i \neq 0,$$

$$x_{i_0 j}(\mu) = 1,$$

где i_0 определяется формулой

$$d_{i_0 j} + \mu_{i_0 j} = \min_{i \in I} \{d_{ij} + \mu_{ij}\}.$$

Кроме того, вектор, (i, j)-я составляющая которого равна

$$\bar{x}_{ij}(\mu) - \bar{y}_i(\mu),$$

есть субградиент дуальной функции w в точке μ. В отличие от предыдущего, этот метод даже в случае существования многих очень близко расположенных объектов позволяет получить хорошие целочисленные решения исходной задачи в ходе процедуры оптимизации двойственной функции (на каждом шаге итерации нужно выбирать в качестве представителей те элементы, которые соответствуют p переменным $\bar{y}_i(\mu)$, равным 1, и определять решение исходной задачи — разбиение объектов,— привязывая каждый объект к ближайшему представителю в смысле исходных расстояний d_{ij}). Полученные с помощью задачи (RCP_2) миноранты очевидным образом сравнимы по качеству с минорантами, получаемыми с помощью задачи (RCP_1), поскольку в обоих случаях оптимальные значения дуальной задачи равны «непрерывному» оптимальному значению задачи (CP). С другой стороны, хотя задача (RCP_2) заставляет путешествовать по сопряженному пространству увеличенной размерности ($N \times M$ вместо N), но свойства сходимости субградиентных алгоритмов типа «сходящегося ряда» или «ослабления» (см. гл. 4, п. 3.5 и 3.7) оказываются относительно мало зависящими от размерности пространства (см., например, [62]), и число итераций, необходимых для получения хорошей аппроксимации оптимума двойственной задачи, практически в обоих случаях одинаково. Важно, однако, что (RCP_2) приводит к лучшим решениям исходной задачи, чем (RCP_1), и приводит, таким образом, как правило, к очень хорошим интервальным оценкам (порядка 2—3 %) точного целочисленного оптимума. Это позволяет решать на практике задачи относительно существенного объема (порядка 100—200 объектов), полностью избегая обращения к методам разветвленного исследования.

В заключение заметим, что та же самая техника лагранжева ослабления, которая здесь была описана в частном случае задач классификации, может быть применена в задаче простой локализации и составляет другой возможный подход к этой задаче, кроме уже упоминавшегося выше алгоритма Эрленкоттера.

§ 3. Задача о дереве Штейнера в графах

Пусть $G = [X, U]$ — связный неориентированный граф порядка N ($|X| = N$), в котором каждое ребро $u \in U$ снабжено длиной $l_u > 0$. Множество вершин X разбивается на два подмножества:

X_e (множество «обязательных» вершин);

X_s (множество «вспомогательных» вершин, или *точек Штейнера*).

Задача состоит в определении связного подграфа $T = [X', U']$ графа G, множество вершин которого содержит все «обязательные» вершины ($X_c \subseteq X' \subseteq X = X_c \cup X_*$), и имеющего наименьшую возможную полную длину $l(T) = \sum_{u \in U'} l_u$.

Эта задача, которая, как и задача о коммивояжере и задачи локализации, принадлежит классу NP-полных задач, рассматривается в теории графов как трудная задача.

Мы сейчас опишем в общих чертах принцип подхода к этой задаче с точки зрения двойственности и лагранжева ослабления, изученный в [36, 37]. Используем с этой целью постановку задачи в терминах неодновременных потоков на графах, предложенную в [12] (см. также [51]).

Рассмотрим сначала произвольное дерево Штейнера $T = [X', U']$ в графе G.

Выберем произвольную вершину $s \in X_c$ в качестве корня (тогда необходимо $s \in X'$, поскольку X' должно содержать все обязательные вершины), а затем шаг за шагом присвоим всем ребрам нашего дерева ориентацию, пробегая это дерево, исходя из корня s. Свяжем с каждой полученной таким образом дугой емкость 1. Заметим тогда, что полученная таким образом транспортная сеть обладает следующим свойством: она позволяет протечь независимо один от другого $|X_c| - 1$ потокам со значением 1 между корнем s и каждой из других вершин графа X_c. Это замечание позволяет следующим образом переформулировать задачу Штейнера.

Рассмотрим ориентированный граф $\bar{G} = [X, V]$, получаемый из G заменой каждого ребра $u = (i, j)$ двумя дугами $u^+ = (i, j)$ и $u^- = (j, i)$ с длинами $l_{u^+} = l_{u^-} = l_u$. Если U^+ (соответственно U^-) обозначает множество дуг вида u^+ (соответственно u^-), где u пробегает семейство U, то $V = U^+ \cup U^-$. Обозначим через N и M число вершин и число дуг графа $\bar{G}$. Задача сводится тогда к определению емкостей y_v (равных 0 или 1), которые должны быть так приписаны каждой дуге $v \in V$ графа $\bar{G}$, чтобы полученная транспортная сеть допускала протекание потока со значением 1 между корнем s и произвольной вершиной $i \in I = X_c - \{s\}$, причем полная длина $\sum_{v \in V} l_v y_v$ должна быть при этом минимальной.

Обозначим через A матрицу инцидентности между вершинами и дугами графа $\bar{G}$ и для любого $i \in I = X_c - \{s\}$ обозначим через b^i вектор размерности N, все компоненты которого равны нулю, кроме компоненты s (равной (-1)) и компоненты i (равной $(+1)$). Для всех $i \in I$ определим M-вектор $\varphi^i \geqslant 0$, представляющий поток в нашем графе со значением 1 между s и i. Тогда задачу о дереве Штейнера можно поставить следующим образом:

$$
\begin{aligned}
& ly = \sum_{v \in V} l_v y_v \to \min, \\
& \forall i \in I = X_c - \{s\}: \\
& A\varphi^i = b^i; \qquad (8) \\
& \varphi^i \leqslant y, \qquad (9) \\
& y = (y_v)_{v \in V} \in \{0, 1\}^M, \\
& \varphi^i \geqslant 0 \qquad \forall i \in I.
\end{aligned}
\qquad \text{(SP)}
$$

Мы видели выше, что любое дерево Штейнера есть решение задачи (SP). В обратную сторону можно установить следующее

Предложение 1. *Пусть y^* — оптимальное (целочисленное) решение задачи* (SP). *Тогда множество дуг $\widetilde{V}(y^*) = \{v \mid v \in V;\ y_v^* = 1\}$ индуцирует на G частичный граф $\widetilde{G}(y^*)$, который является деревом Штейнера.*

Доказательство. Вследствие ограничений (8) и (9) в множестве $\widetilde{G}(y^*)$ существует по крайней мере один путь между любыми двумя вершинами из X_c.

С другой стороны, если $\widetilde{G}(y^*)$ содержит цикл, то, удаляя из этого цикла некоторое ребро, мы получили бы (частичный) подграф графа G, связывающий все вершины семейства X_c и имеющий строго меньшую стоимость, чем стоимость решения y^*. Отсюда получаем противоречие, которое завершает доказательство.

К задаче (SP) можно применить подход, связанный с лагранжевым ослаблением, связывая с каждым ограничением типа (9) вектор размерности M, образованный множителями Лагранжа (двойственными переменными) θ^i, где $\theta^i \geqslant 0$ для всех $i \in I = X_c - \{s\}$. Для заданного вектора $\theta = (\theta^i)_{i \in I}$ функция Лагранжа нашей задачи записывается в виде

$$L(\varphi, y, \theta) = ly + \sum_{i \in I} \theta^i (\varphi^i - y) = \left(l - \sum_{i \in I} \theta^i\right) y + \sum_{i \in I} \theta^i \varphi^i \qquad (10)$$

и значение дуальной функции $w(\theta)$ представляется в виде

$$w(\theta) = \min_{y \in \{0,1\}^M} \left\{\left(l - \sum_{i \in I} \theta^i\right) y\right\} + \sum_{i \in I} \min_{\substack{\varphi^i \geqslant 0 \\ A\varphi^i = b^i}} \{\theta^i \varphi^i\}. \qquad (11)$$

В первом слагаемом в правой части минимум $\bar{y}(\theta)$ достигается для каждой дуги $v \in V$, если мы положим

$$\bar{y}_v = 1, \quad \text{если} \quad l_v - \sum_{i \in I} \theta_v^i < 0,$$

$$\bar{y}_v = 0, \quad \text{если} \quad l_v - \sum_{i \in I} \theta_v^i \geqslant 0.$$

Второй член в правой части может быть определен для любого $i \in I$ с помощью решения задачи типа

$$(\mathrm{SHP}_i) \qquad \begin{aligned} &\theta^i \varphi^i \to \min, \\ &A\varphi^i = b^i, \\ &\varphi^i \geqslant 0, \end{aligned}$$

которая есть не что иное, как задача о кратчайшем пути между вершинами s и i в графе $\bar{G}$, если дугам $v \in V$ приписать длины θ_v^i.

Обозначим через $\bar{\varphi}^i(\theta)$ оптимальное решение задачи (SHP_i).

В качестве субпродукта при вычислении $w(\theta)$ мы получаем также и субградиент w в точке θ, определяемый равенством

$$g(\theta) = \begin{bmatrix} \bar{\varphi}^1(\theta) - \bar{y}(\theta) \\ \bar{\varphi}^2(\theta) - \bar{y}(\theta) \\ \cdots\cdots \\ \bar{\varphi}^k(\theta) - \bar{y}(\theta) \end{bmatrix}$$

(где предполагается, что вершины набора $I = X_c - \{s\}$ занумерованы числами 1, 2, ..., k).

Двойственная задача

$$(\mathrm{DSP}) = \begin{cases} w(0) \to \max, \\ 0 \geqq 0 \end{cases}$$

тогда может быть, таким образом, решена с помощью некоторого алгоритма субградиента.

Вычислительные эксперименты, осуществленные в некотором количестве пробных задач (содержащих до 20 вершин), показывают, что этот подход систематически приводит к минорантам очень высокого качества (менее 5 %) и очень хорошим приближенным решениям (см. [36]). Во многих случаях дуальный алгоритм завершается явным определением седловой точки и, следовательно, оптимального дерева Штейнера.

Здесь важно отметить, что эта программа была использована также и для задач, имеющих ненулевой скачок двойственности, но не допускающих применения никаких сколько-нибудь ясных эффективных вычислительных подходов. Так как задача Штейнера является *NP*-полной (см. [26]), то гипотеза, согласно которой задача (SP) никогда не допускает скачка двойственности, выглядит чрезвычайно малоправдоподобной. Тем не менее трудность выявления ее скачков двойственности позволяет предположить, что эта задача, вероятно, обладает важными специальными свойствами. Среди существенных еще не решенных вопросов можно отметить следующие. Частный случай задачи о дереве Штейнера, в которой все вершины обязательны ($X_c = X$), есть не что иное, как классическая задача о дереве наименьшей длины [44], которая может быть решена за полиномиальное время ($\sigma(N)^2$ для полных графов) с помощью алгоритма, аналогичного приведенному в [66]. В таких задачах разумно предполагать, что связанные с ними линейные задачи (SP) всегда имеют целочисленные оптимальные решения (иначе говоря, скачки двойственности равны нулю). Однако теперь, когда мы вводим вспомогательные точки (точки Штейнера), то соответствующие ограничения типа (8) и (9) пропадают, но сама задача (SP) сохраняет в точности ту же структуру. Как тогда выразить, что именно может привести к скачку двойственности? Можно ли в таких случаях охарактеризовать неравенства, определяющие выпуклую оболочку целых решений задачи (SP), с помощью подхода типа «полиэдральной комбинаторики» (см. п. 3.6)?

Замечание. Существует другой возможный подход для (непрерывного) решения задач линейного программирования, подобных задаче (SP): речь идет о методе разложения Бендерса (см. гл. 8, § 4), который представляет собою не что иное, как метод [31] для оптимального синтеза сетей, пробегаемых не одновременными потоками (частный случай этой задачи исследован в гл. 8, п. 5.3).

В [2] предложен алгоритм, использующий принцип порождения ограничений (обобщенное линейное программирование, см. гл. 8, § 1) для получения миоранта для задачи Штейнера. Хотя автор статьи [2] и не заметил сам этого сходства, но предлагаемый им подход есть не что иное, как метод Гомори и Ху [31], примененный к задаче (SP), в которой ослаблены условия целочисленности.

§ 4. Задачи разбиения и слияния гиперграфов («set packing», «упаковка» и «set partitioning», разбиение)

Эти задачи возникают в огромном количестве приложений, среди которых следует отметить задачи диагностики или отыскания повреждений, задачи локализации или автоматической классификации [60], задачи о чередовании экипажей в авиационных компаниях [1] или составления расписаний в автобусных линиях [43], некоторые задачи распределения поездок и т. д. (За обзором множества таких приложений мы можем отослать читателя, например, к [32], гл. 10.) Типичная задача разбиения ставится следующим образом. Рассмотрим конечное множество $E = \{1, 2, \ldots, m\}$ и семейство подмножеств множества E:

$$\mathscr{S} = \{S_1, S_2, \ldots, S_n\}.$$

Свяжем с каждым подмножеством S_j семейства $\mathscr{S}$ некоторую цену c_j. Задача состоит в том, чтобы выбрать некоторое количество подмножеств, принадлежащих семейству $\mathscr{S}$, таким образом, чтобы они образовывали разбиение множества E и чтобы стоимость этого разбиения была минимальной (минимум берется по множеству всех возможных разбиений).

Обозначим через $A = (a_{ij})$ $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ матрицу инцидентности семейства $\mathscr{S}$ в E, т. е. матрицу с элементами 0 и 1, определяемую соотношением

$$a_{ij} = 1, \quad \text{если} \quad i \in S_j,$$
$$a_{ij} = 0, \quad \text{если} \quad i \notin S_j.$$

Рассмотрим, кроме того, множество двузначных переменных x_j $(j = 1, 2, \ldots, n)$ таких, что $x_j = 1$ тогда и только тогда, когда подмножество S_j выбрано как входящее в данное разбиение.

Тогда задача разбиения может быть следующим образом сформулирована как задача целочисленного линейного программирования:

$$cx = \sum_{j=1}^{n} c_j x_j \to \min,$$
$$\text{(PP)} \qquad Ax = 1,$$
$$x_j = 0 \quad \text{или} \quad 1 \quad \forall j = 1, 2, \ldots, n.$$

Классический подход к этой задаче состоит в применении к ней метода разветвленного поиска с использованием в качестве функции оценки оптимального значения непрерывной задачи (получаемой при замене ограничений целости: $x_j = 0$ или 1 — условием $0 \leqslant x_j \leqslant 1$). Однако, как отмечают многие авторы (например [53]), непрерывные линейные программы, получаемые на каждом шаге, часто оказываются трудно разрешимыми в связи с проблемами вырожденности (многие базовые переменные равны нулю).

Поэтому в [65] был предложен метод лагранжева ослабления, основанный на формулировании задачи (PP) как задачи о совершенном соединении с минимальной стоимостью в графе (MP) с дополнительными ограничениями. Напомним здесь, что задача о совершенном соединении сама по себе есть частный случай задачи разбиения (PP), в которой матрица A содержит в точности

два элемента в каждом столбце, равных единице, и в этом случае представляет собой не что иное, как матрицу инцидентности между вершинами и ребрами для рассматриваемого графа. Использование задач соединения в графах для построения ослаблений задачи разбиения представляет двойной интерес.

а) Во-первых, задачи соединения составляют семейство хорошо решаемых задач целочисленного линейного программирования — в том смысле, что для исследования этих задач существуют эффективные (т. е. полиномиальные) алгоритмы. По существу, *принцип этих алгоритмов восходит к* [16], *но наиболее* эффективные из известных улучшений были описаны затем в [47] и [14] (сложность $O(N^3)$) и в [25] (сложность $O(MN \ln N)$ для графов малой плотности). В [57] был описан также алгоритм, основанный на других принципах, но приводящий к сравнимым с указанными выше границами сложности (см., например, [32], гл. 7, упражнение 17). Применение этих новых алгоритмов позволяет также избежать наибольшего препятствия в применении линейного программирования, а именно проблем вырожденности, о которых мы уже упоминали выше.

б) Во-вторых, тот факт, что решение ослабления исходной задачи явно учитывает ограничения целости на переменные задачи, позволяет получать миноранты более высокого качества, чем те, к которым приводит классический подход, состоящий в решении соответствующей непрерывной линейной задачи (*или ее двойственной задачи*).

Ослабление, предложенное в [65], основано на следующем принципе. Предположим сначала, что матрица $A = (a_{ij})$ $(i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n)$ задачи разбиения (РР) устроена так, что в каждом столбце число членов, равных 1, либо равно в точности 1 (тогда соответствующее переменное может рассматриваться как переменное скачка), либо четно, так что

$$\sum_{i=1}^{m} a_{ij} = 2K_j,$$

где K_j — целое.

Это предположение не очень ограничительно вследствие следующего замечания.

Если для столбца j выполняется равенство $\sum_{i=1}^{m} a_{ij} = 2K_j - 1$ (где K_j — целое $\geqslant 2$), то достаточно добавить дополнительное переменное, обозначаемое, например, через s_j (переменное скачка), и дополнительное ограничение вида $x_j + s_j = 1$ $(x_j \geqslant 0,\ s_j \geqslant 0)$, и это последнее ограничение, как легко заметить, *тривиальным образом выполняется для любого решения задачи* (РР), так что мы, таким образом, получаем задачу, равносильную исходной.

Осуществляя это преобразование для всех столбцов j, для которых это необходимо, мы можем, таким образом, всегда вести себя так, как будто рассматриваемая задача разбиения обладает требуемым свойством. При этих условиях мы, исходя из матрицы A, построим матрицу M (размера $m \times K$), связывая с каждым столбцом A_j матрицы A, удовлетворяющим условию $\sum_{i=1}^{m} a_{ij} = 2K_j$ $(K_j \geqslant 2)$, набор из K_j столбцов матрицы M, обозначаемых $M_j^1, M_j^2, \ldots, M_j^{K_j}$, каждый из которых содержит в точности два числа,

равных 1, причем должно выполняться условие

$$\sum_{k=1}^{K_j} M_j^k = A_j.$$

С каждым столбцом M_j^k мы связываем целочисленное переменное y_j^k со значениями 0 и 1 со стоимостью

$$\gamma_j^k = \frac{c_j}{K_j}.$$

Если потребовать от переменных y_j^k, чтобы они удовлетворяли всем дополнительным ограничениям типа

$$y_j^k = y_j^{k+1} = 0, \qquad k = 1, 2, \ldots, K_j - 1, \tag{12}$$

то мы получим новую формулировку задачи разбиения (PP) как задачи соединения с дополнительными ограничениями:

$$\text{(CM)} \qquad \begin{aligned} &\gamma y \to \min, \\ &My = 1, \\ &Sy = 0, \\ &y \in \{0, 1\}^\kappa, \end{aligned} \tag{12'}$$

где $Sy = 0$ есть матричная форма ограничений (12) и где через γ обозначен вектор с компонентами $\left(\gamma_j^k\right)$ и через y — вектор с компонентами $\left(y_j^k\right)$. Матрица M может на самом деле рассматриваться как матрица инцидентности между вершинами и ребрами некоторого графа G, и, ослабляя ограничения $Sy = 0$ в задаче (CM), мы получаем задачу о совершенном соединении с минимальной стоимостью γ на этом графе. Тогда метод решения состоит в исследовании задачи (12′) с помощью лагранжева ослабления.

Связывая с этими ограничениями вектор множителей Лагранжа λ (без ограничений на знак), мы построим двойственную функцию как оптимальное значение — при данном фиксированном значении λ — задачи о соединении на графе G:

$$\begin{aligned} w(\lambda) &= \min(\gamma + \lambda S) y, \\ My &= 1, \\ y &\in \{0, 1\}^\kappa. \end{aligned}$$

Если, с другой стороны, $\bar{y}(\lambda)$ обозначает оптимальное решение этой задачи, то вектор $S\bar{y}(\lambda)$ есть субградиент функции w в точке λ. Отыскание максимума функции (решение двойственной задачи) может быть, таким образом, осуществлено с помощью некоторого алгоритма субградиента — типа алгоритмов, описанных в гл. 4, § 3. В [65] предлагалось также применить метод циклических координат в пространстве переменных λ (см. гл. 4, п. 4.1), но их вычислительные результаты показывают, что алгоритмы субградиента приводят, вообще говоря, к лучшим значениям дуальной функции и обычно требуют меньшего времени вычислений. С другой стороны, хотя непрерывное линейное программирование и кажется конкурентноспособным для задач разбиения небольшого размера (для определенности — до 100 ограничений и до 200 пере-

менных), то тем не менее кажется, что для задач большого размера лагранжево ослабление может проявить себя лучше, как с точки зрения времени вычислений, так и с точки зрения качества полученных границ.

§ 5. Задачи о кратчайшем пути с дополнительным(и) ограничением(ями) и связные комбинаторные задачи

Сначала мы рассмотрим задачу о кратчайшем пути с единственным дополнительным ограничением типа «задачи о рюкзаке». Затем мы рассмотрим некоторый более общий класс задач о прокладывании пути без ограничений. Рассмотрим некоторый граф $G = [X, U]$, где X есть множество вершин ($|X| = N$), а U — множество дуг ($|U| = M$), и пусть $S \in X$, $t \in X$ — некоторые вершины этого графа.

Свяжем с каждой дугой $u \in U$ два числа l_u и α_u (например, число l_u можно интерпретировать как длину дуги u, а α_u — как время пробегания дуги u). Тогда любой путь π между вершинами s и t в графе G может быть определен с помощью своего характеристического вектора x, т. е. такого M-вектора с компонентами 0 и 1, что $x_u = 1 \Leftrightarrow u \in \pi$.

Обозначим через X_{st} множество характеристических векторов всех элементарных путей в G между s и t. Пусть $x \in X_{st}$ и $lx = \sum_{u \in U} l_u x_u$ — длина пути с характеристическим вектором x (сумма длин дуг пути), а $\alpha x = \sum_{u \in U} \alpha_u x_u$ — время пробега по этому пути (сумма времен пробега по дугам пути). Задача о кратчайшем пути с дополнительным ограничением может быть представлена в следующей общей форме:

$$\text{(CSP)} \qquad \begin{gathered} lx \to \min, \\ \underline{\beta} \leqq \alpha x \leqq \overline{\beta}, \\ x \in X_{st}. \end{gathered} \tag{13}$$

Заметим, что эта формулировка является достаточно общей, чтобы включить два важных на практике частных случая:

1) случай ограничения с неравенством типа «рюкзак», который соответствует в рассматриваемой ситуации условию, что все компоненты вектора α неотрицательны и $\underline{\beta} = 0$. Этот вариант будет обозначаться через (CSP_1);

2) случай ограничения с равенством типа «рюкзак» с $\alpha \leqslant 0$ и $\underline{\beta} = \overline{\beta}$. Этот вариант будет обозначаться через (CSP_2).

Все эти задачи априори являются трудными комбинаторными задачами, поскольку они включают задачу о рюкзаке как частный случай (если взять в качестве X_{st} множество $\{0, 1\}^M$ всех возможных M-векторов с компонентами 0 и 1). Чаще всего в приложениях встречается вариант (CSP_1). В [55] можно найти весьма подробное исследование этих приложений, включающее следующие задачи: кратчайшего пути с ограничениями ослабления для задач доставки в телефонных сетях; контроль за ходом выполнения промышленного заказа при бюджетных ограничениях; оптимальное расширение сетей коммуникации (минимизация необходимых вложений при выполнении заданных

требований); путь минимальной средней длины и с ограничениями на изменения в вероятностных графах; отыскание критического пути с минимальным средним временем пробега и с ограничениями на изменения в вероятностных задачах управления производством (где задачи имеют случайное время жизни).

Применение технических приемов двойственности при решении комбинаторных задач вида (CSP_1), по-видимому, впервые было применено в [55]. Напомним здесь главное.

Свяжем с ограничением $\alpha x = \overline{\beta}$ множитель Лагранжа $\lambda \geqslant 0$ и определим двойственную функцию $w(\lambda)$ как оптимальное значение «обычной» задачи о кратчайшем пути (без ограничений):

$$w(\lambda) = \min_{x \in X_{st}} \{(l + \lambda\alpha)\, x\} - \lambda\overline{\beta}, \tag{14}$$

в которой дуге u графа сопоставлена длина $l_u + \lambda\alpha_u$.

Чтобы предыдущая задача всегда имела смысл, мы предположим, что ни для значений l_u, ни для значений α_u на дугах нашего графа не существует никаких циклов с отрицательной полной длиной (в частности, это условие выполняется, если $l \geqslant 0$ и $\alpha \geqslant 0$).

Обозначим в дальнейшем через $\bar{x}(\lambda)$, где $\lambda \geqslant 0$ задано, путь в X_{st}, минимизирующий функцию Лагранжа $(l + \lambda\alpha)x - \lambda\overline{\beta}$, и положим для простоты $\overline{l}(\lambda) = l\bar{x}(\lambda)$ и $\overline{\alpha}(\lambda) = \alpha x(\lambda)$.

Тогда двойственная функция $w(\lambda)$ оказывается в этом случае функцией одного переменного, и ей можно дать следующую интересную графическую интерпретацию.

Действительно, для любого $\lambda \geqslant 0$ величина $w(\lambda)$ есть минимум значений, принимаемых всевозможными аффинными функциями вида

$$z_x(\lambda) = lx + \lambda[\alpha x - \overline{\beta}]$$

для x, пробегающего (конечное) множество X_{st} всех элементарных путей между вершинами s и t графа G. Следовательно, график функции w есть нижняя *оболочка конечного семейства прямых, где каждая прямая семейства соответствует* некоторому определенному элементарному пути в графе между вершинами s и t.

С другой стороны, легко показать, что значения $\overline{l}(\lambda)$ и $\overline{\alpha}(\lambda)$ с ростом λ соответственно монотонно не убывают и монотонно не возрастают.

Для иллюстрации рассмотрим пример графа на рис. 3, где рядом с каждой дугой u указаны значения l_u и α_u соответственно и где мы выбрали $s = 1$, $t = 4$ и $\overline{\beta} = 10$.

В этой задаче есть четыре элементарных пути между началом $s = 1$ и концом $t = 4$:

путь	$x^1 = \{1, 2, 4\}$	$lx^1 = 13,$	$\alpha x^1 = 14,$
путь	$x^2 = \{1, 3, 4\},$	$lx^2 = 17,$	$\alpha x^2 = 9,$
путь	$x^3 = \{1, 2, 3, 4\},$	$lx^3 = 22,$	$\alpha x^3 = 18,$
путь	$x^4 = \{1, 3, 2, 4\},$	$lx^4 = 20,$	$\alpha x^4 = 7.$

Для значения $\bar{\beta} = 10$ мы проводим на рис. 4 четыре прямых, соответствующих уравнениям

$$z_{x1}(\lambda) = 13 + 14\lambda - 10\lambda, \quad \text{(D1)}$$

$$z_{x2}(\lambda) = 17 + 9\lambda - 10\lambda, \quad \text{(D2)}$$

$$z_{x3}(\lambda) = 22 + 18\lambda - 10\lambda, \quad \text{(D3)}$$

$$z_{x4}(\lambda) = 20 + 7\lambda - 10\lambda. \quad \text{(D4)}$$

Функция $w(\lambda)$, являющаяся нижней оболочкой этих четырех прямых, вогнута, кусочно линейна и имеет две точки излома: для $\lambda_1 = 0{,}8$ (пересечение прямых (D1) и (D2)) и для $\lambda_2 = 1{,}5$ (пересечение прямых (D2) и (D4)). Оптимум дуальной функции, равный $w(\lambda_1) = 16{,}2$, есть также мипоранта длины оптимального пути при наличии ограничения $lx^2 = 17$).

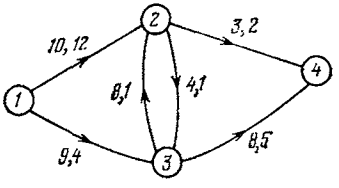


Рис. 3. Пример, иллюстрирующий задачу о кратчайшем пути с ограничениями. На каждой дуге u указаны соответствующие значения l_u и α_u

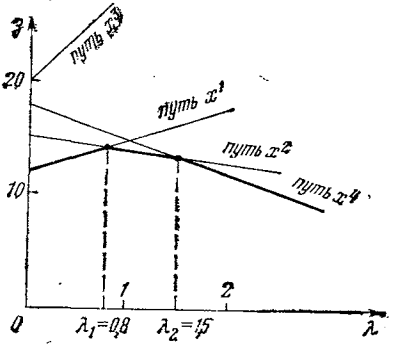


Рис. 4. Двойственная функция для задачи о кратчайшем пути с ограничениями, соответствующая примеру на рис. 3

Вернемся к общей задаче. Как мы видим, решение двойственной задачи состоит в отыскании максимума вогнутой кусочно линейной функции (с конечным числом точек излома) от одного переменного λ. К этой задаче можно применить один из многочисленных методов одномерной оптимизации, описанных в главе 3 (дихотомический метод, метод Фибоначчи и т. д.). В применении этих методов неудобно то, что они, вообще говоря, не сходятся за конечное число итераций. Но можно заметить, что в этой задаче можно найти точное решение за конечное число шагов с помощью следующего метода последовательных приближений [55].

Процедура 1 (решение двойственной задачи последовательными приближениями).

а) Вычислить $w(\lambda)$ для $\lambda_1 = 0$, затем для $\lambda_2 = +\infty$ (получаем два оптимальных пути: $\bar{x}(\lambda_1)$ и $\bar{x}(\lambda_2)$). Заметим, что $\bar{x}(\lambda_2)$ есть не что иное, как кратчайший путь между s и t относительно значений α_u на дугах графа.

б) Если $\bar{\alpha}(\lambda_2) \geqslant \bar{\beta}$ КОНЕЦ: задача не имеет решения. Если $\bar{\alpha}(0) \leqslant \bar{\beta}$ КОНЕЦ: путь $\bar{x}(0)$ с длиной $\bar{l}(0)$ есть оптимальное решение задачи (в этом случае дополнительное ограничение не действует). В остальных случаях:

в) Пусть λ' — абсцисса точки пересечения двух прямых, заданных уравнениями

$$z = \overline{l}(\lambda_1) + \lambda(\overline{\alpha}(\lambda_1) - \overline{\beta}),$$

$$z = \overline{l}(\lambda_2) + \lambda(\overline{\alpha}(\lambda_2) - \overline{\beta}),$$

иначе говоря,

$$\lambda' = \frac{\overline{l}(\lambda_2) - \overline{l}(\lambda_1)}{\overline{\alpha}(\gamma_1) - \overline{\alpha}(\lambda_2)} \quad (\geqq 0).$$

г) Вычислить $w(\lambda')$. Пусть $\bar{x}(\lambda')$ — полученный оптимальный путь.

Если $w(\lambda') = \overline{l}(\lambda_1) + \lambda'(\overline{\alpha}(\lambda_1) - \overline{\beta}) = \overline{l}(\lambda_2) + \lambda'(\overline{\alpha}(\lambda_2 - \overline{\beta}))$, КОНЕЦ: $\lambda^* = \lambda'$ и путь $\bar{x}(\lambda_2)$ есть допустимое (оптимальное, если $\overline{\alpha}(\lambda_2) = \overline{\beta}$, и, вообще говоря, субоптимальное в остальных случаях) решение рассматриваемой задачи. Два значения $w(\lambda^*)$ и $l(\lambda_2)$ дают интервал для оптимума.

В противном случае

д) Если $\overline{\alpha}(\lambda') > \overline{\beta}$, положить $\lambda_1 \leftarrow \lambda'$ и вернуться к в).

Если $\overline{\alpha}(\lambda') \leqslant \overline{\beta}$, положить $\lambda_2 \leftarrow \lambda'$ и вернуться к в).

Очевидно, как было предложено уже в [55], можно использовать оптимальное значение $w(\lambda^*)$ двойственной функции как функцию оценки в процедуре разветвленного поиска типа «ветвей и границ». Вычислительные результаты, полученные затем в [38], подтверждают полезность этого подхода для задач типа (CSP_1). Конечная сходимость «Процедуры 1» вытекает из того, что функция $w(\lambda)$ есть нижняя оболочка конечного числа аффинных функций (каждая из которых соответствует некоторому элементарному пути в графе), и того факта, что до тех пор, пока оптимум λ^* еще не достигнут, все последовательно порождаемые элементарные пути различны.

Процедура 1 на практике проявила себя как исключительно эффективная, и необходимое для нее число оценок двойственной функции оказывается всегда очень небольшим, даже для графов большого размера (обычно от 5 до 10 для графов, содержащих тысячи дуг, как показывают вычислительные эксперименты, результаты которых приводятся, например, в [68]). Аналогичные наблюдения могут быть сделаны и по материалам других авторов, которые, в дальнейшем (и независимо) применяли этот метод к решению других классических комбинаторных задач, включающих дополнительное ограничение (например, задача о рюкзаке с квадратичной целью, см. [40]). Заметим, что также именно по причинам своей эффективности «процедура 1» была выбрана в [68] как составная часть базы (для быстрого получения оценок снизу) в алгоритме типа «допустимого поиска» (называемого также алгоритмом A^*, см. гл. 9, п. 3.3) для решения задач типа (CSP_2) (кратчайший путь с ограничением типа равенства) или типа (CSP) с малыми значениями разности $\overline{\beta} - \underline{\beta}$. Качество результатов, получаемых в этих задачах,— гораздо более трудных (из-за гораздо более жестких ограничений), чем задача (CSP_1),— ясно показывает, насколько в задачах комбинаторной оптимизации полезно применение методов, основанных на теории двойственности и лагранжевом ослаблении.

Другой, классический, подход к задачам о кратчайшем пути с ограничениями (предложенный Гомори еще в 1965 году для решения асимптотической

задачи целочисленного программирования, см. гл. 7, п. 4.4) состоит в последовательном определении 1-го, 2-го, ..., k-го кратчайшего пути относительно длин l_u: первый из полученных путей, удовлетворяющий дополнительному (или дополнительным) ограничению (ограничениям), есть оптимальное решение поставленной задачи (для определения k кратчайших элементарных путей применимы различные алгоритмы, например, предложенные в [75] или [71]). Этот подход, примененный без каких-либо предосторожностей, может привести к перечислению огромного числа путей и поэтому проявляет себя на практике, вообще говоря, как малоэффективный. Тем не менее в [38] показано, что, решая задачу о k кратчайших путях относительно пересчитанных длин $l_u + \lambda^* \alpha_u$ (где λ^* — оптимальное значение двойственного переменного, связанного с дополнительным ограничением) вместо исходных длин l_u, можно, как правило, заметно уменьшить число подлежащих исследованию путей (обычно порядка дюжины вместо нескольких сотен). В алгоритме допустимого поиска, описанном в [68], порядок, в котором рассматриваются переменные, также выбирается в соответствии с критерием возрастания приведенных длин, и это позволяет весьма чувствительно ограничить количество разветвлений, которые должны быть порождены при выполнении такого алгоритма. Эти результаты представляют собою еще и пример того, как использование в ходе классического алгоритма некоторой информации относительно двойственной задачи позволяет заметно улучшить его эффективность.

Подход с точки зрения теории двойственности и лагранжева ослабления, который мы описали здесь для задач о кратчайших путях при наличии одного дополнительного ограничения, может быть без труда обобщен, как это предложено в [55], на случай произвольного количества $p > 1$ ограничений в форме неравенств типа

$$\alpha^i x \leqq \beta^i, \qquad i = 1, 2, \ldots, p.$$

Предложим, что относительно длин l_u, как и относительно длин α_u^i $(i = 1, 2, \ldots, p)$, не существует цикла с отрицательной длиной. Тогда, связывая с каждым из этих p ограничений соответствующий множитель Лагранжа (что дает набор двойственных переменных $\lambda_1, \lambda_2, \ldots, \lambda_p \geqslant 0$), мы определим вогнутую и не всюду дифференцируемую двойственную функцию. Решение двойственной задачи (либо с помощью обобщенного линейного программирования, либо с помощью субградиентной техники) приводит в случае задач с небольшим числом ограничений к хорошим приближенным решениям исходной задачи, и во всех случаях — к минорантам, которые могут быть использованы в процедурах типа «ветвей и границ».

§ 6. Общая задача пересечения двух семейств комбинаторных объектов и ее решение с помощью лагранжева ослабления

В этом разделе мы введем некоторый класс комбинаторных задач, обобщающий задачи о кратчайших путях с ограничениями, которые рассматривались в предыдущем параграфе. Мы изучим также решение этих задач с помощью лагранжева ослабления.

Пусть $E = \{1, 2, \ldots, n\}$ — конечное множество с n элеме
жество $S \subset E$ будет в дальнейшем отождествляться со своим
ческим вектором $c \in \{0, 1\}^n$, определяемым по правилу

$$x_i = 1, \quad \text{если } i \in S,$$

$$x_i = 0 \quad \text{в противном случае.}$$

Пусть $X \subset \{0, 1\}^n$ и $Y \subset \{0, 1\}^n$ — два семейства комбинатс

Задача	Природа элементов множества X	Природа элементов множества Y
Пересечение двух матроидов	Матроид над E	Матроид над E
Матроид с условием четности	Матроид над E	Семейство всех подмножеств, удовлетворяющих условию четности
Коммивояжер с ориентацией и без ориентации	Множество 1-деревьев в графе	Множество назначений (пересечение двух матроидов разбиения)
Минимальное дерево с ограничениями типа «рюкзака»	Множество деревьев в графе	Множество решений задачи о рюкзаке
Кратчайший путь с ограничением типа «рюкзак»	Множество элементарных путей между двумя данными вершинами графа	Множество решений задачи о рюкзаке
Двумерный рюкзак	Множество решений задачи о рюкзаке	Множество решений задачи о рюкзаке
Матроид с дополнительными ограничениями типа рюкзака	Матроид над E	Множество решений задачи о рюкзаке
Коммивояжер с ограничениями расписания поездок (гамильтонов путь)	Множество элементарных путей, удовлетворяющих ограничениям на расписание между двумя вершинами	Семейство подмножеств множества дуг мощности $N - 1$, где N есть число вершин в данном графе.

*) Прочерк означает, что задача, по-видимому, к настоящему вре

Сопоставим каждому элементу $i \in E$ его стоимость c_i. Тогда в определении такого элемента x в пересечении X и Y, которь мальную стоимость; иначе говоря,

$$\text{(IP)} \qquad cx = \sum_{i \in E} c_i x_i \to \min,$$

$$x \in X \cap Y$$

(задача о пересечении).

Эта общая модель включает как частный случай задачу о кратчайшем пути с ограничением типа «рюкзака», изученную в § 5 (нужно взять в качестве E множество дуг графа, в качестве X — множество путей между s и t, а в качестве Y — множество решений неравенства (11): $\underline{\beta} \leqslant ax \leqslant \overline{\beta}$), но, как показывает табл. 1, ее частными случаями являются многие другие интересные комбинаторные задачи (очевидно, что список, приведенный в табл. 1, не является исчерпывающим).

В случае, если одно из двух условий $x \in X$ или $x \in Y$ может быть представлено в форме одного алгебраического ограничения (или нескольких ограничений) типа (11), можно применить технику, описанную в § 5. Для общего случая (где ни одно из двух условий $x \in X$ или $x \in Y$ не сводится к простому алгебраическому условию) мы изучим здесь другой технический прием, который предполагает только, что мы можем эффективно решить каждую из задач вида

$$\begin{aligned} &\gamma x \to \min, \\ &x \in X; \end{aligned} \qquad \begin{aligned} &\gamma x \to \min, \\ &x \in Y, \end{aligned}$$

при любом векторе стоимости γ.

Идея метода состоит в представлении задачи пересечения в эквивалентной форме

$$(\mathrm{IP})' \qquad \begin{aligned} &cx \to \min, \\ &x \in X, \\ &y \in Y, \\ &x - y = 0. \end{aligned} \tag{15}$$

Связывая с ограничениями (15) вектор множителей Лагранжа λ (без ограничений на знак его компонент), мы можем определить соответствующее этому вектору λ значение двойственной функции $w(\lambda)$ формулой

$$w(\lambda) = \min_{x \in X} \{(c + \lambda) x\} + \min_{x \in Y} \{-\lambda y\}, \tag{16}$$

так что $w(\lambda)$ определяется с помощью раздельного решения (декомпозиции) задачи минимизации для X и задачи минимизации для Y. Как и в случаях задач о кратчайшем пути с ограничениями из § 5, это приводит к раздельному решению для данного λ задачи о кратчайшем пути и задачи о рюкзаке. Если $\bar{x}(\lambda)$ и $\bar{y}(\lambda)$ обозначают оптимальные решения, полученные для каждого из слагаемых в правой части (16), то вектор $\bar{x}(\lambda) - \bar{y}(\lambda)$ есть вектор субградиента функции w в точке λ. Следовательно, двойственная задача может быть решена с помощью алгоритма субградиента для получения минорант.

Приложение этой задачи к решению задачи о кратчайшем пути с ограничением типа (CSP_1) или (CSP_2) (см. § 5) было изучено в [56] и [68], где показано, что границы, полученные переходом к двойственной к (IP) задаче с помощью (16), могут быть только больше (или, в худшем случае, равносильны) границам, полученным с помощью перехода к двойственной к (CSP) или (CSP_2) задаче с помощью (14). Это можно объяснить тем фактом, что, в отли-

чие от ослабления в § 5, в условии (16) минимизация по множеству Y (задача о рюкзаке) явно учитывает условие целочисленности переменных. Хотя скачки двойственности для задач типа (CSP_2) могут оказаться значительными (как правило, до 30 %, и это, кстати, является показателем трудности этих задач), но вычислительные результаты [68] показывают, что рассматриваемый сейчас подход позволяет систематически получать для задачи о кратчайшем пути с двойным условием равенства (CSP_2) хорошие реализуемые решения (а часто даже оптимальные).

Важно отметить, что этот общий класс задач, включающий такие трудные задачи, как задача о коммивояжере при наличии ограничений, задача о двумерном рюкзаке и т. д. (см. табл. 1) содержит также и легко исследуемые задачи. Действительно, в случае если X и Y суть два матроида над E, мы получаем задачу, известную под названием «пересечение двух матроидов», для которой разработаны эффективные (полиномиальные) алгоритмы [46, 16]. Во всех случаях пересечения двух матроидов в [24] установлена теорема о минимаксе, которая есть не что иное, как свойство сильной двойственности для задачи $(IP)'$ и ее двойственной задачи. Этот результат не удивителен, так как практически все известные полиномиальные задачи имеют свойство сильной двойственности (нулевые скачки двойственности). Обобщается ли на случай пересечения двух матроидов теорема Радо — Эдмондса (ср. [17]), иначе говоря, позволяет ли это свойство сильной двойственности охарактеризовать задачи о пересечении двух матроидов?

Для конкретности предположим, что: а) множества X и Y являются *системами независимости* над E*); б) задача $(IP)'$ с двойственной задачей в смысле (14) дает нулевой скачок двойственности для любого вектора стоимости $c \in R^n$. Следует ли из условий а) и б), что X и Y суть матроиды? Этот вопрос остается нерешенным.

§ 7. Обобщенная задача об ассигнованиях

Эта задача возникает, например, при определении оптимального состава воздушного флота с заданным средним значением для данной авиакомпании (см. [49]).

Пусть $I = \{1, 2, \ldots, m\}$ — множество типов самолетов, соревнующихся между собой, и $J = \{1, 2, \ldots, n\}$ — множество подлежащих обслуживанию маршрутов.

Для каждого маршрута $j \in J$ и для каждого типа самолетов $i \in I$ предположим, что нам известна частота f_{ij} полетов самолетов i по маршруту j (например, число полетов в неделю), позволяющая удовлетворить коммерческим ограничениям и приводящая к максимальной полной прибыли при эксплуатации, равной β_{ij}.

Тогда определение оптимального состава воздушного флота состоит в присвоении каждому маршруту j такого типа самолетов i, чтобы стоимость (расходы по эксплуатации минус прибыль) была минимальной, с полным учетом

---

*) X есть *система независимости*, если
1) $0 \in X$;
2) $x \in X$ и $0 \leqslant x' \leqslant x \Rightarrow x' \in X$.

ограничений на число имеющихся в наличии самолетов каждого типа (ограничения на размеры воздушного флота).

Определим двузначные целочисленные переменные x_{ij} следующим образом: $x_{ij} = 1$, если тип самолетов i присвоен маршруту j, и $x_{ij} = 0$ в противном случае. Ограничения на размеры воздушного флота выражаются тем, что для каждого типа самолетов i задано полное число часов полета, которые могут быть использованы этим типом самолетов; обозначим это число через b_i. Если h_{ij} — число часов, которое требуется от самолета типа i для осуществления им полета по маршруту j, то сумма часов для самолетов типа i, позволяющая покрыть требования маршрута j, равна

$$a_{ij} = f_{ij}h_{ij}.$$

Искомое назначение $x = (x_{ij})$ должно, таким образом, удовлетворять m ограничениям:

$$\sum_{j \in J} a_{ij}x_{ij} \leqq b_i \qquad \forall i \in I.$$

Если мы обозначим через F_i почасовые финансовые расходы, соответствующие каждому типу самолетов i, то стоимость ассигнования (выделения) типа самолетов i на маршрут j равна $c_{ij} = F_i a_{ij} - \beta_{ij}$.

Тогда задача сводится к решению следующей задачи целочисленного линейного программирования (с двузначными переменными x_{ij}):

$$\sum_{i \in I} \sum_{j \in J} c_{ij}x_{ij} \to \min,$$

$$\sum_{j \in J} a_{ij}x_{ij} \leqq b_i \qquad \forall i \in I, \tag{17}$$

(GAP)

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J, \tag{18}$$

$$x_{ij} = 0 \text{ или } 1.$$

Эта задача известна как *обобщенная задача о назначениях* на основании следующего замечания: если $|I| = |J| = n$ и все коэффициенты a_{ij}, как и все правые части b_i, равны 1, то мы снова получаем классическую задачу о назначениях

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} \to \min,$$

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I,$$

(AP)

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J,$$

$$x_{ij} = 0 \text{ или } 1,$$

которая может быть эффективным образом решена с помощью «венгерского алгоритма» [45] или, равноспильным образом, как задача о потоке с минимальной стоимостью в двухчастном графе (см., например, [32], гл. 5).

Что же касается обобщенной задачи о назначениях, то она входит в класс *NP*-полных задач (см. [26]), и для ее решения неизвестен никакой

полиномиальный алгоритм. Для ее решения могут быть использованы различные подходы, использующие лагранжево ослабление.

В [49] предлагается ослаблять ограничения (17), связывая с ними вектор множителей $\lambda = (\lambda_i)_{i \in I}$ с неотрицательными элементами. Значение двойственной функции очень легко получается с помощью решения задачи

$$\omega(\lambda) = \sum_{j \in J} \sum_{i \in I} \tilde{c}_{ij} x_{ji} - \sum_{i \in I} \lambda_i b_i,$$

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J,$$

$$x_{ij} = 0 \text{ или } 1,$$

где $\tilde{c}_{ij} = c_{ij} + \lambda_i a_{ij}$ для всех i, j.

Для этого достаточно определить $\tilde{c}_{i_0 j} = \min\limits_{i \in I} \{\tilde{c}_{ij}\}$ для каждого $j \in J$ и присвоить соответствующему переменному значение 1 (другие переменные x_{ij} с $i \in I - \{i_0\}$ полагаются равными нулю). В [49] сообщается о замечательных результатах применения этого подхода к задачам о назначениях для воздушного флота.

Второй подход, предложенный ранее в [69], состоит в предпочтении ослаблений ограничений (18) ослаблению ограничений (17).

Тогда вычисление двойственной функции требует на каждом шаге решения некоторого семейства задач о рюкзаке (в количестве, равном числу ограничений $i \in I$).

Если эти задачи решаются как непрерывные, то можно показать, что этот подход совершенно равносилен предыдущему (и полученные нижние границы — совершенно такие же, как и в предыдущем случае). Если же эти задачи о рюкзаке решаются как целочисленные задачи, то мы получим, вообще говоря, строго лучшие миноранты.

Следовательно, второе из рассматриваемых ослаблений позволяет получить лучшее приближение целочисленного оптимума, но ценой увеличения объема вычислений, который может оказаться много большим, чем для первой задачи (и даже невыполнимым для задач большого размера).

§ 8. Другие примеры приложения лагранжева ослабления в задачах комбинаторной оптимизации

Кроме задач, достаточно подробно изученных в § 1—7, и которые могут рассматриваться как представителями класса наиболее характерных задач, техника лагранжева ослабления была с успехом применена во многих других задачах, в частности:

— подход к обобщенным задачам целочисленного программирования с точки зрения теории групп (см. § 4 гл. 7);

— задачи покрытия (см. [6]);

— некоторые задачи управления производством (см. [21]);

— задачи о запуске теплоцентралей (см. [63]);

— задачи о согласованных мультипотоках и в целых числах (см. [42; 32, гл. 6]);

— задачи о потоках с дополнительными ограничениями (см. [57]);

— квадратичная задача о рюкзаке (см. [40]);

— задача о наименьшем дереве с ограничениями на степени вершин (см. [32], гл. 4, упражнение 7);

— задачи о пересечении матроидов (см. [52]);

— задачи о матроидах с условиями четности (см. [10]);

— задачи о распределении рабочего времени (см. [72]);

— задачи о наименьшем дереве с ограниченными емкостями, возникающими в оптимизации сетей вычислительных машин (см. [27, 28]).

В заключение отметим, что для точного решения трудных комбинаторных задач и/или задач больших размерностей лагранжево ослабление все более и более плодотворно используется в ходе численных процедур в сочетании с другими методами, в частности, методами сечений и техникой полиэдральной комбинаторики, которая добивается максимально полной характеризации ограничений, определяющих выпуклую оболочку целых точек. Дополнительные ограничения (сечения, грани), порожденные этой техникой, могут быть исследованы с помощью лагранжева ослабления, и позволяют либо улучшить нижние границы, либо получать оптимальные целые решения. В [4, 67] или [59] можно найти обзор недавно достигнутых успехов в этой области.

СПИСОК ЛИТЕРАТУРЫ

1. Agard J., Arabeyre J. P., Vautier J. (1967), Génération automatique de rotations d'équipages, R. A. I. R. O., 6, p. 107-117.
2. Aneja Y. P. (1980), An integer linear programming approach to the Steiner problem in graphs, *Networks* 10, p. 167-178.
3. Arditti D., Minoux M. (1983), Un algorithme de détermination de partition utilisant la dualité lagrangienne, *Actes regroupés des journées de classification de Toulouse* (*mai* 1980) *et de Nancy* (*juin* 1981) (I. C. Lerman, editeur).
4. Bachem A., Grötschel M. (1982), New aspects of Polyhedral Theory in: *Modern Applied Mathematics. Optimization and Operations Research* (B. Korte éd.), North Holland.
5. Balas E., Christofides N. (1981), A restricted lagrangian approach to the Travelling Salesman Problem, *Mathematical Programming* 21, p. 19-46.
6. Balas E., Ho A. (1980), Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study *Mathematical Programming* 12, p. 37-60.
7. Balas E., Padberg M. W. (1975), *Set partitioning in Combinatorial Programming: Methods and Applications* (B. Roy ed.), Reidel, p. 205-258.
8. Bazaraa L. S., Goode J. J. (1977), The travelling Salesman Problem: a duality approach, *Mathematical Programming* 13, p. 221-237.
9. Bilde O., Krarup J. (1967), Bestemmelse of optimal beliggenhed of productionssteder, *Research Report*, IMSOR, Danmark Tekniske Hojskole.
10. Camerini P/M., Maffioli F. (1978), Heuristically guided algorithms for *k*-parity matraid problems, *Discrete Math.* 21, p. 103-116.
11. Christofides N. (1970), The shortest hamiltonian chain of a graph, *J. SIAM* 19, p. 689-697.
12. Claus A., Maculan N. (1983), Une nouvelle formulation du problème de Steiner sur un graphe, *Prépublication* n° 280, Centre de Recherche sur les Transports, Universite de Montreal.
13. Crowder H. P., Padberg M. W. (1980), Solving large Scale Symmetric Travelling Salesman Problems to optimality, *Management Science* 26, p. 495-509.
14. Derigs U. (1983), Solving Matching Problems via Shortest Paths Techniques, *Rep.* n° 83263-OR, Institut für Ökonometrie und Operations Research, Universitat Bonn,

15. Desrosiers J., Pelletier P., Soumis F. (1981), Plus court chemin avec contraintes d'horaires, *Rapport de Recherche* 81—21, École des Hautes Etudes Commerciales, Montreal.
16. Edmonds J. (1965), Maximum matching and a polyhedron with 0—1 vertices, *Journal Res. Nat. Bur. Stand*; 69-b (1—2), p. 125-130.
17. Edmonds J. (1971), Matroids and the greedy algorithm, *Mathematical Programming* 1, p. 127-136.
18. Edmonds J. (1979), Matroid intersection, *Annals of Discrete Mathematics* 4, p. 39-49.
19. Erlenkotter D. (1978), A dual-based proceduce for uncapacitated facility location, *Operations Research* 26, 1, p. 992-1009.
20. Everett H. (1963), Generalized Lagrange Multiplier Method for solving problems of optimums allocation of resources, *Ops. Res.* 11, p. 399-417.
21. Fischer M. L. (1976), A dual algorithm for the one machine scheduling problem, *Mathematical Programming* 11, p. 229-251.
22. Fischer M. L. (1981), The Lagrangian Relaxation Method for solving Integer Programming Problems, *Management Science* 27, 1, p. 1-18.
23. Fischer M. L., Jaikumar R., Van Wassenhove L. N. (1981), A multiplier adjustment method for the generalized assignment problem, *Preprint* 81-07-06, Dept. of Decision Sciences, the WHARTON School, University of Pennsylvaniya, Philadelphia PA.
24. Frank A. (1981), A weighted matroid intersection algorithm, *Journal of Algorithms* 2, p. 328-336.
25. Galil Z., Micali S., Gabow H. (1983). Priority Queues with Variables Priority and an $O(|E|\,|V|\log|V|)$ Algerithm for finding a maximal weight matching in general Graphs.
26. Garey M. R., Johnson D. S. (1979), *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman & Co.
27. Gavish B. (1982), Topological design of Centralized Computer Networks — Formulations and algorithms, *Networks* 12, p. 355-377.
28. Gavish B. (1982), Formulations and algorithms for theo capacitated minimal directed tree problem, *Journal of A. C. M.*, 30, p. 118-132.
29. Geoffrion A. M. (1974), Lagrangian relaxation and its uses in integer programming, *Mathematical Programming Study* 2, p. 82-114.
30. Gilmore P. C., Gomory R. E. (1966), The theory and computation of knapsack functions, *Operations Research* 13, p. 1045-1074.
31. Gomory R. E., Hu T. C. (1962), An application of generalized linear programming to network flows, *S. I. A. M. J. Appl. Math.*, 10, 2, p. 260-283.
32. Gondran M., Minoux M. (1979), *Graphes et Algorithmes*, Eyrolles, Paris (traduction anglaise, J. Wiley & Sons, 1984).
33. Grötschel M., Padberg M. W. (1979), On the Symmetric Travelling Salesman Problem II: lifting theorems and facets, *Mathematical Programming* 16, p. 281-302.
34. Grotschel M., Padberg M. W. (1982a), Polyhedral aspects of the Travelling Salesman Problem I: Theory, in: *The Travelling Salesman Problem* (E. Lawler, J. Lenstra, A. Rinnooy Kaneds), Wiley, 1983.
35. Grötschel M., Padberg M. W. (1982b), Polyhedral aspects of the Travelling Salesman Problem II: Computation, in: *The Travelling Salesman Problem* (E. Lawler, J. Lenstra, A. Rinnooy Kaneds), Wiley, 1983.
36. Guyard L. (1985), Le problème de l'arbre de Steiner: Modélisation par Programmation linéaire et résolution par des techniques de décomposition, *Thèse Docteur — Ingenieur*, École Nationale Supérieure des Télécommunications, Paris.
37. Guyard L., Minoux M. (1983), Expérimentation numérique d'une nouvelle approche du problème de l'arbre de Steiner par relaxation lagrangienne. Voir Guyard (1985).
38. Handrel G. Y., Zang I. (1980), A Dual Algorithm for the Constrained Shortest Path Problem, *Networks* 10, p. 293-310.
39. Hansen K. H., Krarup J. (1974), Improvements of the Held-Karp algorithm for the symmetric Travelling Salesman Problem, *Mathematical Programming* 7, p. 87-96.

40. Hansen P., Chaillou P., Mahieu Y. (1983), Best network flow bounds for the quadratic knapsack problem, *International Workshop on Network flow Theory*, NETFLOW 83, Pisa (Italie).
41. Held M., Karp R. M. (1971), The Travelling Salesman Problem and Minimum Spanning Trees: Part II, *Mathematical Programming* 1, p. 6-25.
42. Held M., Wolfe P., Crowder H. P. (1974), Validation of subgradient optimization, *Mathematical Programming* 6, p. 62-88.
43. Heurgon E. (1972), Un problème de recouvrement: l'habillage des horaires des linges d'autobus, R. A. I. R. O., 1, p. 13-29.
44. Kruskal J. B. (1956), On the shortes spanning subtree of a graph and the travelling salesman problem, *Proc. Am. Math. Soc.* 71, p. 48-50.
45. Kuhn H. W. (1955), The Hungarian method for the assignement problem, *Nav. Res. Log. Quart.* 2, p. 83-97.
46. Lawler E. L. (1975), Matroid intersection algorithms, *Mathematical Programming* 9, p. 31-56.
47. Lawler E. L. (1976), *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston.
48. Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., Shmoys D. B. (1985), *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*, J. Wiley & Sons, 465 p.
49. Legendre J. P., Minoux M. (1977), Une application de la notion de dualité en programmation en nombres entiers — sélection et affectations optimales d'une flotte d'avions, *R. A. I. R. O.* 11, 2, p. 201-222.
50. Lovász L. (1979), Graph Theory and Integer Programming, Annals of *Discrete Mathematics* 4, p. 141-158.
51. Maculan N. (1987), The Steiner Problem in Graphs, in: *Surveys in Combinatorial Optimization* (S. Martello, G. Laporte, M. Minoux, C. Ribeiro eds.) *Anals of discrete Mathematics* 31, North — Holland, p. 185-212.
52. Maffioli F. (1976), Subgradient optimization, matroid problems and heuristic evaluation, Proc. *7th IFIP Conf. on Optimization Techniques* (Nice, France, 1975), Springer Verlag.
53. Marsten R. E. (1974), An algorithm for Large Set Patrioning Problems, *Management Scilnce* 20, p. 774-787.
54. Milliotis T. (1976), Integer Programming approaches to the Travelling Scalsman Problem, *Mathematical Programming* 10, p. 367-378.
55. Minoux M. (1975), Plus court chemin avec contraintes, algorithmes et applications, *Annales des Télécommunications* 30, 11-12, p. 383-394.
56. Minoux M. (1982), Lagrangian Relaxation of the Constrained Shortest path problem, *Private Communication to* C. Ribeiro.
57. Minoux M. (1983), A Lagrangian relaxation approach to the minimum cost flow problem with additional constraints, *Note interne* CNET.
58. Minoux M. (1983), A new polynomial culling plane algorithm for maximum weight matchings in general graphs, Preprint n° 302, Department d'Informatique et de Recherche Opérationelle, Université de Montréal.
59. Minoux M. (1986), Recent developments in the Theory of Cutting planes in: *Mathematical Programming: Theory and Algorithms* Chapter 7, Wiley & Sons.
60. Minoux M. (1987), A class of Combinatorical Problems with Polynomially Solvable Large Scale Set Covering, Partitioning relaxationss, *R. A. I. R. O.*, vol. 21, N° 2, p. 105-136.
61. Minoux M., D'Hoeraene J., Chauvin J. (1980), Reduction du kilometrage ou du temps ou du coût direct en prospection géophysique, Proc. Symp. IFAC/IFORS, Toulouse (6-8 mars 1979).
62. Minoux M., Serreault J. Y. (1981), Subgradient optimization and large scale programming: an application to optimum multicommodity network synthesis with security constraints, *R. A. I. R. O.*, vol. 15, n° 2, p. 185-203.
63. Muckstadt J., Koenig S. A. (1977), An application of lagrangian relaxation to scheduling in power generation systems, *Operations Research* 25, p. 387-403.
64. Mulvey J. M., Crowder H. P. (1979), Cluster Analysis: an application on lagrangian relaxation, *Management Science* 25, 4, p. 329-340.

65. Nemhauser G. L., Weber G. M. (1979), Optimal Set partitioning, matchings and Lagrangian duality, *Nav. Res. Log. Quart.* 26, 4, p. 553-563.
66. Prim R. C. (1957), Short connections networks and some generalizations, *Bell Syst. Techn.* J., 36, p. 1389.
67. Pulleyblank W. R. (1983), Polyhedral Combinatorics, in: *Mathematical Programming: the state of the Art* (A. Bachem, M. Grötschel, B. Korte eds.), Springer Verlag, p. 312-345.
68. Ribeiro C. (1983), Algorithmes de recherche de plus courts chemins avec contraintes: étude théorique, implémentation et parallélisation, Départment de Mathématiques Appliquées du CNET, *Thèse de Docteur-Ingénieur soutenue à l'ENST le 13 octobre* 1983.
69. Ross G. T., Soland R. M. (1975), A Branch and Bound algorithm for the generalized assignment problem, *Mathematical Programming* 8, p. 91-103.
70. Shapiro J. F. (1979), A survey of lagrangian techniques for discrete optimization, *Annals of Discrete Mathematics*, 5, p. 113-138.
71. Shier D. R. (1976), Iterative Methods for determing k-shortest paths in a network, *Networks* 6, p. 205-229.
72. Tripathy A. (1980), A lagrangian relaxation approach to course time-tabling, J. ORSA 31, p. 599-603.
73. Wong R. T. (1980), Integer Programming formulations of the Travelling Salesman problem, *Proc. IEEE Internat. Conf. on Circuits and Computers.*
74. Yao A. C. C. (1975), An $O(|E| \log \log |V|)$ algorithm for finding minimum spanning trees, *Info. Processing letters* 4, p. 21-25.
75. Yen J. Y. (1971), Finding the k-shortest loopless paths in a network, *Management Science* 17, p. 711-715.

СПИСОК ОБОЗНАЧЕНИЙ

Множества

$\mathbf{R}$ — множество действительных чисел
$\mathbf{N}$ — множество натуральных чисел
$\mathbf{Z}$ — множество целых чисел
$\{x, y, z\}$ — множество из трех элементов: x, y, z
$x \in X$ — x есть элемент множества X
$x \notin X$ — x не принадлежит множеству X
$A \subset X$ — A содержится в X (и может быть $A = X$)
$A \subsetneq X$ — A строго содержится в X
$|X|$ — мощность множества X: число элементов множества X
$A \cup B$ — объединение множеств A и B
$A \cup \{x\}$ — объединение множества A и множества, состоящего из одного элемента x (часто обозначается $A + x$, если не возникает путаницы)
$X - A$ — множество элементов из X, не принадлежащих A
$\{x \mid x \ldots\}$ — множество таких элементов x, что...
$\exists x$: — существует такое x, что
$\forall x \in X$ для любого элемента $x \in X$
$A \times B$ — декартово произведение множеств A и B (множество пар (a, b) с $a \in A$ и $b \in B$)
$(P) \Rightarrow (Q)$ — свойство (P) влечет свойство (Q)
$[a, b]$ — замкнутый интервал (отрезок): $\{x \mid x \in \mathbf{R};\ a \leqslant x \leqslant b\}$, где a и b — действительные числа $(a \leqslant b)$
$]a, b[$, (a, b) — открытый интервал (интервал): $\{x \mid x \in \mathbf{R};\ a < x < b\}$, где a и b — действительные числа $(a < b)$
$[a, b[$, $[a, b)$ — полуоткрытый интервал (полуинтервал): $\{x \mid x \in \mathbf{R};\ a \leqslant x < b\}$, где a и b — действительные числа $(a < b)$
$\operatorname{int} S$ — внутренность множества S
$\operatorname{cl}(S)$ — замкнутые множества S
$\inf\limits_{x \in X}(x)$ — если X имеет миноранту (нижнюю границу), то $\operatorname{int}(x)$ по определению есть наибольшая из минорант множества X. Если X не имеет миноранты, то по соглашению $\inf\limits_{x \in X}(x) = -\infty$; если $X = \varnothing$, то по соглашению $\inf\limits_{x \in X}(x) = +\infty$, где $X \neq \varnothing$ — множество из $\mathbf{R}$)
$\sup\limits_{x \in X}(x)$ — по определению $\sup\limits_{x \in X}(x) = -\inf\limits_{x \in X}(-x)$

Последовательности и пределы

$\{x^k\}_{k \in N}$ — последовательность элементов x^k для $k = 1, 2, \ldots$
$\{x^k\}_{k \in I}$ — последовательность элементов x^k, где k пробегает множество индексов I (будем просто писать $\{x^k\}$, когда не возникает путаницы с множеством индексов)
$\limsup \{x^k\}$ — верхний предел числовой последовательности $\{x^k\}_{k \in N}$ (действительных чисел). По определению это есть такое действительное число y, что:
1) $\forall \varepsilon > 0\ \exists K$: $\forall k > K\ \ x^k < y + \varepsilon$;
2) $\forall \varepsilon > 0$, $\forall K > 0\ \exists k > K\ \ x^k > y - \varepsilon$;

$\liminf\{x^k\}$ — нижний предел числовой последовательности $\{x^k\}_{k\in N}$ (действительных чисел). По определению $\liminf\{x^k\} = -\limsup\{-x^k\}$

$\lim\{x^k\}$ — предел последовательности $\{x^k\}$. Для некоторой последовательности $\{x^k\}$ элементов из $\mathbf{R}^n$ элемент $x \in \mathbf{R}^n$ называется пределом, если $\liminf\{x^k\} = \limsup\{x^k\} = x$. Последовательность, имеющая предел, называется сходящейся.

Векторы и матрицы

$\mathbf{R}^n$ — n-кратное декартово произведение множества на себя (множество векторов с n действительными компонентами)

$\mathbf{R}^{n+}$ — положительный ортант пространства $\mathbf{R}^n$: множество векторов из $\mathbf{R}^n$, все компоненты которых неотрицательны

$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$ — вектор из $\mathbf{R}^n$ с компонентами $x_1, \dots, x_n$ (будет в книге отождествляться со столбцом матрицы)

x^T — вектор, транспонированный по отношению к вектору $x \in \mathbf{R}^n$ (матрица-строка)

$x^T y$, $x^T \cdot y$ — скалярное произведение векторов x и y

$\|x\|$ — евклидова норма вектора x

$x \geqslant y$ — каждая компонента вектора x больше или равна соответствующей компоненте вектора y

$A = [a_{ij}]_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$ — матрица из m строк и n столбцов; a_{ij} — элемент матрицы, стоящий в i-й строке и j-м столбце (часто, если не возникает путаницы, обозначают $[a_{ij}]$)

AB — произведение матрицы A на матрицу B

A^T — матрица, транспонированная к матрице A

$\det A$ — определитель (квадратной) матрицы A

$\operatorname{rang} A$ — ранг матрицы A (наибольший порядок регулярной квадратной подматрицы матрицы A)

Функции

$C[a, b]$ — векторное пространство функций, непрерывных на $[a, b]$

$C^1[a, b]$ — векторное пространство функций, непрерывно дифференцируемых на $[a, b]$ (функции, принадлежащие $C^1[a, b]$, называют функциями из класса C^1)

$C^p[a, b]$ — векторное пространство функций, p раз непрерывно дифференцируемых на $[a, b]$ (функций из класса C^p)

$\nabla f(x)$ — если f: $\mathbf{R}^n \to \mathbf{R}$ — функция переменных $x_1, \dots, x_m$, то $\nabla f(x)$ есть градиент функции f в точке x, т. е. n-вектор с компонентами $\dfrac{\partial f}{\partial x_1}(x), \dots, \dfrac{\partial f}{\partial x_n}(x)$ (при условии, что частные производные в точке x существуют)

$\dfrac{\partial f}{\partial x}(x)$ — обозначение, эквивалентное обозначению $\nabla f^T(x)$, т. е. обозначение вектора, транспонированного к $\nabla f(x)$ $\left($значит, $\dfrac{\partial f}{\partial x}(x)$ отождествляется с матрицей-строкой с компонентами $\dfrac{\partial f}{\partial x_1}(x), \dots, \dfrac{\partial f}{\partial x_n}(x)\right)$

$\nabla_x f(x, y)$ — если f: $\mathbf{R}^{n+m} \to \mathbf{R}$ — функция переменных $x = (x_1, \dots, x_n)$ и $y = (y_1, \dots, y_m)$, то $\nabla_x(x, y)$ есть градиент по x функции f в точке (x, y), т. е. n-вектор с компонентами $\dfrac{\partial f}{\partial x_j}(x, y)$ $(j = 1, \dots, n)$ (при условии, что частные производные в точке (x, y) существуют)

$\dfrac{\partial f}{\partial x}(x, y)$ — эквивалентное обозначение для $\nabla_x f^T(x, y)$

$\nabla^2 f(x)$ — если $f\colon \mathbf{R}^n \to \mathbf{R}$ — функция переменных $x_1, \ldots, x_n$, то $\nabla^2 f(x)$ обозначает гессиан функции f в точке x, т. е. действительную симметричную матрицу порядка n с (i, j)-м членом $\dfrac{\partial^2 f}{\partial x_i\, \partial x_j}(x)$ (при условии, что частные производные в точке x существуют)

$\nabla_x^2 f(x, y)$ — если $f\colon \mathbf{R}^{n+m} \to \mathbf{R}$ — функция переменных $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_m)$, то $\nabla_x^2 f(x, y)$ означает гессиан по x функции f в точке (x, y), т. е. матрицу порядка n: $\left[\dfrac{\partial^2 f}{\partial x_i\, \partial x_j}(x, y)\right]$ (при условии, что частные производные в точке (x, y) существуют)

$\dfrac{\partial g}{\partial x}(x)$ — если $g\colon \mathbf{R}^n \to \mathbf{R}^m$ — функция, которая вектору $x = (x_1, \ldots, x_n)$ ставит в соответствие столбец $g(x) = \begin{bmatrix} g_1(x) \\ \cdots \\ g_m(x) \end{bmatrix}$, то $\dfrac{\partial g}{\partial x}(x)$ обозначает якобиан функции g в точке x, т. е. $m \times n$-матрицу $\left[\dfrac{\partial g_i}{\partial x_j}(x)\right]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}}$ (при условии, что частные производные в точке x существуют)

$\partial f(x)$ — субдифференциал функции f по x: множество субградиентов функции f по x (для выпуклой или вогнутой функции). Операторы min и max

$\inf\limits_{x \in X} \{f(x)\}$ — по определению $\inf\limits_{x \in X} \{f(x)\} = \inf\limits_{y \in Y} (y)$, где $f\colon \mathbf{R}^n \to \mathbf{R}$, $X \subset \mathbf{R}^n$, $X \neq \varnothing$, $Y = \{y \mid y \in \mathbf{R};\ \exists x \in X\colon y = f(x)\}$

$\sup\{f(x)\}$ — по определению $\sup\limits_{x \in X} \{f(x)\} = -\inf\limits_{x \in X} \{-f(x)\}$

$\min\limits_{x \in X} \{f(x)\}$ — пусть $\mathbf{R}^n \to \mathbf{R}$ и $X \subset \mathbf{R}^n$; если $-\infty < \inf\limits_{x \in X} \{f(x)\} < +\infty$ и если существует также $\bar{x} \in X$, что $f(\bar{x}) = \inf\limits_{x \in X} \{f(x)\}$, то пишут $\min\limits_{x \in X} \{f(x)\} = f(\bar{x})$; если $\inf\limits_{x \in X} \{f(x)\} = -\infty$, то по соглашению пишут $\min\limits_{x \in X} \{f(x)\} = -\infty$ (и тогда говорят, что f не имеет конечного оптимума); если $X = \varnothing$, то по соглашению пишут $\min\limits_{x \in X} \{f(x)\} = +\infty$

$\max\limits_{x \in X} \{f(x)\}$ — по определению $\max\limits_{x \in X} \{f(x)\} = -\min\limits_{x \in X} \{-f(x)\}$

$\min\{a; b; c\}$ — минимальное из трех действительных чисел: a, b, c

ЛИТЕРАТУРА НА РУССКОМ ЯЗЫКЕ

1. Гасс С. Линейное программирование. (Методы и приложения.)— М.: Физматгиз, 1961.
2. Гилл Ф., Мюррей У. Численные методы условной оптимизации/Пер. с англ.— М.: Мир, 1977.
3. Гловински Р., Лионс Ж.-Л., Тремольер Р. Численное исследование вариационных неравенств/Пер. с фр.— М.: Мир, 1979.
4. Гурин Л. Г., Поляк Б. Т., Райк Е. В. Методы проекций для отыскания общей точки выпуклых множеств.— Журн. выч. мат. и мат. физ., 1967, 7, 6, с. 1211—1228.
5. Данциг Дж. Линейное программирование, его обобщения и применения/ Пер. с англ.— М.: Прогресс, 1966.
6. Ермольев В. М. Методы решения нелинейных экстремальных задач.— Кибернетика, 1966, 4, с. 1—17.
7. Зангвилл У. Нелинейное программирование. Единый подход/Пер. с англ.— М.: Сов. радио, 1973.
8. Зенкевич О. Метод конечных элементов в технике/Пер. с англ.— М.: Мир, 1975.
9. Зойтендейк Г. Методы возможных направлений/Пер. с англ.— М.: Мир, 1963.
10. Карлин С. Математические методы в теории игр/Пер. с англ.— М.: Мир, 1961.
11. Курант Р., Гильберт Д. Методы математической физики/Пер. с англ.— М.— Л.: ГИТГЛ, 1951.
12. Левин А. Ю. Об одном алгоритме минимизации выпуклых функций.— ДАН СССР, 1965, 160, 6, с. 1244—1247.
13. Лионс Ж.-Л. Оптимальное управление системами, описываемыми уравнениями с частными производными/Пер. с франц.— М.: Мир, 1972.
14. Лэсдон Л. С. Оптимизация больших систем/Пер. с англ.— М.: Наука, 1975.
15. Пойа Г. Математика и правдоподобные рассуждения/Пер. с англ.— М., Мир, 1963.
16. Полак Е. Численные методы оптимизации/Пер. с англ.— М.: Мир, 1974.
17. Поляк Б. Т. Один общий метод решения экстремальных задач.— ДАН СССР, 1967, 174, 1, с. 33—36.
18. Поляк Б. Т. Методы минимизации при наличии ограничений/В сб. Математический анализ. Сер. Итоги науки и техники.— М.: ВИНИТИ АН СССР, 1974, т. 12, с. 147—198.
19. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов.— М.: Наука, 1976.
20. Рокафеллар Р. Выпуклый анализ/Пер. с англ.— М.: Мир, 1973.
21. Сеа Ж. Оптимизация, теория и алгоритмы/Пер. с англ.— М.: Мир, 1973.
22. Скоков В. А. Замечание к методам минимизации, использующим операцию растяжения пространства.— Кибернетика, 1974, 4, с. 115—117.
23. Стренг Г., Фикс Дж. Теория метода конечных элементов/Пер. с англ.— М.: Мир, 1977.

24. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной оптимизации/Пер. с англ.— М.: Мир, 1972.
25. Хачиян Л. Г. Полиномиальный алгоритм в линейном программировании.— ДАН СССР, 1979, 244, 5, С. 1093—1096.
26. Хедли Дж. Нелинейное и динамическое программирование/Пер. с англ.— М.: Мир, 1967.
26а. Химмельблад Д. Прикладное нелинейное программирование/Пер. с англ.— М.: Мир, 1973.
27. Ховард Р. А. Динамическое программирование и марковские процессы/Пер. с англ.— М.: Мир, 1967.
28. Ху Т. Целочисленное программирование и потоки в сетях/Пер. с англ.— М.: Мир, 1974.
29. Шор Н. З. О скорости сходимости обобщенного градиентного спуска.— Кибернетика, 1968, 3, с. 98—99.
30. Шор Н. З. О скорости сходимости метода обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1970, 2, с. 80—85.
31. Шор Н. З. Исследование сходимости метода градиентного типа с растяжением пространства в направлении разности двух последовательных градиентов.— Кибернетика, 1975, 4, с. 48—53.
32. Шор Н. З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования.— Кибернетика, 1977, 1, с. 94—95.
33. Шор Н. З., Журбенко Н. Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов.— Кибернетика, 1971, 3, с. 51—59.
34. Шор Н. З., Шабашова Л. П. О решении минимаксных задач методом обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1972, 1, с. 82—88.
35. Экланд И., Темам Р. Выпуклый анализ и вариационные проблемы/Пер. с англ.— М.: Мир, 1979.

СПИСОК ЛИТЕРАТУРЫ, ДОБАВЛЕННОЙ ПРИ ПЕРЕВОДЕ

1. Бахвалов Н. С. Численные методы.— М.: Наука, 1975.
2. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы.— М.: Наука, 1987.
3. Евтушенко Ю. Г. Методы решения экстремальных задач и их применение в системах оптимизации.— М.: Наука, 1982.
4. Карманов В. Г. Математическое программирование.— М.: Наука, 1986.
5. Марчук Г. И. Методы расщепления.— М.: Наука, 1988.
6. Михалевич В. С., Гупал А. М., Норкин В. И. Методы невыпуклой оптимизации.— М.: Наука, 1987.
7. Моисеев Н. Н., Иванилов Ю. П., Столярова Е. М. Методы оптимизации.— М.: Наука, 1978.
8. Пшеничный Б. Н. Выпуклый анализ и экстремальные задачи.— М.: Наука, 1980.
9. Сухарев А. Г., Тимохов А. В., Федоров В. В. Курс методов оптимизации.— М.: Наука, 1986.
10. Шор Н. З. Методы минимизации невыпуклых функций и их приложения.— Киев: Наукова думка, 1979.
11. Winter P. Steiner problem in networks. A survey // Networks.— 1987.— V. 17, № 2.— P. 129—167.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Научное издание

МИПУ Мышель

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Теория и алгоритмы

Заведующий редакцией *Е. Ю. Ходан.* Редактор *А. И. Логинов*
Художественный редактор *Г. М. Коровина*
Технический редактор *С. Я. Шкляр.* Корректоры: *Е. Ю. Рычагова, Л. С. Сомова*

ИБ № 32363

Сдано в набор 09.12.88. Подписано к печати 04.07.90. Формат 60×90¹/₁₆. Бумага тип. Гарнитура обыкновенная. Печать высокая. Усл. печ. л. 30,5. Усл. кр.-отт. 30,5. изд. л. 30,94. Тираж 16 700 экз. Заказ № 511. Цена 2 р. 50 к.

Издательско-производственное и книготорговое объединение «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

Четвертая типография издательства «Наука»
630077 Новосибирск 77, Станиславского, 25

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
117071 Москва В-71, Ленинский проспект, 15

ГОТОВИТСЯ К ПЕЧАТИ
в 1990 году
(ТЕМПЛАН 1990 г., ПОЗИЦИЯ 33)

Темам Р. **Математические задачи теории пластичности:** Пер. с фр.— М.: Наука. Гл. ред. физ.-мат. лит.

Исследуется математическая постановка для задач равновесия пластичного тела. Эти модели приводят к интересным математическим задачам нелинейной механики и вариационного исчисления.

Решение этих задач связано с тонкими теоремами вложения для некоторых функциональных пространств, играющих существенную роль в изучаемых автором вариационных постановках теории пластичности, и с приведенными в книге изящными результатами математического анализа в этих функциональных пространствах. В частности, тщательно изучается различие между задачами о деформациях и о напряжениях для рассматриваемого пластичного тела и при некоторых условиях доказывается существование решений этих вариационных задач.

Для математиков, механиков, инженеров, а также для студентов старших курсов соответствующих специальностей.