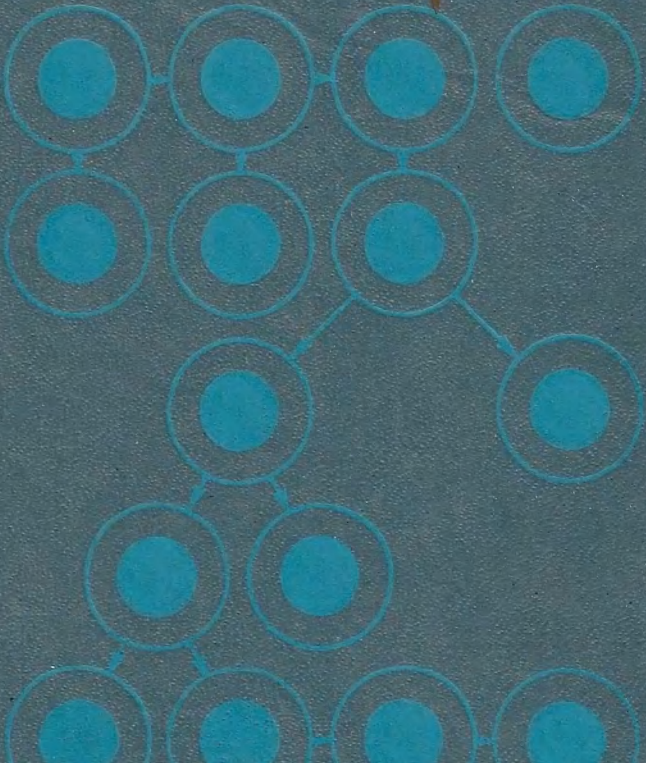
Ю. П. ЗАЙЧЕНКО

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ



Ю. П. ЗАЙЧЕ

ЛЕДОВАН
ОПЕРАЦ

ИЗДАНИЕ
ПЕРЕРАБОТАННОЕ И ДОПО

Допущено Министерс

ББК 22.18я73
517.8
З-17

УДК 517.8(07)

Зайченко Ю. П. Исследование операций: Учеб. пособие для студентов вузов. — 2-е изд., перераб. и доп.— Киев: Вища школа. Головное изд-во, 1979. 392 с. —30502. 1504000000.

В настоящем учебном пособии излагаются основы исследования операций — науки, которая ставит своей целью оптимизацию решений, принимаемых человеком в системах организационного управления и в повседневной практической деятельности. В книге рассматриваются основные принципы операционного исследования, обсуждаются особенности принятия решений в условиях определенности, риска и неопределенности. Большое внимание уделено изложению современного математического аппарата исследования операций — методам линейного, нелинейного, дискретного и динамического программирования. По сравнению с предыдущим настоящее издание дополнено изложением современных методов решения задач большой размерности, задач на транспортных сетях, элементов геометрического программирования; новых методов решения задач целочисленного программирования. Рассматриваются вопросы разработки моделей и применение имитационного моделирования в задачах исследования операций. Описание теоретических методов иллюстрируется многочисленными примерами.

Учебное пособие рассчитано на студентов специальностей «АСУ» и «Прикладная математика» технических вузов и университетов, а также на специалистов, которые занимаются приложениями исследования операций при разработке и внедрении АСУ.

Табл. 103. Ил. 86. Список лит.: 60 назв.

Редакция литературы по кибернетике, электронике и энергетике
Зав. редакцией *А. В. Дьячков*

З $\frac{30502-286}{M211(04)-79}$ 199-79. 1504000000.

ПРЕДИСЛОВИЕ

XXV съезд КПСС в качестве одной из актуальных задач десятой пятилетки поставил задачу обеспечения дальнейшего развития и повышения эффективности автоматизированных систем управления и вычислительных центров.

В свете поставленных задач важное значение для подготовки специалистов по автоматизации управления приобретают теоретические дисциплины, в которых изучаются научные основы управления производством и разрабатываются методы повышения эффективности управления экономическими системами.

Особое место среди этих дисциплин занимает исследование операций. Возникшая в начале 40-х годов XX в. в области задач управления боевыми операциями эта новая научная дисциплина быстро расширила свои исследования, охватив многие сферы целенаправленной человеческой деятельности, внесла в традиционные задачи управления производством новую методологию, известную под названием «системный подход».

Следует особо подчеркнуть связь между задачами повышения эффективности внедряемых АСУ и методологией исследования операций. Целью исследования операций являются изучение и системный анализ систем организационного типа (организаций), отыскание в них оптимизационных задач управления, постановка и внедрение которых могут оправдать затраты на создание АСУ. Важными этапами исследования операций являются: постановка оптимизационных задач, разработка математических моделей (формализация), нахождение оптимального решения и реализация его на практике. При этом реализация полученных решений рассматривается как самостоятельная операционная задача. Итак, исследование операций в отличие от чисто теоретических дисциплин имеет явно выраженную прикладную направленность.

В настоящее время курс «Исследование операций» читается во многих вузах нашей страны и за рубежом для специалистов по АСУ, организации и управлению производством и стал одним из фундаментальных курсов будущих инженеров-системотехников.

Настоящее учебное пособие подготовлено на основе лекций по курсу «Исследование операций», читаемых в Киевском политехническом институте для студентов специальности «АСУ». По сравнению с первым изданием пособия, вышедшим в 1975 году, во второе внесены существенные изменения и дополнения.

Во введении излагаются цель и задачи курса исследования операций, дается краткая характеристика особенностей этой научной дисциплины.

В первой главе рассматриваются основные этапы операционного исследования, обсуждаются содержательные постановки основных классов задач исследования операций. Рассмотрены методы и критерии принятия решений в условиях определенности, риска и неопределенности. Материал дополнен принципами разработки моделей в задачах исследования операций и имитационным моделированием систем организационного управления.

Вторая глава посвящена изложению методов и моделей линейного программирования. Описаны вопросы теории линейного программирования, рассмотрены симплекс-метод, методы нахождения допустимых базисных решений, элементы теории двойственности. Содержание главы значительно расширено: излагаются вопросы исследования моделей задач линейного программирования на чувствительность, двойственный симплекс-метод, метод обратной матрицы. Излагается метод решения ЛП-задач большой размерности, использующий идеи декомпозиции.

В новой главе «Транспортная задача линейного программирования» помимо вопросов, освещенных в первом издании (метод потенциалов, венгерский метод), рассмотрены новые вопросы, в частности, элементы теории транспортных сетей, задача о нахождении максимального потока на сети, метод решения транспортной задачи в сетевой постановке.

Завершается глава изложением декомпозиционного метода решения транспортных задач большой размерности.

Четвертая глава посвящена изложению методов и моделей дискретного программирования. Описаны метод отсекающих плоскостей Гомори, универсальный метод ветвей и границ. Применение метода ветвей и границ иллюстрируется решением задачи целочисленного программирования (ЦП) и популярной задачи о коммивояжере. Эта глава дополнена новыми параграфами, в которых рассматривается современный метод решения задач ЦП — асимптотический метод Гомори.

Заново написана глава «Нелинейное программирование». Описан классический метод условной оптимизации, метод множителей Лагранжа, методы решения задач выпуклого программирования на основе применения теоремы Куна-Таккера. Рассмотрены элементы теории двойственности в задачах оптимизации.

Описаны модели и методы решения специальных классов задач нелинейного программирования: квадратичного программирования и геометрического программирования.

Важное значение для решения разнообразных прикладных задач исследования операций, в частности задач управления запасами, распределения ресурсов, календарного планирования, сетевого планирования и управления, имеет метод динамического программирования (глава 6). На конкретном примере разбирается сущность вычислительного метода динамического программирования, формулируется принцип оптимальности Беллмана, отмечаются особенности применения метода динамического программирования при непрерывных переменных. Рассматриваются динамические задачи последовательного принятия решений и оптимального управления на основе динамического программирования. Рассмотрены многомерные задачи динамического программирования, отмечаются главные трудности решения таких задач, описан метод снижения их размерности. Описаны различные модели задач управления запасами и методы их решения, в частности динамические задачи управления запасами, для решения которых используется аппарат динамического программирования.

Завершается книга краткой характеристикой некоторых перспективных направлений и проблем исследования операций.

Для понимания материала пособия достаточно знания обычного втузовского курса высшей математики, а также основ линейной алгебры. Вспомогательный материал, используемый при изложении элементов теории линейного и нелинейного программирования, вынесен в приложения. Для лучшего усвоения материала и приобретения практических навыков в применении теоретических методов при решении задач изложение методов иллюстрируется решением примеров с подробным описанием хода решения.

Все примеры доведены до числовых ответов.

В книге приведен подробный перечень литературы по исследованию операций и смежным вопросам, снабженный аннотированным указателем.

Пособие рассчитано на студентов вузов специальности «АСУ» и «Прикладная математика». Будет также полезно аспирантам, инженерам и научным работникам, работающим в области автоматизации управления, которые желают самостоятельно пополнить свои знания в области теории оптимальных решений и ее приложений современными методами оптимизации.

Автор благодарен проф. В. И. Костюку, который был инициатором написания настоящего учебного пособия, просматривал рукопись в ходе ее написания и дал ряд полезных советов, а также выражает свою признательность рецензенту проф. А. А. Волкову за критические замечания, которые способствовали улучшению содержания книги.

Все отзывы и критические замечания относительно содержания учебного пособия просим направлять по адресу:

252054, Киев-54, Гоголевская, 7, Головное издательство издательского объединения «Вища школа».

ВВЕДЕНИЕ

ОБЩАЯ ХАРАКТЕРИСТИКА ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

Научно-техническая революция вызвала появление нового объекта исследований в области управления, получившего название большие, или сложные системы. Поскольку общая теория управления большими системами пока еще находится в стадии становления, любая попытка добиться практических результатов в управлении подобной системой вынуждает ограничивать области исследований конкретным классом систем.

Одним из самых значительных классов «больших систем» являются системы организационного управления. К ним относятся промышленные предприятия, производственные объединения, отрасли, экономика целого государства, а также глобальные системы (макросистемы), например группа государств Совета экономической взаимопомощи (СЭВ).

Исследование операций представляет собой комплекс научных методов для решения задач эффективного управления организационными системами.

Корни исследования операций уходят в далекую историю. Резкое увеличение размеров производства, разделение труда в сфере производства обусловило постепенную дифференциацию и управленческого труда. Появилась необходимость в планировании материальных, трудовых и денежных ресурсов, в учете и анализе результатов труда и выработке прогноза на будущее. В управленческом аппарате начали выделяться подразделения: отдел финансов, сбыта, бухгалтерии и планово-экономический отдел и др., принявшие на себя отдельные управленческие функции.

К этому периоду относятся первые работы по исследованию в области организации труда и управления — предвестники будущей науки.

Как самостоятельное научное направление исследование операций оформилось в начале 40-х годов. Первые публикации по исследованию операций относятся к 1939—1940 гг., в которых методы исследования операций применены для решения военных задач, в частности для анализа и исследования боевых операций. Отсюда и возникло название дисциплины.

Позднее принципы и методы исследования операций стали применяться в сфере промышленно-финансового управления. С увеличением масштабов производства, развитием и совершенствованием форм и методов организации управления экономическими системами расширялись масштабы операционных исследований, круг решаемых задач, совершенствовались методы новой науки.

Возникла необходимость в подготовке кадров специалистов по исследованию операций — операционистов. В ведущих университетах США и Англии впервые было начато систематическое преподавание курса исследования операций.

Возникла необходимость в координации работы многотысячной армии операционистов, в регулярном обмене теоретическими исследованиями и прикладными разработками.

С этой целью в 1957 г. была создана Международная федерация обществ исследования операций — IFORS, в состав которой вошли национальные общества и комитеты по исследованию операций многих стран. Активным участником Международной федерации является Национальный комитет Советского Союза.

Большой вклад в формирование и развитие новой науки внесли зарубежные ученые Р. Акоф, Р. Беллман, Г. Данциг, Г. Кун, Т. Саати, Р. Черчмен (США), А. Кофман, Р. Фор (Франция) и др.

Важная роль в создании современного математического аппарата и развитии многих направлений исследования операций принадлежит советским ученым Л. В. Канторовичу, Б. В. Гнеденко, Н. П. Бусленко, В. С. Михалевичу, Н. Н. Моисееву, Д. Б. Юдину, Ю. М. Ермольеву и др. Выдающийся вклад в создание аппарата линейного программирования и его применение для решения практических задач экономики внес акад. Л. В. Канторович. В его работе «Математические методы организации и планирования производства» (1939 г.) были заложены основы линейного программирования.

Важнейшее значение имело введение Л. В. Канторовичем понятия объективно-обусловленных оценок. Они составляют решение задачи, двойственной к задаче расчета оптимального плана. С помощью этих оценок можно в ряде случаев, учитывая изменения обстановки, варьирования ограничений на ресурсы, оперативно корректировать рассчитанный оптимальный план. Объективно-обусловленные оценки, характеризуя эффективность использования ограниченных производственных ресурсов, имеют большое значение при анализе проблем формирования системы цен и других экономических показателей. За разработку методов линейного программирования и экономико-математического моделирования акад. Л. В. Канторовичу совместно с акад. В. С. Немчиновым и проф. В. В. Новожиловым была присуждена Ленинская премия в 1965 году.

В 1975 году Л. В. Канторовичу была присуждена Нобелевская премия по экономике за выдающийся вклад в разработку теории оптимального использования ресурсов в экономике. Большой вклад в разработку и использование аппарата имитационного моделирования для исследования сложных систем внес член-кор. АН СССР Н. П. Бусленко. Важная роль в разработке новых методов решения комбинаторных задач дискретной оптимизации принадлежит В. С. Михалевичу и В. В. Шкурбе. Предложенные ими методы нашли широкое применение при решении задач теории расписаний. В последние годы интенсивно развиваются методы принятия оптимальных решений в условиях неполной информации, когда существенную роль играют случайные факторы. Соответствующее направление в теории оптимизации — стохастическое программирование развивается в работах Ю. М. Ермольева, Д. Б. Юдина и др.

Наконец, разработке теории иерархических систем управления, численным методам в теории оптимального управления посвящены работы Н. Н. Моисеева.

Широко используется при оптимизации непрерывных систем, описываемых дифференциальными уравнениями, принцип максимума Л. С. Понтрягина.

Поскольку исследование операций — это новая интенсивно развивающаяся наука, вряд ли возможно дать сейчас ее исчерпывающее определение. Тем не менее для понимания предмета исследования операций, ее содержания и целей такое определение необходимо.

Исследование операций — это наука, занимающаяся разработкой и практическим применением методов наиболее эффективного (или оптимального) управления организационными системами.

Итак, предмет исследования операций — это системы организационного управления (организации), которые состоят из большого числа взаимодействующих между собой подразделений, причем интересы подразделений не всегда согласуются между собой и могут быть противоположными.

Целью исследования операций является количественное обоснование принимаемых решений по управлению организациями.

Решение, которое оказывается наиболее выгодным для всей организации, называется оптимальным, а решение, наиболее выгодное одному или нескольким подразделениям, будет субоптимальным.

В качестве примера типичной задачи организационного управления, где сталкиваются противоречивые интересы подразделений, рассмотрим задачу управления запасами предприятия.

Производственный отдел стремится выпускать как можно больше продукции при наименьших затратах. Поэтому он заинтересован в возможно более длительном и непрерывном производстве, т. е. в выпуске изделий большими партиями, ибо такое производство снижает затраты на переналадку оборудования, а следовательно и общие производственные затраты. Однако выпуск изделий большими партиями требует создания больших объёмов запасов материалов, комплектующих изделий и т. д.

Отдел сбыта также заинтересован в больших запасах готовой продукции, чтобы удовлетворить любые запросы потребителя в любой момент времени. Заключая каждый контракт, отдел сбыта, стремясь продать как можно больше продукции, должен предлагать потребителю максимально широкую номенклатуру изделий. Вследствие этого между производственным отделом и отделом сбыта часто возникает конфликт по поводу номенклатуры изделий. При этом отдел сбыта настаивает на включении в план многих изделий, выпускаемых в небольших количествах даже тогда, когда они не приносят большой прибыли, а производственный отдел требует исключения таких изделий из номенклатуры продукции.

Финансовый отдел, стремясь минимизировать объем капитала, необходимого для функционирования предприятия, пытается уменьшить количество «связанных» оборотных средств. Поэтому он заинтересован в уменьшении запасов до минимума. Как видим, требования к размерам запасов у разных подразделений организации оказываются различными. Возникает вопрос, какая стратегия в отношении запасов будет наиболее благоприятной для всей организации. Это типичная задача организационного управления. Она связана с проблемой оптимизации функционирования системы в целом и затрагивает противоречивые интересы ее подразделений.

Теперь рассмотрим основные особенности исследования операций.

1. Характерной особенностью исследования операций является так называемый *системный подход* к анализу поставленной проблемы. Системный подход, или системный анализ, является основным методологическим принципом исследования операций, который состоит в следующем. Любая задача, какой бы частной она не казалась на первый взгляд, рассматривается с точки зрения ее влияния на критерий функционирования всей системы. Выше системный подход был проиллюстрирован на примере задачи управления запасами.

Глубокое изложение основ системного анализа имеется в [37].

2. Для исследования операций характерно, что при решении каждой проблемы возникают все новые и новые задачи. Поэтому если сначала ставятся узкие, ограниченные цели, применение операционных методов не эффективно. Наибольший эффект может быть достигнут только при непрерывном исследовании, обеспечивающем преемственность в переходе от одной задачи к другой.

3. Одной из существенных особенностей исследования операций является стремление найти *оптимальное решение поставленной задачи*. Однако часто такое решение оказывается недостижимым из-за ограничений, накладываемых имеющимися в наличии ресурсами (денежные средства, машинное время) или уровнем современной науки. Например, для многих комбинаторных задач, в частности задач календарного планирования при числе станков $n > 4$, оптимальное решение при современном развитии математики оказывается возможным найти лишь простым перебором вариантов. Однако даже при небольших n число возможных вариантов оказывается настолько велико, что перебор всех вариантов при существующих ограничениях на быстродействие ЭВМ и допустимое машинное время практически не мыслим. Тогда приходится ограничиваться поиском «достаточно хорошего», или *субоптимального решения*. Поэтому исследование операций один из его создателей — Т. Саати — определил как «*искусство давать плохие ответы на те практические вопросы, на которые даются еще худшие ответы другими методами*» [1, 10].

4. Особенность операционных исследований состоит в том, что они проводятся *комплексно*, по многим направлениям. Для проведения такого исследования создается *операционная группа*. В ее состав входят специалисты разных областей знания: инженеры, математики, экономисты, социологи, психологи. Задачей создания подобных операционных групп является комплексное исследование всего множества факторов, влияющих на решение проблемы, и использование идей и методов различных наук.

Глава 1. ОСНОВНЫЕ ПРИНЦИПЫ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

§ 1. ОСНОВНЫЕ ЭТАПЫ ОПЕРАЦИОННОГО ИССЛЕДОВАНИЯ

При всем многообразии содержания конкретных работ по исследованию операций каждое операционное исследование проходит последовательно следующие основные этапы: 1) постановка задачи, 2) построение математической модели, 3) нахождение решения, 4) проверка и корректировка модели, 5) реализация найденного решения на практике.

Постановка задачи — это чрезвычайно ответственный этап операционного исследования. Первоначально задачу формулируют с точки зрения заказчика. Такая постановка задачи обычно не бывает окончательной. Во время анализа исследуемой системы задача постепенно уточняется. На этом этапе роль операционной группы состоит в проведении тщательного обследования объекта, изучении множества факторов, влияющих на результаты исследуемого процесса.

После сбора данных обследования и их анализа операционная группа выделяет совокупность существенных факторов, проводит консультации с заказчиками и уточняет окончательно содержательную (словесную) постановку задачи.

Для выяснения упущенных факторов и их взаимосвязей при необходимости проводят дополнительное обследование объекта.

Построение математической модели. Получив достаточно строгую *и логически непротиворечивую*, содержательную постановку задачи, нужно построить ее математическую модель. Этот процесс называется формализацией задачи.

В самом общем случае математическая модель задачи имеет вид: найти

$$\max E = f(\mathbf{x}, \mathbf{y}) \tag{1.1}$$

при ограничениях

$$g_i(\mathbf{x}, \mathbf{y}) \leqslant b_i, \quad i = \overline{1, m}, \tag{1.2}$$

где $E = f(\mathbf{x}, \mathbf{y})$ — целевая функция (показатель качества или эффективность) системы; $\mathbf{x}$ — вектор управляемых переменных; $\mathbf{y}$ — вектор неуправляемых переменных; g_i — функция потребления i-го ресурса; b_i — величина i-го ресурса (например, плановый фонд машинного времени группы токарных автоматов в станко-часах).

Нахождение решения. Для нахождения оптимального решения задачи (1.1) — (1.2) в зависимости от вида и структуры целевой функции и ограничений используют те или иные методы теории оптимальных решений (методы математического программирования)[1].

---

[1] Термин «математическое программирование» здесь неточен, его употребляют с учетом исторических причин и традиций.

1. *Линейное программирование*, если $f(\mathbf{x}, \mathbf{y})$, $g_i(\mathbf{x}, \mathbf{y})_{i=\overline{1,m}}$ — линейны относительно переменных $\mathbf{x}$.

2. *Нелинейное программирование*, если $f(\mathbf{x}, \mathbf{y})$ или $g_i(\mathbf{x}, \mathbf{y})$ — нелинейны относительно переменных $\mathbf{x}$.

3. *Динамическое программирование*, если целевая функция $f(\mathbf{x}, \mathbf{y})$ имеет специальную структуру, являясь *аддитивной* или *мультипликативной* функцией от переменных $\mathbf{x}$.

Укажем, что $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$ — аддитивная функция, если $f(x_1, x_2, \ldots, x_n) = \sum\limits_i f_i(x_i)$, и функция $f(x_1, x_2, \ldots, x_n)$ — мультипликативная функция, если $f(x_1, x_2, \ldots, x_n) = \prod\limits_i f_i(x_i)$.

4. *Геометрическое программирование*, если целевая функция $f(\mathbf{x})$ и ограничения $g_k(\mathbf{x})$ представляют собой так называемые функции-позиномы $g_k(\mathbf{x}) = \sum\limits_i c_i x_1^{\alpha_{i_1}} x_2^{\alpha_{i_2}} \ldots x_n^{\alpha_{i_n}}$.

Математическая модель задачи в этом случае записывается в виде

$$\min \sum_{i \in I[0]} c_i x_1^{\alpha_{i_1}} x_2^{\alpha_{i_2}} \ldots x_n^{\alpha_{i_n}} \tag{1.3}$$

при условиях

$$g_k(\mathbf{x}) = \sum_{i \in I[k]} c_i x_1^{\alpha_{i_1}} x_2^{\alpha_{i_2}} \ldots x_n^{\alpha_{i_n}} \leqslant 1, \tag{1.4}$$

$$k = \overline{1, p}, \tag{1.5}$$

где

$$I[0] = (m_0, m_0 + 1, \ldots, n_0); \quad I[k] = [m_k, m_k + 1, \ldots, n_k];$$
$$m_{k+1} = n_k + 1; \quad m_0 = 1; \quad n_p = n. \tag{1.6}$$

5. *Стохастическое программирование*, когда вектор неуправляемых переменных $\mathbf{y}$ случаен.

В этом случае математическая модель задачи (1.1—1.2) будет иметь вид:

$$\max \mathrm{M}_y E = M_y \{f(\mathbf{x}, \mathbf{y})\}, \tag{1.7}$$

при ограничениях

$$\mathrm{M}_y \{g_i(\mathbf{x}, \mathbf{y})\} \leqslant b_i, \quad i = \overline{1, m}, \tag{1.8}$$

или вероятностных ограничениях

$$P\{g_i(\mathbf{x}, \mathbf{y}) \leqslant b_i\} \geqslant 1 - \varepsilon_i; \quad i = \overline{1, m}, \tag{1.9}$$

где M_y — математическое ожидание по y; $P\{g_i(\mathbf{x}) \leqslant b\}$ — вероятность того, что выполняется условие $g_i(\mathbf{x}) \leqslant b$.

6. *Дискретное программирование*, если на переменные x_j наложено условие дискретности (например целочисленности): x_j — целое, $j = \overline{1, n_1} \leqslant n$.

7. *Эвристическое программирование* применяют для решения тех задач, в которых точный оптимум найти алгоритмическим путем невозможно из-за огромного числа вариантов. В таком случае отказываются от поиска оптимального решения и отыскивают достаточно хорошее (или удовлетворительное с точки зрения практики) решение. При этом пользуются специальными приемами — эвристиками, поз-

воляющими существенно сократить число просматриваемых вариантов. Эвристические методы также применяют, когда оптимальное решение в принципе может быть найдено (т. е. задача алгоритмически разрешима), однако для этого требуются объемы ресурсов, значительно превышающие наличные.

Из перечисленных выше методов математического программирования наиболее развитым и законченным является линейное программирование. В его рамки укладывается широкий круг задач исследования операций.

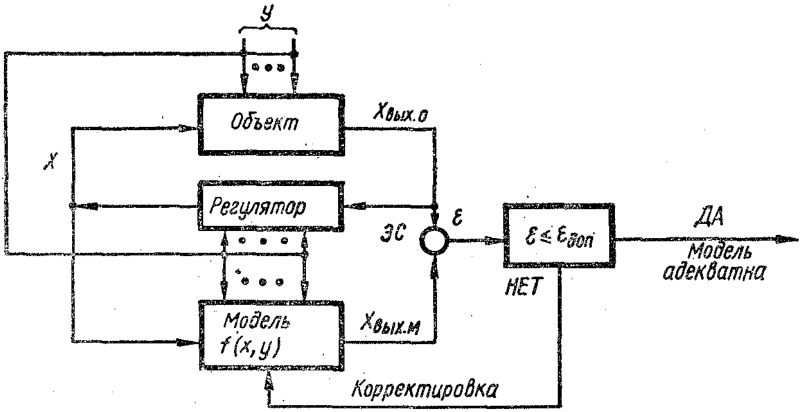


Рис.1.1.

Проверка и корректировка модели. В сложных системах, к которым относятся системы организационного типа, модель лишь частично отражает реальный процесс. Поэтому необходима проверка степени соответствия или адекватности модели и реального процесса. Проверку производят сравнением предсказанного поведения с фактическим при изменении значений внешних неуправляемых воздействий.

Проверку и корректировку модели можно производить, например, по логической схеме, показанной на рис. 1.1, где **X** — вектор управляемых переменных, **Y** — вектор неуправляемых переменных, $X_{\text{вых.о}}$ — выходные параметры объекта, $X_{\text{вых.м}}$ — выходные параметры модели.

Величины $X_{\text{вых.о}}$ и $X_{\text{вых.м}}$ подаются на элемент сравнения ЭС, и вычисляется величина ошибки $\varepsilon(y) = | X_{\text{вых.о}} - X_{\text{вых.м}} |$. Если величина $\varepsilon(y)$ превышает допустимое значение отклонения $\varepsilon_{\text{доп}}$ (его выбирают, исходя из требуемой степени адекватности модели), то это свидетельствует о том, что упущены некоторые важные факторы и взаимосвязи. В этом случае производят корректировку модели.

Корректировка может потребовать дополнительных исследований объекта, уточнения структуры математической модели, многочисленных изменений переменных модели. Таким образом, четыре названные выше этапа повторяют многократно до тех пор, пока не будет достигнуто удовлетворительное соответствие между выходами объекта и модели.

Реализация найденного решения на практике является важнейшим этапом, завершающим операционное исследование. Внедрение можно рассматривать как самостоятельную задачу, применив к ней системный подход и анализ. Полученное предварительно математическое решение облекают в соответствующую содержательную форму и представляют заказчику в виде инструкций и рекомендаций.

С точки зрения реализации оптимального решения на практике исследование операций занимает особое место в проблематике АСУ. Известно, что внедрение АСУ эффективно для решения таких задач управления, которые невозможно решить при сложившейся ранее практике управления. Академиком В. М. Глушковым был выдвинут так называемый *принцип новых задач АСУ*. Под этим принципом понимается поиск и постановка на производстве действительно новых задач оптимального управления, которые могут окупить затраты на создание АСУ. Исследование операций и является методологической основой для нахождения таких задач, разработки их математических моделей и алгоритмов решения, а также для практического внедрения найденных оптимальных решений.

На производстве выполняют операционное исследование. На основе материалов исследования проводят системный анализ объекта и определяют задачи управления, которые дают наиболее ощутимый эффект в результате автоматизации. Исходя из принципа системного подхода разрабатывают математические модели этих задач и алгоритмы решения. Практическая же реализация указанных задач, т. е. их внедрение, будет осуществлена при создании соответствующих АСУ.

§ 2. ТИПИЧНЫЕ КЛАССЫ ЗАДАЧ

Накопленный опыт в решении практических задач исследования операций и его систематизация позволяют выделить по содержательной постановке следующие типичные классы задач [1]: 1) управления запасами, 2) распределения ресурсов, 3) ремонта и замены оборудования, 4) массового обслуживания, 5) упорядочения, 6) сетевого планирования и управления, 7) выбора маршрута, 8) комбинированные.

Рассмотрим краткие особенности каждого класса задач.

Задачи управления запасами составляют самый распространенный и изученный в настоящее время класс задач исследования операций. Они обладают следующей особенностью. С увеличением запасов увеличиваются расходы на их хранение, но уменьшаются потери из-за возможной их нехватки. Следовательно, одна из задач управления запасами заключается в определении такого уровня запасов, который минимизирует следующий критерий: сумму ожидаемых затрат по хранению запасов, а также потерь из-за их дефицита.

В зависимости от условий задачи управления запасами делятся на следующие три группы:

1. Моменты поставок или оформления заказов на пополнение запасов фиксированы. Определить объемы производимых или закупаемых партий запасов.

2. Объемы производимых или закупаемых партий запасов фиксированы. Определить моменты оформления заказов на поставки.

3. Моменты оформления заказов и объемы производимых или закупаемых партий не фиксированы. Определить эти величины, исходя из сформулированного выше критерия.

Задачи распределения ресурсов возникают, когда существует определенный набор работ (операций), которые необходимо выполнять, а наличных ресурсов для выполнения каждой работы наилучшим образом не хватает.

В зависимости от условий задачи распределения ресурсов делятся на три группы:

1. Заданы и работы, и ресурсы. Распределить ресурсы между работами таким образом, чтобы максимизировать некоторую меру эффективности (прибыль) или минимизировать ожидаемые затраты (издержки производства).

Пример 1.1. Известны производственное задание и производственные мощности предприятия. При существующих различных способах получения изделий (например, обработка на разных станках) ограничение по мощности не позволяет для каждого изделия использовать наилучшую технологию.

Какие способы производства надо выбрать для каждого вида изделий, чтобы выполнить задание с минимальными затратами?

2. Заданы только наличные ресурсы. Определить, какой состав работ можно выполнить с учетом этих ресурсов, чтобы обеспечить максимум некоторой меры эффективности.

Пример 1.2. Задано предприятие с определенными производственными мощностями. Какую продукцию следует производить, чтобы получить максимальный доход?

3. Заданы только работы. Определить, какие ресурсы необходимы для того, чтобы минимизировать суммарные издержки производства.

Пример 1.3. Известно месячное расписание движения пассажирских самолетов по авиалиниям. Какое количество экипажей необходимо подобрать, чтобы выполнить план перевозок с минимальными эксплуатационными затратами?

Задачи ремонта и замены оборудования появляются в тех случаях, когда работающее оборудование изнашивается, устаревает и со временем подлежит замене.

Изношенное оборудование подвергают либо предупредительно-восстановительному ремонту, улучшающему его технологические характеристики, либо полной замене. При этом возможная постановка задачи такова.

Определить такие сроки восстановительного ремонта и момент замены оборудования модернизированным, при которых минимизируется сумма затрат на ремонт и замену оборудования при его старении за все время эксплуатации.

Существует и такое оборудование, в котором детали полностью выходят из строя, не восстанавливаются и подлежат замене (например, перегоревшие электронные лампы). Постановка задачи в этом случае следующая.

Определить такие сроки профилактического контроля по обнаружению неисправностей, при которых минимизируется сумма затрат на проведение контроля и ожидаемых потерь от простоя оборудования, вследствие выхода из строя деталей в интервале между соседними проверками (контролем).

Задачи массового обслуживания посвящены изучению и анализу систем обслуживания с очередями заявок, или требований. С явлением образования очередей приходится сталкиваться как в производственной практике, так и в быту. Типичными примерами являются очереди самолетов, ожидающих взлета или посадки, очереди клиентов в ателье бытового обслуживания, абонентов, ожидающих вызова на междугородной АТС, и т. д.

Очереди возникают из-за того, что поток требований или клиентов на обслуживание неуправляем и случаен. Если количество приборов обслуживания (взлетно-посадочные полосы аэродрома, приемщики в ателье бытового обслуживания, каналы связи и т. д.) достаточно велико, то очередь образуется редко, однако неизбежны длительные простои оборудования. С другой стороны, при малом количестве приборов создается значительная очередь и будут большие потери из-за ожидания в очереди. Поэтому возможна следующая постановка задач массового обслуживания. Определить, какое количество приборов обслуживания необходимо, чтобы минимизировать суммарные ожидаемые потери от несвоевременного обслуживания и простоев оборудования.

Задачи упорядочения характеризуются следующими особенностями. Например, имеется множество различных деталей с определенными технологическими маршрутами, а также несколько единиц оборудования (фрезерный, токарный и строгальный станки), на которых эти детали обрабатываются. Так как одновременно обрабатывать более одной детали на одном станке невозможно, у некоторых из станков может образоваться очередь работ, т. е. деталей, ждущих обработки. Время обработки каждой детали известно. Определить такую очередность обработки деталей на каждом станке, при которой минимизируется некоторый критерий оптимальности, например, суммарная продолжительность завершения комплекса работ. Такая задача называется *задачей календарного планирования* или составления расписания, а выбор очередности запуска деталей в обработку — *упорядочением*.

Критерии оптимальности, используемые в задачах календарного планирования, имеют различный вид. Наиболее часто встречаются следующие критерии:

1. Минимизация общей продолжительности работ, т. е. интервала времени между моментом начала первой операции и моментом окончания последней.

2. Минимизация общего запаздывания. Запаздывание определяется как разность между фактическим и директивным сроком завершения обработки по каждой детали. Общее запаздывание представляет собой сумму запаздываний по всем деталям.

3. Минимизация максимального запаздывания, т. е. минимизация запаздывания по той детали, для которой эта величина является наибольшей.

4. Минимизация потерь, обусловленных запаздыванием.

Задачи сетевого планирования и управления (СПУ). В этом классе задач рассматривается соотношение между сроком окончания крупного комплекса операций и моментами начала всех операций комплекса. Они актуальны при разработке сложных и дорогостоящих проектов.

Для строгой постановки задач необходимы такие условия:

должно существовать точно определяемое множество операций, которые надо выполнить для завершения всего комплекса, включающего эти операции в качестве своих элементов;

множество операций комплекса упорядочено так, что для каждой операции известно, какие операции непосредственно ей предшествуют, а какие следуют за ней;

в пределах заданного отношения упорядочения операции можно начинать и заканчивать независимо друг от друга;

известна взаимосвязь между величиной потребляемого ресурса и длительностью каждой операции.

Комплекс операций в этом случае можно представить в виде сетевого графика, состоящего из вершин (узлов) и ориентированных дуг. В этом случае операции изображают дугами, а вершины представляют собой некоторые события. Дуги, входящие в вершину, отвечают операциям, которые должны быть закончены раньше, чем можно будет начать операции, изображенные исходящими дугами. Событию, соответствующему началу выполнения комплекса, присваивают номер 0. Остальные события нумеруют так, что, если события i, j связаны некоторой операцией (i, j), то выполняется неравенство $t_{\text{н}}(j) \geqslant t_{\text{н}}(i) + t_{ij}$, где $t_{\text{н}}(i)$, $t_{\text{н}}(j)$ — моменты наступления событий i, j и t_{ij} — длительность операции (i, j). Если выполняются вышеизложенные условия и допущения, возможны следующие постановки задач сетевого планирования и управления.

1. Задана продолжительность всего комплекса. Определить сроки начала каждой операции, при которых минимизируется один из следующих критериев: *а*) общие затраты на выполнение всего комплекса работ; *б*) среднеквадратичный показатель неравномерности потребляемых ресурсов; *в*) вероятность невыполнения комплекса работ в директивный срок; *г*) среднеквадратичное отклонение требуемых ресурсов от наличных.

2. Заданы общие ресурсы. Определить сроки начала каждой операции, при которых минимизируется продолжительность выполнения всего комплекса работ. Методы решения задач СПУ изложены в [1].

Задачи выбора маршрута, или сетевые задачи чаще всего встречаются при исследовании разнообразных процессов на транспорте и в системах связи. Типичной задачей является задача нахождения некоторого маршрута проезда из города A в город B при наличии нескольких маршрутов для разных промежуточных пунктов. Стоимость проезда и затрачиваемое на проезд время зависят от выбранного маршрута. Определить наиболее экономичный маршрут по выбранному критерию оптимальности.

На допустимые маршруты может быть наложен ряд ограничений. Так, вводят запрет на возврат к уже пройденному пункту (узлу сети) или требование обхода всех пунктов сети с условием, что в каждом пункте можно побывать только один раз (задача коммивояжера).

В пунктах сети возможны задержки (например, на перекрестках шоссейных дорог). Эти задержки зависят от нагрузки на узел, занятости исходящих коммуникаций, ограниченной пропускной способности пунктов сети или носят случайный характер.

Среди сетевых задач наиболее распространенными являются задача выбора кратчайшего пути между произвольными пунктами сети, задача коммивояжера, задача о максимальном потоке [1].

Комбинированные задачи включают в себя несколько типовых моделей задач одновременно. Например, при планировании и управлении производством приходится решать следующий комплекс задач:

1) сколько изделий каждого типа необходимо выпустить и каковы оптимальные размеры партий изделий? (Типичная задача планирования производства);

2) распределить производственные заказы по видам оборудования после того как определен оптимальный план производства. (Типичная задача распределения);

3) в какой последовательности и когда следует выполнять производственные заказы? (Типичная задача календарного планирования).

Так как эти три задачи нельзя решить изолированно, независимо друг от друга, то возможен следующий подход к решению данной комбинированной задачи. Сначала получают оптимальное решение задачи планирования производства. Затем в зависимости от этого оптимума находят наилучшее распределение оборудования. Наконец, на основе такого распределения составляют оптимальный график выполнения работ.

Однако такая последовательная оптимизация частных подзадач не всегда приводит к оптимальному решению задачи в целом. В частности, например, может оказаться, что нельзя произвести все изделия в оптимальных количествах из-за ограниченности имеющихся ресурсов. Пока еще не найден метод, позволяющий получить одновременный оптимум для всех трех задач, а возможно он не существует для конкретных задач. Поэтому для решения подобных комбинированных задач применяется метод последовательных приближений, позволяющий приблизиться к искомому решению комбинированной задачи достаточно близко [1, 49].

Предложенная классификация задач исследования операций не является окончательной. Со временем некоторые классы задач объединяются и становится возможным их совместное решение, стираются границы между указанными классами задач, а также появляются новые классы задач.

Следует также отметить, что ряд задач исследования операций не укладывается ни в один из известных классов и представляет наибольший интерес с научной точки зрения.

§ 3. НЕКОТОРЫЕ ПРИНЦИПЫ ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

Теория принятия решений является фундаментом науки исследования операций. В процессе принятия решений возникают такие трудности:

1. Большое число критериев, которые не всегда согласованы между собой. Например, при проектировании нового устройства часто выдвигается требование максимальной надежности и минимальной стоимости изделия. Эти критерии являются противоречивыми, поэтому возникает задача компромисса между ними;

2. Высокая степень неопределенности, которая обусловлена недостаточной информацией для обоснованного принятия решений.

Элементы процесса принятия решений и классификация задач

Любой процесс принятия решений включает следующие элементы:

1. Цель. Необходимость принятия решений определяется целью или несколькими целями, которые должны быть достигнуты.
2. Лицо, принимающее решение, должно нести ответственность за последствия этих решений.
3. Альтернативные решения (различные варианты достижения целей).
4. Внешняя среда (совокупность всех внешних факторов, влияющих на исход решения).
5. Исходы решений.
6. Правила выбора решений (решающие правила).

Эти правила позволяют определить наиболее предпочтительное в смысле выбранного критерия решение.

Решающее правило отражает информированность лица, принимающего решение, о возможных исходах выбранных решений, а также предпочтительность тех или иных исходов. Итак, основой для построения решающих правил служит информация о предпочтении различных альтернатив для лица, принимающего эти решения.

Теория принятия решений использует различные процедуры, позволяющие формализовать предпочтения, т. е. выразить их в единой количественной мере. Основой для таких процедур является *теория полезности*, разработанная Дж. фон Нейманом и О. Моргенштерном [35]. Ее математическая основа — система аксиом, в которых утверждается, что существует некоторая мера ценности, позволяющая упорядочить результаты решений. Эта мера называется *функцией полезности решений* или *полезностью* [43].

В зависимости от условий внешней среды и степени информированности лица существует следующая классификация задач принятия решений: а) в условиях определенности, б) в условиях риска, в) в условиях неопределенности, г) в условиях конфликтных ситуаций или противодействия (активного противника).

Принятие решений в условиях определенности

Принятие решений в условиях определенности характеризуется однозначной или детерминированной связью между принятым решением и его исходом. Основная трудность — наличие нескольких критериев, по которым следует сравнивать исходы.

Здесь возникает задача принятия решений при так называемом «векторном критерии» [12].

Случай 1. Пусть имеется совокупность критериев:

$$F_1(x),\ F_2(x),\ \ldots,\ F_n(x)\ x \in X.$$

Найти решение, которое окажется наилучшим в смысле выбираемого критерия.

Если все критерии измеряются в одной шкале, то обобщенный критерий $F_о(x)$ можно записать в виде взвешенной суммы этих критериев

$$F_о(x) = \sum_{i=1}^{n} w_i F_i(x), \tag{3.1}$$

где w_i — вес соответствующего критерия.

В этом случае необходимо найти $\max\limits_x F_о(x)$.

Если же эти критерии измеряются в различных шкалах, то необходимо привести их к одной шкале. Для этого формируют критерий

$$\min_x F_о(x) = \min_x \sum_{i=1}^{n} w_i \frac{F_i(x_i^*) - F_i(x)}{|F_i(x_i^*)|}, \tag{3.2}$$

где

$$F_i(x_i^*) = \max_{x_i} F_i(x).$$

Следовательно, требуется свести к минимуму величину уклонения каждого критерия от его максимального значения.

При таком формировании обобщенного критерия возникает некоторое несоответствие, связанное с тем, что можно добиться высоких показателей по одним критериям за счет ухудшения показателей по другим. В этом случае значения некоторых частных критериев могут оказаться меньше предельно допустимых значений $F_i(x) < F_{i\text{доп}}$.

Однако часто необходимо, чтобы выполнялось следующее условие:

$$F_i(x) \geqslant F_{i\text{доп}}. \tag{3.3}$$

Поэтому можно предложить еще один способ образования обобщенного критерия.

Допустим, что по каждому критерию определены предельные значения $F_{i\,\text{доп}}$, $i = \overline{1,\ n}$.

Если условие (3.3) выполняется, то можно принять $F_i(x)$ равным его собственному значению.

Если это условие не выполняется, то нужно принять $F_i(x) = -\infty$.

В таком случае задача сводится к нахождению

$$\max_x F_о = \sum_{i=1}^{n} w_i F_i(x) \tag{3.4}$$

при условии (3.3).

Случай 2. Предположим, что критерии упорядочены в последовательности $F_1, F_2, \ldots, F_n$.

Тогда задача отыскания оптимального решения может быть записана как

$$\max_{x \in X} F_1(x) \tag{3.5}$$

при ограничениях:

$$F_2(x) \geqslant F_{2доп};$$

$$\cdots\cdots\cdots$$

$$F_n(x) \geqslant F_{nдоп}.$$

Два варианта логического объединения критериев. Предположим, что критерии F_1, F_2, ..., F_n могут принимать только два значения 0 или 1.

$F_i(x) = 1$, если i-ая цель достигнута. В противном случае $F_i(x) = 0$.

Тогда обобщенный критерий может быть записан:

а) в виде конъюнкции критериев F_i, если общая цель операции состоит в выполнении всех целей одновременно, т. е.

$$F_o = \prod_{i=1}^{n} F_i(x); \tag{3.6}$$

б) в виде дизъюнкции критериев, когда общая цель операции достигается, если достигается хотя бы одна частная цель, т. е.

$$F_o = 1 - \prod_{i=1}^{n} [1 - F_i(x)]. \tag{3.7}$$

Методика определения полезности. Для принятия решений необходимо установить предпочтительность различных критериев (меру полезности тех или иных исходов) для лица, принимающего решение.

Методика определения полезности возможных результатов разработана в [1].

Практическое применение теории полезности основывается на следующих аксиомах [43]:

1) Результат x_i оказывается предпочтительнее x_j (это записывается так: $x_i > x_j$), тогда и только тогда, когда $u(x_i) \geqslant u(x_j)$, где $u(x_i)$ и $u(x_j)$ — полезности результатов x_i и x_j соответственно.

2) *Транзитивность:* если $x_i > x_j$, а $x_j > x_k$, то $u(x_i) > u(x_k)$.

3) *Линейность:* если некоторый результат x представлен в виде $x = (1-k)x_1 + kx_2$, где $0 \leqslant k \leqslant 1$, то $u(x) = (1-k)u(x_1) + ku(x_2)$.

4) *Аддитивность:* если $u(x_1, x_2)$ — полезность от достижения одновременно результатов x_1 и x_2, то свойство аддитивности следующее: $u(x_1, x_2) = u(x_1) + u(x_2)$. Аналогично, если имеется n результатов, $x_1, x_2, ..., x_n$, достигаемых одновременно, то

$$u(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} u_i(x_i). \tag{3.8}$$

Рассмотрим несколько вариантов методики определения полезности в различных случаях.

I. Случай, когда имеются два результата.

Методика определения полезности такова:

1. Определяем, какой результат более предпочтителен для лица, принимающего решение. Пусть $x_1 > x_2$, т. е. x_1 предпочтительнее, чем x_2.

2. Затем определяем такую вероятность α, при которой достижение результата x_1 будет эквивалентно результату x_2, получаемому с вероятностью 1.

Таблица 1.1

1	x_1 или $x_2 + \cdots + x_n$	$n+1$	x_2 или $x_3 + \cdots + x_{n-1}$
2	x_1 или $x_2 + \cdots + x_{n-1}$	$n+2$	x_2 или $x_3 + \cdots + x_{n-2}$
3	x_1 или $x_2 + \cdots + x_{n-2}$	$n+3$	x_2 или $x_3 + \cdots + x_{n-3}$
...	. . . .	...	. . . .
n	x_2 или $x_3 + \cdots + x_n$	N	x_{n-2} или $x_{n-1} + x_n$

3. Оцениваем соотношение между полезностями результатов x_1 и x_2. Для этого примем полезность

$$u(x_2) = 1.$$

Тогда

$$\alpha u(x_1) = u(x_2), \quad u(x_1) = \frac{1}{\alpha}.$$

II. Случай, когда имеются n возможных результатов $x_1, x_2, \ldots, x_n$, между которыми установлено отношение предпочтения $x_1 > x_2 > x_3 > \ldots > x_n$.

Для этого случая методика определения полезности следующая:

1. Определяем величину α_1 из условия

$$\alpha_1 u(x_1) = u(x_2).$$

2. Аналогично определяем

$$\alpha_2 u(x_2) = u(x_3),$$

$$\cdots\cdots\cdots$$

$$\alpha_{n-1} u(x_{n-1}) = u(x_n).$$

3. Положив полезность наименее предпочтительного результата x_n равной единице, находим

$$u(x_n) = 1,$$

$$u(x_{n-1}) = \frac{1}{\alpha_{n-1}},$$

$$\cdots\cdots\cdots$$

$$u(x_1) = \frac{1}{\prod\limits_{i=1}^{n-1} \alpha_i}.$$

III. Случай, когда некоторые критерии являются качественными. Применяется методика, основанная на алгоритме, предложенном Р. Акофом и Р. Черчменом [1].

Предположим, что имеется n результатов x_1, x_2, ..., x_n. Методика определения полезности состоит из следующих этапов:

1. Упорядочивают все результаты по убыванию предпочтительности. Пусть x_1 — наиболее, а x_n — наименее предпочтительный результат.

Составляют таблицу возможных комбинаций результатов, достигаемых одновременно, и затем устанавливают их предпочтение относительно отдельных результатов x_1, x_2, ..., x_n (табл. 1.1).

Эту информацию о предпочтительности результатов получают от экспертов.

2. Приписывают начальные оценки полезностям отдельных результатов $u_0(x_1)$, $u_0(x_2)$, ..., $u_0(x_n)$. Затем подставляют начальные оценки в последнее соотношение табл. 1.1. Если оно удовлетворяется, то оценки не изменяют.

В противном случае производят коррекцию полезностей так, чтобы удовлетворялось данное соотношение.

3. После этого переходят к следующему соотношению. Процесс коррекции продолжается до тех пор, пока не образуется система оценок $u^*(x_1)$, $u^*(x_2)$, ..., $u^*(x_n)$, которая будет удовлетворять всем указанным в таблице соотношениям. Коррекцию следует производить таким образом, чтобы по возможности изменять оценки для минимального количества результатов.

Пример 1.4. Пусть эксперт упорядочивает пять результатов x_1, x_2, ..., x_5, приписав им следующие оценки: $u_0(x_1) = 7$; $u_0(x_2) = 4$; $u_0(x_3) = 2$;$u_0(x_4) = 1{,}5$; $u_0(x_5) = 1$.

Рассмотрев возможные варианты выбора, он высказал следующее суждение относительно ценности тех или иных комбинаций результатов:

1) $x_1 < x_2 + x_3 + x_4 + x_5$,
2) $x_1 < x_2 + x_3 + x_4$,
3) $x_1 < x_2 + x_3 + x_5$,
4) $x_1 > x_2 + x_3$,
5) $x_2 < x_3 + x_4 + x_5$,
6) $x_2 > x_3 + x_4$,
7) $x_3 > x_4 + x_5$.

Нужно произвести оценку полезности результатов так, чтобы удовлетворить всем неравенствам.

Подставляем начальные оценки в неравенство 7):

$$u_0(x_3) = 2 < u_0(x_4) + u_0(x_5) = 2{,}5.$$

Следовательно, неравенство 7) не удовлетворяется.

Изменяем полезность результата x_3

$$u_1(x_3) = 3$$

и проверяем неравенство 6):

$$u_0(x_2) = 4 < u_1(x_3) + u_0(x_4) = 4{,}5.$$

Это неравенство также не удовлетворяется.

Примем $u_1(x_2) = 5$. При этом неравенство 5) удовлетворяется.

Обращаемся к неравенству 4):

$$u_0(x_1) = 7 < u_1(x_2) + u_1(x_3) = 8.$$ Оно не выполняется.

Поэтому примем $u_1(x_1) = 8{,}5$.
Теперь неравенства 3), 2), 1) удовлетворяются.

Проверяем еще раз неравенства 6) и 7) при измененных значениях полезностей: $5 > 3 + 1{,}5$ и $3 > 1{,}5 + 1$. Оба неравенства выполняются.

Выпишем окончательные оценки полезности результатов: $u_1(x_1) = 8{,}5$; $u_1(x_2) = 5$; $u_1(x_3) = 3$; $u_1(x_4) = 1{,}5$; $u_1(x_5) = 1$.

Такая методика определения полезности применима, когда количество результатов n ограничено $n \leqslant 6$—7.

В случаях, когда $n > 7$, авторами предложена следующая модификация данной методики — способ коррекции оценок [1].

Множество результатов разбивают на подмножества, состоящие из 5—7 результатов и имеющие один общий результат, например, x_1. Затем приписывают начальные значения полезностям для всех результатов, причем полезность общего результата x_1 одинакова во всех подмножествах. Далее применяют способ коррекции оценок полезности независимо в каждом из подмножеств с ограничением $u(x_1) = \text{const}$. В результате получают систему полезностей с единой мерой для всех подмножеств $u(x_1)$.

Принятие решений в условиях риска

Эта задача возникает в том случае, когда с каждой принимаемой стратегией x_i связано целое множество возможных результатов $O_1, O_2, \ldots, O_m$ с известными вероятностями $p(O_j/x_i)$.

Формально модель задачи может быть представлена в виде следующей матрицы:

$$\mathbf{L} = \begin{array}{|c|c|c|c|c|}
\hline
x_i \backslash O_j & O_1 & O_2 & \ldots\ldots O_j \ldots\ldots & O_m \\
\hline
x_1 & l_{11} & l_{12} & \ldots\ldots l_{1j} \ldots\ldots & l_{1m} \\
\hline
x_2 & l_{21} & l_{22} & \ldots\ldots l_{2j} \ldots\ldots & l_{2m} \\
\ldots & \ldots & \ldots & \ldots\ldots\ldots\ldots & \ldots \\
x_i & l_{i1} & l_{i2} & \ldots\ldots l_{ij} \ldots\ldots & l_{im} \\
\ldots & \ldots & \ldots & \ldots\ldots\ldots\ldots & \ldots \\
\hline
x_n & l_{n1} & l_{n2} & \ldots\ldots l_{nj} \ldots\ldots & l_{nm} \\
\hline
\end{array}$$

$$l_{ij} = u(O_j, x_i),$$

где l_{ij} — полезность результата O_j при использовании решения x_i.

Пусть заданы условные вероятности $p(O_j/x_i)$, $j = \overline{1, m}$, $i = \overline{1, n}$.

Вводят ожидаемую полезность для каждой стратегии

$$E\{u(x_i)\} = \sum_{j=1}^{m} u(O_j, x_i)\, p(O_j/x_i), \quad i = 1, \ldots, n. \tag{3.9}$$

Решающее правило для определения оптимальной стратегии x_i записывают так:

$$E\{u(x_i)\} = \max_{x_k} E\{u(x_k)\}. \quad (3.10)$$

Принятие решений в условиях неопределенности

Одним из определяющих факторов в таких задачах является внешняя среда или природа, которая может находиться в одном из состояний $S_1, \ldots, S_k$, которые неизвестны лицу, принимающему решение (наблюдатель).

Тогда математическую модель задачи в условиях неопределенности можно сформулировать следующим образом.

Имеется некоторая матрица L размерностью $m \times n$.

$$\mathbf{L} = \begin{array}{|c|c|c|c|c|}
\hline
x_i \backslash O_j & O_1 & O_2 & \ldots & O_n \\
\hline
x_1 & l_{11} & l_{12} & \ldots & l_{1m} \\
\hline
\ldots & \ldots & \ldots & \ldots & \ldots \\
\hline
x_m & l_{m1} & l_{m2} & \ldots & l_{mn} \\
\hline
\end{array}$$

Элемент этой матрицы l_{ij} можно рассматривать как полезность результата O_j при использовании стратегии x_i

$$l_{ij} = u(O_j, x_i), \quad j = 1, \ldots, n, \quad i = 1, \ldots, m.$$

В зависимости от состояния среды результат O_j достигается с вероятностью $p(O_j/x_i, S_k)$.

Кроме того, наблюдателю неизвестно распределение вероятностей $p(S_k)$. Относительно состояния среды наблюдатель может высказывать определенные гипотезы. Его предположения о вероятном состоянии среды называются *субъективными вероятностями* $\widehat{p}(S_k)$, $k = 1, 2, \ldots, K$.

Если бы величина $p(S_k)$ была известна наблюдателю, то мы бы имели задачу принятия решений в условиях риска. В этом случае решающее правило x_i определяется следующим образом:

$$\max_{x_i} \sum_{j=1}^{n} \sum_{k=1}^{K} u(O_j, x_i)\, p(O_j/x_i, S_k)\, p(S_k) = \max_{x_i} E\{u(x_i)\}. \quad (3.11)$$

На самом деле состояния среды неизвестны и неизвестно также распределение вероятностей $p(S_k)$.

Как выбрать при этом оптимальную стратегию?

Существует несколько критериев для выбора оптимальной стратегии.

Критерий Вальда (критерий осторожного наблюдателя). Этот критерий оптимизирует полезность в предположении, что среда находится в самом невыгодном для наблюдателя состоянии. По данному критерию решающее правило имеет следующий вид:

$$\max_{x_i} \min_{S_k} u\,(x_i,\ S_k),$$

где

$$u\,(x_i,\ S_k) = \sum_{j=1}^{n} u\,(O_j,\ x_i)\,p\,(O_j/x_i,\ S_k). \qquad (3.12)$$

По критерию Вальда выбирают стратегию, которая дает гарантированный выигрыш при наихудшем варианте состояния среды.

Критерий Гурвица основан на следующих двух предположениях: среда может находиться в самом невыгодном состоянии с вероятностью $1 - \alpha$ и в самом выгодном — с вероятностью α, где α — коэффициент доверия.

Тогда решающее правило записывается так:

$$\max_{x_i} [\alpha \max_{S_k} u\,(x_i,\ S_k) + (1 - \alpha) \min_{S_k} u\,(x_i,\ S_k)], \qquad (3.13)$$

$$0 \leqslant \alpha \leqslant 1.$$

Если $\alpha = 0$, получаем критерий Вальда.

Если $\alpha = 1$, то приходим к решающему правилу вида $\max_{x_i} \max_{S_k} u\,(x_i,\ S_k)$, так называемая стратегия «здорового оптимиста», который верит в удачу.

Критерий Лапласа. Если неизвестны состояния среды, то все состояния среды считают равновероятными:

$$p\,(S_i) = p\,(S_j) = \cdots\ p\,(S_k).$$

В результате решающее правило определяется соотношением (3.11) при условии $p\,(S_k) = \frac{1}{K}$.

Критерий Сэвиджа (критерий минимизации «сожалений»). «Сожаление» — это величина, равная изменению полезности результата при данном состоянии среды относительно наилучшего возможного решения.

Чтобы определить «сожаление», поступают следующим образом. Строят матрицу $\overline{U} = \| u_{ik} \|$, где $u_{ik} = u\,(x_i,\ S_k)$; $i = \overline{1,\ m}$; $k = \overline{1,\ K}$.

В каждом столбце этой матрицы находится максимальный элемент

$$u_k = \max_i u_{ik}. \qquad (3.14)$$

Его вычитают из всех элементов этого столбца.

Далее строим матрицу «сожалений»

$$U_c = \| u_{ik\,c} \|,$$

$$u_{ik\,c} = u_{ik} - u_k.$$

Искомую стратегию x_i, которая минимизирует «сожаление», определяют из условия

$$\max_{x_i} \min_{S_k} u_{ikc}. \qquad (3.15)$$

Этот критерий минимизирует возможные потери при условии, что состояние среды наихудшим образом отличается от предполагаемого.

Рассмотрим частный случай предложенной выше модели задачи в условиях неопределенности.

Предположим, что каждому возможному состоянию среды соответствует один возможный исход:

$$p\,(O_j/S_k) = \delta_{jk}, \qquad (3.16)$$

где $\delta_{jk} = \begin{cases} 1, & \text{если } j = k, \\ 0, & \text{если } j \neq k. \end{cases}$

Таким образом, в данном случае математическая модель задачи принятия решений определяется множеством стратегий $X = \{x_i\}$, множеством состояний среды $S = \{S_k\}$, а также следующей матрицей:

$$\mathbf{L} = \begin{array}{|c|c|c|c|c|} \hline x \backslash S & S_1 & S_2 & \dots & S_k \\ \hline x_1 & l_{11} & l_{12} & \dots & l_{1k} \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline x_m & l_{m1} & l_{m2} & \dots & l_{mk} \\ \hline \end{array},$$

где $l_{ij} = u\,(x_i,\ S_j)$.

Множество $\{p\,(S_j)\}$ предполагается неизвестным.

В этом случае критерии для выбора оптимальной стратегии имеют следующий вид:

Критерий Вальда

$$\max_{x_i} \min_{S_k} u\,(x_i,\ S_k). \qquad (3.17)$$

Критерий Гурвица

$$\max_{x_i} \left[\alpha \max_{S_k} u\,(x_i,\ S_k) + (1 - \alpha) \min_{S_k} u\,(x_i,\ S_k)\right]. \qquad (3.18)$$

Критерий Лапласа

$$\max_{x_i} \frac{1}{K} \sum_{k=1}^{K} u\,(x_i,\ S_k). \qquad (3.19)$$

Критерий Сэвиджа:

$$\max_{x_i} \min_{S} u_c\,(x_i,\ S_k), \qquad (3.20)$$

где

$$u_c(x_i, S_k) = u(x_i, S_k) - \max_i u(x_i, S_k).$$

Рассмотрим использование данных критериев в условиях неопределенности для следующей практической ситуации [31].

Пример 1.5. Некоторая фирма решает построить отель в одном из курортных мест. Необходимо определить наиболее целесообразное количество мест или комнат в этой гостинице.

Составляют смету расходов по строительству гостиницы с различным количеством комнат, а также рассчитывают ожидаемый доход в зависимости от количества комнат, которые будут сняты.

В зависимости от принятого решения — количества комнат в гостинице $x = 20$, 30, 40, 50 и количества снятых комнат $S = 0$, 10, 20, 30, 40, 50, которое зависит от множества случайных факторов и неизвестно фирме, получают следующую таблицу ежегодных прибылей:

Таблица 1.2

S_k / x_i	0	10	20	30	40	50
20	—121	62	245	245	245	245
30	—168	14	198	380	380	380
40	—216	—33	150	332	515	515
50	—264	—81	101	284	468	650

Наиболее подходящее количество комнат в гостинице определяют по вышеприведенным критериям.

Критерий Вальда:

$$\max_{x_i} \min_{S_k} l_{ik} = -121; \quad x_{\text{опт}} = 20.$$

Судя по результатам, критерий Вальда неприменим, так как в этом случае от постройки гостиницы следует отказаться.

Критерий Лапласа:

$$\max_{x_i} \frac{1}{6} \sum_{k=1}^{6} l_{ik} = \max_{x_i} \overline{l_i} = \max\{153, 198, 210, 193\} = 210; \quad x_{\text{опт}} = 40.$$

Критерий Гурвица:

$$\max_{x_i} [\alpha \max_k l_{ik} + (1 - \alpha) \min_k l_{ik}].$$

Для разных α можно построить таблицу доходов по критерию Гурвица $H = \| h_{i\alpha} \|$, где

$$h_{i\alpha} = [\alpha \max_k l_{ik} + (1 - \alpha) \min_k l_{ik}]:$$

Таблица 1.3

α x_i	0,1	0,2	0,5	0,9
20	—84	—47	62	206
30	— 114	—59	108	325
40	—143	—70	150	442
50	—172	—81	193	560

Тогда оптимальное количество комнат в гостинице в зависимости от α:

α	0,1	0,2	0,5	0,9
$x_{\text{опт}}$	20	20	50	50

Критерий Сэвиджа. Строят матрицу «сожалений»:

Таблица 1.4

S_k x_i	0	10	20	30	40	50
20	0	0	0	—135	—270	—405
30	—47	—48	—47	0	—135	—270
40	—95	—95	—95	—48	0	—135
50	—143	—143	—144	—96	—47	0

$$\max_{x_i} \min_{S_k} u_{ikc} = \max\{-405,\ -270,\ -135,\ -143\} = -135.$$

Таким образом, предстоит сделать выбор между различными решениями:

по критерию Вальда строить 20 комнат;

по критерию Лапласа строить 40 комнат;

по критерию Гурвица строить 20 комнат, если заказчик — пессимист и 50 комнат, если он оптимист;

наконец, по критерию Сэвиджа следует строить 40 комнат.

Какое из возможных решений предпочтительнее? Это определяется выбором соответствующего критерия (Вальда, Лапласа, Гурвица или Сэвиджа).

Выбор критерия принятия решений является наиболее сложным и ответственным этапом в исследовании операций. При этом не существует каких-либо общих рекомендаций или советов. Выбор критерия должен производить заказчик на самом высоком уровне и в максимальной степени согласовывать этот выбор с конкретной спецификой задачи, а также со своими целями.

В частности, если даже минимальный риск недопустим, то следует применять критерий Вальда. Если наоборот, определенный риск вполне приемлем и заказчик намерен вложить в некоторое предприятие столько средств, чтобы потом он не сожалел, что вложено слишком мало, то выбирают критерий Сэвиджа.

При отсутствии достаточной информации для выбора того или иного критерия возможен альтернативный подход, который связан с вычислением шансов на успех и разорение на основе прошлого опыта.

§ 4. РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ В ЗАДАЧАХ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

Важнейшим этапом всякого операционного исследования является формализация постановки задачи, т. е. разработка математической модели операционной задачи. Под моделью понимают некоторый материальный или абстрактный объект, находящийся в определенном объективном соответствии с исследуемым объектом, несущий о нем определенную информацию и способный замещать его на определенных этапах познания. Главным свойством модели является то, что она несет определенную информацию об исследуемом объекте.

По степени соответствия оригиналу модели делятся на *изоморфные* и *гомоморфные*. Изоморфные модели находятся в строгом соответствии с оригиналом и дают о нем исчерпывающую информацию. Очевидно, изоморфные модели существуют только для простых систем, в частности механических.

Для сложных систем оказывается возможным построить модель только после упрощающих предположений. Такие модели, отражающие лишь некоторые определенные свойства оригинала, называются гомоморфными.

В частности, большинство операционных моделей организационных систем относится к числу гомоморфных. Их получают путем сознательного огрубления исследуемого процесса, значительного сокращения числа факторов, отбора среди них наиболее существенных. Тем самым, модель оказывается значительно проще объекта, благодаря чему и становится возможным ее исследование и анализ.

Классификация моделей

Классификация моделей многоаспектная. В частности, модели различаются по степени соответствия оригиналу, по принципу моделирования, назначению и т. д.

По степени соответствия объекта и модели последние делятся на изоморфные и гомоморфные.

По основному принципу моделирования модели делятся на: физические (модели геометрического подобия); аналоговые; символические; модели на универсальных ЭВМ.

Физические модели имеют ту же физическую природу, что исследуемый объект, но отличаются от него размерами, формой и другими характеристиками. Создание физических моделей основано на теории подобия.

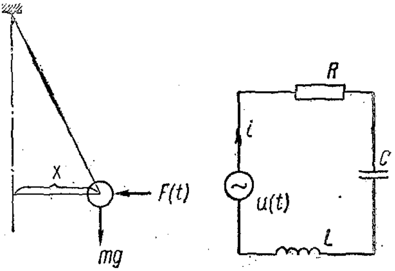


Рис. 1.2.

Типичным примером такой модели является аэродинамическая труба. Здесь свойства летательного аппарата в полете отрабатываются на модели той же физической природы.

Другой пример физической модели: действующая модель транспортных средств (например, глиссер, вездеход, планер и т. д.).

В *аналоговых моделях* набор одних свойств используется для отображения свойств другой физической природы. Наиболее распространенными моделями такого типа являются электрические аналоги, с помощью которых удается смоделировать механические, гидравлические, транспортные, экономические и другие системы. Например, пусть рассматривается объект, который представляет собой тяжелую материальную точку, подвешенную на невесомой оси. Процесс движения маятника может изучаться как при непосредственном наблюдении за его колебаниями, так и при исследовании его электрического аналога — колебательного контура, включающего катушку индуктивности, конденсатор и активное сопротивление (рис. 1.2).

Символические модели — это абстрактные модели, в которых используется язык математики.

Символические модели задач исследования операций обычно имеют вид:

$$\text{найти} \quad \max f(x),$$

$$\text{при условии} \quad \{g_i(x) \leqslant b_i;\ i=\overline{1,m};\ x \geqslant 0\}.$$

Символические модели наиболее удобны для исследования и количественного анализа. Они позволяют не только получить решение для конкретного случая (задачи), но и (что чрезвычайно важно) определить влияние параметров системы на результат решения.

Символической моделью маятника является уравнение движения его центра масс вида

$$m\frac{d^2x}{dt^2}+k\frac{dx}{dt}+ex=F(t). \qquad (4.1)$$

Сравнивая (4.1) с уравнением для колебательного контура

$$L\frac{d^2i}{dt^2}+R\frac{di}{dt}+\frac{1}{c}\,i=\frac{dU(t)}{dt}, \qquad (4.2)$$

видим, что эти уравнения одного класса и совпадают с точностью до параметров ($m = L$, $k = R$, $\frac{1}{c} = e$). Именно это и позволяет заменить рассмотрение механической системы ее электрическим аналогом.

Если решить дифференциальное уравнение (4.1), мы тем самым найдем решение символической модели.

Итак, символическая модель — это абстрактно-математическая модель объекта, заданная в виде формул, функций или функциональных отображений, уравнений и других математических соотношений. Символические модели носят универсальный характер.

Моделирование на ЭВМ является наиболее мощным средством моделирования сложных систем произвольной физической природы. Для применения этого метода необходимо получить первоначально символическую модель. Если аналитические решения на модели получить невозможно или требуется провести глубокое исследование влияния варьируемых параметров на характеристики модели, прибегают к моделированию на ЭВМ. Моделирование на ЭВМ является основным средством моделирования в исследовании операций.

В зависимости от необходимости учета статистических факторов модели разделяются на *детерминированные,* где все параметры и воздействия предполагаются причинно обусловленными, и *стохастические,* в которых важную роль играют случайные переменные.

По свойствам в переходном режиме модели принято делить на статические и динамические. В динамических моделях рассматриваются характеристики и взаимодействия системы, меняющиеся во времени, в отличие от статических систем, в которых эти параметры не зависят от времени. Наконец, в зависимости от того, содержат ли модели управляемые факторы или не содержат, они делятся на конструктивные и дескриптивные, или описательные. Описательные модели не включают управляемых переменных и потому не позволяют решать задачу о нахождении наиболее эффективного управления.

Конструктивные модели включают управляемые переменные и потому в принципе служат основным целям исследования операций — отысканию наиболее эффективного управления организационной системой.

Описательная модель обычно играет вспомогательную роль и чаще всего предшествует созданию конструктивной модели на первом этапе формулировки задачи.

Принципы построения моделей в исследовании операций

Качество разрабатываемых моделей в значительной мере определяется опытом, интуицией, а также творческими способностями операциониста. Невозможно дать готовые рецепты, как строить математическую модель в той или иной конкретной ситуации, и тем более, как написать учебник, как создавать модели.

Даже если бы такой учебник и был написан, то, как справедливо заметили Р. Акоф и М. Сасиени, «он скорее всего привел бы к ограничению творческих способностей, а не к их развитию» [1].

Тем не менее анализ накопленного операционистами опыта позволяет выявить рациональные принципы, которые облегчают и рационализируют процесс построения моделей. Использование того или иного принципа определяется как степенью понимания структуры исследуемой системы, так и возможностью проникновения в ее внутренний механизм.

Ниже будут рассмотрены некоторые принципы построения моделей. Разумеется, что ими не исчерпываются все известные принципы разработки моделей, и опытный операционист может дополнить их перечень.

Принцип 1. Изучение и анализ причинно-следственных связей. Этот интуитивно понятный принцип используется всегда, когда структура системы достаточно проста и может быть вскрыта путем непосредственного обследования системы и опроса лиц, вовлеченных в ее функционирование.

Рассмотрим следующий пример. Имеется некоторая снабженческая контора, которая поставляет некоторый вид продукта своим клиентам. Хранение единицы продукта связано с постоянными расходами.

Спрос на продукты случаен и неуправляем.

Предположим, что на основе статистических исследований построена оценка функции распределения спроса $p(r)$, $r = 0, 1, \ldots, N_{\max}$. Если спрос превышает запас, то контора платит штраф, пропорциональный величине нехватки. За каждую единицу продукта, поставленного клиентам, контора получает прибыль.

Требуется определить такой объем запасов, при котором минимизируется сумма ожидаемых расходов по хранению запасов и потерь вследствие дефицита.

Возможна и несколько другая постановка. Найти средний объем заказа x, при котором максимизируется ожидаемый доход конторы.

Структура причинно-следственных связей здесь оказывается ясной и можно непосредственно перейти к составлению математической модели.

Введем следующие обозначения:

x — число заказываемых единиц продукта в неделю; a — прибыль на каждую проданную единицу продукта; d — штраф за единицу дефицита; s — стоимость хранения единицы продукта ($s < a$); r — спрос за неделю; $p(r)$ — вероятность спроса r.

Перейдем к составлению модели.

Общий выигрыш системы за неделю — ax (при объеме заказа x); потери по хранению товаров — sx; вероятность дефицита $p(r > x) = \sum_{r=x+1}^{\infty} p(r)$; потери из-за дефицита $d\sum_{r=x+1}^{\infty}(r-x)p(r)$; общий ожидаемый доход за неделю

$$D(x) = a\sum_{r=0}^{x} p(r)\cdot r + ax\sum_{r=x+1}^{N_{\max}} p(r) - sx - d\sum_{r=x+1}^{N_{\max}}(r-x)p(r), \qquad (4.3)$$

где $N_{\max}$ — максимально возможная величина спроса.

Требуется найти такой объем заказа x, при котором максимизируется доход $D(x)$. Как видим, в данной модели имеются управляемая переменная x, неуправляемая — r и неуправляемые константы: a, d, s.

Принцип 2. Использование аналогии.

В ситуациях, когда структура системы (задачи) достаточно очевидна, но метод ее математического описания неясен, можно воспользоваться ее сходством с системой, имеющей более простую структуру либо более изученной. В этих случаях можно воспользоваться либо более простой аналогичной системой либо (что гораздо удобнее) ее математической моделью.

Приведем следующий пример.

Имеется m пунктов возникновения информации $A_1, \ldots, A_m$, интенсивность потока информации из пункта A_i равна λ_i, $i = \overline{1, m}$.

Имеется также n пунктов централизованной обработки информации $B_1, B_2, \ldots, B_n$. Эти пункты могут представлять собой ВЦ коллективного пользования

Обозначим через Λ_j производительность ВЦ в пункте B_j.

Пусть стоимость передачи информации объемом x_{ij} из пункта A_i в B_j определяется следующим соотношением:

$$c_{ij}(x_{ij}) = \begin{cases} c_{ij}x_{ij} + d_{ij}, & \text{если } x_{ij} > 0; \\ 0, & \text{если } x_{ij} = 0, \end{cases} \tag{4.4}$$

где c_{ij} — удельные затраты по передаче единицы информации по каналу (i, j); d_{ij} — фиксированные затраты, например, месячная арендная плата.

Пусть стоимость обработки единицы информации на ВЦ$_j$ равна c_j. *Требуется определить наиболее экономичный вариант обработки информации*, т. е. объемы информационно-вычислительных работ, выполняемых всеми ВЦ для всех абонентов, при которых суммарные затраты на передачу и переработку информации в сети ВЦ будут минимальными.

Обозначим через λ_{ij} объем информации абонента A_i, перерабатываемый в ВЦ$_j$. Тогда необходимо найти такие λ_{ij}^0, при которых

$$L(\lambda_{ij}^0) = \min_{\{\lambda_{ij}\}} \sum_i \sum_j c_{ij}(\lambda_{ij}) + \sum_j c_j \sum_i \lambda_{ij} \tag{4.5}$$

при условии

$$\sum_j \lambda_{ij} = \lambda_i, \tag{4.6}$$

$$\sum_i \lambda_{ij} \leqslant \Lambda_j. \tag{4.7}$$

Хотя по содержательной постановке эта задача не относится к транспортным, по структуре математической модели она аналогична ей. Поэтому для решения ее можно использовать соответствующий аппарат, разработанный для транспортных задач.

Указанный выше принцип положил начало целому направлению в теории моделирования, известному под названием аналогового.

Принцип 3. Проведение экспериментов для выявления существенных переменных.

В тех случаях, когда анализ данных не позволяет определить влияние отдельных переменных на показатели работы системы, возникает необходимость провести эксперименты с целью выявления существенных переменных и их влияния на эти показатели.

Таким образом, отличительной особенностью этого принципа является использование эксперимента.

Приведем следующий пример [1].

Нефтяную компанию интересовала прогностическая оценка ее доходов от новых автозаправочных станций, которые она хотела построить в ряде пунктов. Требовалось исключить такие пункты, где сооружение станций могло оказаться нерентабельным. Первоначально эта задача была предложена группе специалистов по исследованию сбыта, которые опросили многих сотрудников и выяснили их мнение относительно того, какие параметры автозаправочной станции (АЗС) влияют на объем продажи бензина и других горюче-смазочных материалов. Было выявлено около 65 переменных.

Далее был проведен множественный линейный регрессионный анализ статистических данных, собранных по функционированию нескольких сотен станций. В результате этого анализа была отобрана примерно половина всех параметров, попавших в разряд «статистически значимых».

Однако полученная после отсева незначимых переменных модель не обеспечивала требуемой точности оценок будущего сбыта.

Впоследствии эта задача была поставлена перед операционной группой. Операционисты были против учета столь большого числа факторов, так как были убеждены, что, как правило, степень понимания физики процесса обратно пропорциональна числу переменных, фигурирующих в его описании. Поэтому было принято решение выяснить, насколько можно продвинуться в решении задачи, рассматривая сначала только одну переменную. Было высказано предположение, что, возможно, интенсивность движения является наиболее существенной переменной в забракованной модели регрессии. Была поставлена задача вскрыть механизм влияния этой неуправляемой переменной на объем продажи бензина для заправки.

В связи с этим был разработан точный метод описания движения автотранспорта. Существует всего 16 возможных маршрутов, по которым автомашина может проходить через дорожные пересечения, где обычно размещаются АЗС. Изучение движения показало, что лишь небольшое число маршрутов оказывает существенное влияние на объем продажи бензина. Упорядочение маршрутов движения по такому показателю, как процент автомашин, останавливающихся для заправки, навело на мысль, что по мере того, как время, затрачиваемое на остановку, возрастает, процент заправляющихся автомашин, следующих по данному маршруту, уменьшается. Эта гипотеза была затем проверена экспериментально. Была также найдена связь между потерями времени на заправку и процентом останавливающихся на АЗС машин.

Затем были подвергнуты испытанию и остальные переменные, признанные статистически значимыми в ходе регрессионного анализа. Требовалось выяснить, влияет ли какая-либо из этих переменных на время, затрачиваемое на заправку. Было установлено, что большинство из них оказывает такое влияние. В частности, к этим переменным относятся: число въездов и выездов, их расположение, число бензоколонок, численность обслуживающего персонала и т. д.

Сначала влияние этих переменных было оценено теоретическим путем, а затем полученные оценки были проверены и подтверждены экспериментально. В результате была построена новая модель, которая давала значительно более точные прогнозы оценки продажи бензина для вновь строящихся АЗС.

Этого удалось достигнуть благодаря вскрытию внутренних глубинных взаимосвязей между неуправляемыми и контролируемыми переменными и формулировке основной гипотезы о связи между потерями времени на остановку для заправки и объемом продажи бензина.

§ 5. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ ОРГАНИЗАЦИОННОГО УПРАВЛЕНИЯ

В случаях, когда необходимо провести исследование сложных систем организационного управления, для которых отсутствуют аналитические модели (методы описания), традиционные средства моделирования оказываются бессильными. В этом случае прибегают к так называемому «имитационному моделированию».

Имитационное моделирование (от латинского «имитацио» — подражание, копирование) обозначает процесс имитирования поведения системы на модели с целью получения информации о ней.

Согласно Шеннону имитационное моделирование есть «процесс конструирования модели реальной системы и постановки экспериментов на ней с целью понимания поведения системы либо оценки (в рамках ограничений, накладываемых некоторым критерием или совокупностью критериев) различных стратегий, обеспечивающих функционирование данной системы» [53].

Хотя имитационное моделирование было известно до возникновения ЭВМ, однако только с развитием вычислительной техники оно стало универсальным и мощным инструментом исследования систем.

Благодаря большому быстродействию современных ЭВМ оказывается возможным за сравнительно короткое время «проиграть» поведение сложной системы на модели и выполнить анализ ее функционирования.

В достаточно сложных системах организационного управления всегда имеются некоторые случайные или неопределенные факторы, воздействующие на систему и вызывающие флуктуации характеристик ее поведения.

Если этими колебаниями можно пренебречь, то разрабатывается детерминированная модель. Если же случайные факторы играют существенную роль в функционировании системы, то возникает необходимость в создании статистических моделей. Термин «имитационное моделирование» является более широким, чем «статистическое моделирование», и включает в себя статистические модели как составную и неотъемлемую часть.

Имитационное моделирование (ИМ) используется для следующих целей:

а) описания поведения систем;

б) построения теорий и гипотез, которые могут объяснить наблюдаемое поведение;

в) предсказания будущего поведения системы.

ИМ нашло свое применение в самых различных сферах человеческой деятельности, в частности: а) экономике, б) технике, в) биологии, г) социологии, д) на транспорте, е) в связи, ж) торговле, з) исследовании проблем экологии, эволюции и т. п.

О широте использования ИМ при решении задач исследования операций дает представление следующая таблица, в которой представлены результаты проведенного Шенноном и Байлесом обследования научной деятельности группы действительных членов Американского общества исследования операций [53].

Таблица 1.5

Полезность методов исследования операций в повседневной научной работе

№ п.п	Методы	Относительная ценность метода
1	Теория вероятностей и статистические оценки	0,182
2	Экономический анализ (оценка эффективности затрат)	0,150
3	Имитационное моделирование	0,143
4	Линейное программирование	0,120
5	Управление запасами	0,097
6	Массовое обслуживание (теория очередей)	0,085
7	Сетевые модели (включая методы календарного планирования и СПУ)	0,072
8	Модели замены	0,042
9	Теория игр	0,040
10	Динамическое программирование	0,031
11	Методы поиска	0,020
12	Нелинейное программирование	0,018
	Всего	1,000

Отметим условия, при которых может оказаться целесообразным использование имитационного моделирования:

1) не существует законченной математической модели данной задачи либо еще не разработаны аналитические методы решения сформулированной математической модели, к этой группе относятся, например, многочисленные модели массового обслуживания;

2) аналитические методы имеются, но соответствующие им математические процедуры столь сложны и трудоемки, что имитационное моделирование дает более простой способ решения задачи;

3) аналитические решения существуют, но их реализация невозможна вследствие недостаточной математической подготовки имеющегося персонала. В этом случае следует сопоставить затраты на проектирование, испытания и работу модели с затратами, связанными с приглашением специалистов со стороны;

4) имитационное моделирование может оказаться единственной возможностью вследствие невозможности постановки экспериментальных наблюдений в реальных условиях; соответствующим примером может служить изучение поведения космических кораблей в условиях межпланетных полетов;

5) кроме оценки определенных параметров желательно осуществить на имитационной модели наблюдение за ходом процесса в течение определенного периода времени;

6) для долговременного анализа поведения систем или медленно протекающих процессов может понадобиться сжатие временной шкалы; здесь возможности имитационного моделирования чрезвычайно широки, оно позволяет полностью контролировать время изучения процесса, замедляя или ускоряя его путем изменения масштаба времени моделирования. К этой категории задач относятся, например, исследования процессов эволюции, развития городов, экологических систем и др.

Несмотря на столь широкие возможности имитационного моделирования как универсального инструмента исследования сложных систем, ему свойственны и определенные ограничения и недостатки. К их числу относятся следующие:

1. Имитационная модель в принципе неточна и мы не в состоянии измерить степень этой неточности. Это затруднение может быть преодолено лишь частично путем анализа чувствительности модели к изменению определенных параметров;

2. В отличие от аналитической модели, имитационная не позволяет непосредственно оценить влияние на поведение системы варьирования тех или иных ее контролируемых параметров, поскольку она дает численные результаты. Получение такой информации связано с дополнительными экспериментами, это затрудняет применение ИМ для оптимизации систем;

3. Разработка хорошей имитационной модели часто обходится дорого и требует значительного времени и наличия высокоодаренных специалистов. Дж. Форрестер, в частности, указывает, что для создания хорошей модели внутрифирменного планирования может понадобиться от трех до 11 лет [53].

Указанные обстоятельства показывают, что хотя ИМ является чрезвычайно ценным и полезным методом для решения сложных задач управления, этот метод, конечно, не панацея для решения всех проблем управления. Разработка и применение ИМ все еще больше искусство, чем наука. Следовательно, как и в других видах искусства, успех или неудача определяется не столько методом, сколько тем, как он применяется.

Требования к хорошей модели. Определим ряд существенных черт, которыми должна обладать хорошая имитационная модель. Модель должна быть:

1) связана с функционированием исследуемой системы;

2) ориентирована на решение проблем реального мира;

3) построена так, чтобы служить подспорьем тем, кто управляет системой или анализирует ее поведение.

Поскольку имитация связана с решением реальных задач, мы должны быть уверены в том, что конечные результаты точно отражают истинное положение вещей. Следовательно, модель, которая может дать абсурдные ответы, должна быть сразу взята под подозрение. Любая модель должна быть оценена по максимальным пределам изменений величин ее параметров и переменных. Если модель дает нелепые ответы на поставленные вопросы, необходимо вернуться к этапу конструирования модели.

Модель должна быть способна отвечать на вопросы типа, «а что если... (изменить параметры системы, характеристики входных воздействий или управляемых переменных и т. п.)», поскольку это именно те вопросы, которые наиболее важны для исследователя или лица, принимающего решения.

Исходя из вышесказанного, сформулируем основные требования, которым должна удовлетворять хорошая модель. Она должна быть: целенаправленной; простой и понятной пользователю; надежной в смысле гарантии от абсурдных ответов; удобной в управлении и обращении; полной с точки зрения возможностей решения главных задач; адаптивной, позволяющей переходить к другим модификациям или обновлять данные; допускающей постепенные изменения или усложнения.

Основные этапы процесса имитации

Процесс имитационного моделирования включает следующие этапы [53]:

1. Постановку проблемы.
2. Анализ системы — установление границ, ограничений и показателей эффективности системы, подлежащей изучению.
3. Конструирование модели — переход от реальной системы к некоторой логической схеме (модели).
4. Подготовку данных — отбор данных, необходимых для построения модели, и представление их в соответствующей форме.
5. Трансляцию модели — описание модели на языке программирования.
6. Оценку адекватности — повышение до приемлемого уровня степени уверенности, с которой можно судить относительно корректности выводов о реальной системе, полученных на основе обращения к модели.
7. Стратегическое планирование — *планирование эксперимента*, который должен дать необходимую информацию.
8. Тактическое планирование — определение способа проведения каждой серии испытаний, предусмотренных планом эксперимента.
9. Экспериментирование — процесс осуществления имитации с целью получения желаемых данных и анализа чувствительности.

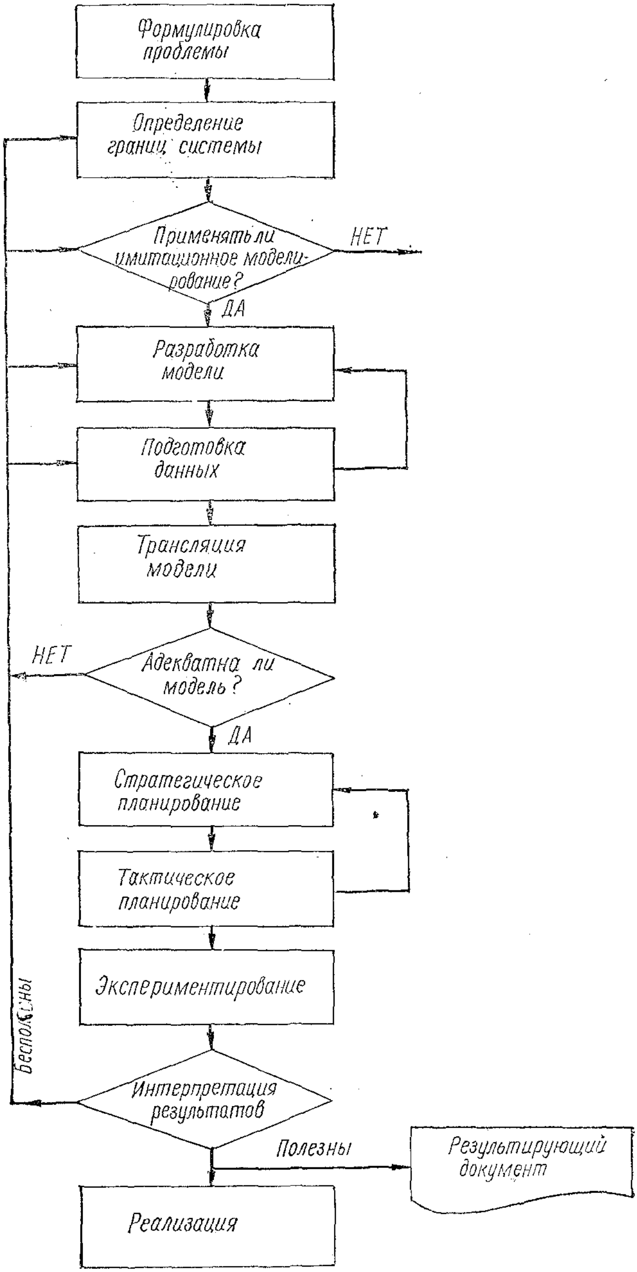

Формулировка проблемы
Определение границ системы
Применять ли имитационное моделирование?
НЕТ
ДА
Разработка модели
Подготовка данных
Трансляция модели
НЕТ
Адекватна ли модель?
ДА
Стратегическое планирование
Тактическое планирование
Экспериментирование
Бесполезны
Интерпретация результатов
Полезны
Результирующий документ
Реализация

Рис. 1.3.

10. Интерпретацию результатов — т. е. построение выводов по данным, полученным путем имитации.

11. Реализацию — практическое использование модели и результатов моделирования.

12. Документирование — т. е. регистрацию хода осуществления проекта и его результатов, а также процесса создания и использования модели.

Этапы процесса имитации в их взаимосвязи приведены на рис. 1.3. Поскольку многие из перечисленных этапов совпадают с соответствующими этапами разработки моделей других типов, дадим характеристику лишь некоторых специфических этапов имитационного моделирования.

Проверка модели. Проверка модели представляет собой процесс, в ходе которого достигается приемлемый уровень уверенности в том, что любой вывод о поведении системы, сделанный на основе моделирования, будет правильным. Проверка модели — этап чрезвычайно важный, ввиду того что имитационные модели создают впечатление реальности и как разработчики моделей, так и их пользователи легко проникаются к ним доверием. Поскольку исходные допущения (предположения), на основе которых строилась модель, могут оказаться скрытыми или непонятными для пользователя за массой деталей в описании функционирования, проверка, выполненная без должной тщательности, может привести к катастрофическим последствиям. Такого процесса, как «испытание» правильности модели, не существует. Вместо этого экспериментатор в ходе разработок должен провести серию проверок, чтобы укрепить свое доверие к модели.

Применяя первую из них, мы должны убедиться в том, что модель верна в первом приближении. Например, следует поставить такой вопрос: не будет ли модель давать абсурдные ответы, если ее параметры будут принимать предельные значения. Необходимо убедиться также и в том, что результаты, которые мы получаем, имеют смысл.

Для этого необходимо сравнить результаты на выходе имитационной модели с данными, получаемыми на выходе реальной системы.

Второй метод оценки адекватности модели состоит в проверке исходных предположений и третий — в проверке преобразований информации от входа к выходу.

Последние два метода могут привести к необходимости использовать статистические выборки для оценки средних значений и дисперсий, дисперсионный анализ, регрессионный анализ, факторный анализ, методы статистической проверки гипотез.

Таким образом, способы оценки имитационной модели делят на три категории [53]: 1) *верификацию модели,* используя которую, экспериментатор хочет убедиться в том, что модель ведет себя так, как было задумано; 2) *оценку адекватности* — проверку соответствия между поведением модели и поведением реальной системы; 3) *проблемный анализ* — формирование статистически значимых выводов на основе данных, полученных путем машинного моделирования.

Стратегическое и тактическое планирование. Поскольку имитационное моделирование представляет собой процесс экспериментирования на модели с целью получения информации о реальной системе, необходимо позаботиться о планировании экспериментов на модели с целью извлечения максимально возможного объема информации при ограниченных затратах на выполнение экспериментов. При этом в зависимости от целей и содержания выделяют стратегическое и тактическое планирование.

Стратегическое планирование отвечает на вопрос, как планировать эксперименты, которые дают желаемую информацию. Планирование эксперимента широко используется в имитационном моделировании систем. В любом экспериментальном исследовании можно выделить два типа задач: 1) определение сочетания параметров, которое оптимизирует переменную отклика (целевую функцию системы); 2) выяснение соотношения между переменной отклика и контролируемыми в системе параметрами. Для обеих этих задач разработаны и могут быть использованы методы планирования экспериментов. Цель использования планируемых экспериментов двоякая: 1) обеспечение экономии с точки зрения уменьшения числа необходимых экспериментов; 2) задание структурной основы обучения самого исследователя.

Тактическое планирование связано с определением способов проведения испытаний, намеченных планом эксперимента. Тактическое планирование прежде всего связано с решением задач двух типов: 1) определением начальных условий в той мере, как они влияют на достижение установившегося режима; 2) возможно бо́льшим сокращением дисперсии получаемых оценок при одновременном сокращении необходимых размеров выборки.

Первая задача возникает вследствие искусственного характера функционирования модели. В отличие от реального объекта, который функционирует непрерывно, имитационная модель работает эпизодически, экспериментатор запускает модель, делает свои наблюдения, а затем выключает ее. Всякий раз, когда начинается новый прогон модели, требуется некоторое время, чтобы модель пришла в установившийся режим, соответствующий режиму работы объекта. Таким образом, начальный период работы модели искажается из-за влияния начальных условий в момент ее запуска. Следовательно, задача состоит в определении таких начальных условий, при которых время переходного процесса в модели было бы по возможности минимальным.

Вторая задача тактического планирования связана с необходимостью оценить точность результатов эксперимента и степень надежности заключений или выводов.

Поскольку при имитационном моделировании мы оперируем реализациями случайных процессов, все определяемые по результатам экспериментов характеристики представляют собой некоторые оценки случайных величин и процессов. Для уменьшения разброса характеристики предложен ряд методов, которые могут существенно снизить требуемый размер выборки и число повторений эксперимента [7, 53].

Экспериментирование и анализ чувствительности. Одним из наиболее важных понятий в ИМ является анализ чувствительности. Под ним понимается определение степени изменчивости (чувствительности) окончательных результатов моделирования к изменению значений параметров модели. Анализ чувствительности состоит в том, что величины параметров модели систематически варьируют в некоторых пределах и при этом наблюдается изменение исследуемых характеристик модели.

Поскольку многие параметры, задаваемые в модели, определяются на основе весьма недостоверной информации, знание чувствительности модели к вариациям значений этих параметров очень важно. Если в ходе экспериментов выясняется, что модель нечувствительна (или малочувствительна) к изменению некоторого параметра, достаточно ограничиться в модели его средним значением. Если же при незначительных вариациях величины некоторого параметра результаты моделирования изменяются сильно, то это является основанием для затраты дополнительного времени или средств с целью получения более точных оценок этого параметра.

Заметим, что имитационное моделирование как нельзя лучше подходит для анализа чувствительности благодаря тому, что экспериментатор может на модели варьировать любой параметр в желаемых пределах и наблюдать за изменениями в поведении модели, полностью контролируя весь ход эксперимента.

Пример 1.6. *В качестве иллюстрации применения имитационного моделирования* в задачах исследования операций рассмотрим типичную производственную задачу обслуживания станков[1].

Среди практических задач массового обслуживания типичной и очень важной в практическом отношении является задача обслуживания станков. В самой простой форме ее можно продемонстрировать на примере наблюдения за рабочим-многостаночником. Закрепленные за ним станки время от времени останавливаются и простаивают до тех пор, пока рабочий не подойдет и не устранит неисправность. Если одновременно остановятся два или более станков, то рабочий может подойти сначала к одному из них, и, следовательно, другие станки будут в ожидании ремонта простаивать и не давать продукции. Такая ситуация приводит к постановке задачи обслуживания станков. Производственные потери имеют место и в период ремонта, и в интервале ожидания обслуживания. Производительность каждого станка и каждого рабочего зависит от частоты ремонта, его продолжительности и числа станков, обслуживаемых одним рабочим. Поэтому важно знать оптимальное число рабочих, требуемых для обслуживания данного количества станков.

Постановка задачи. Опишем реальную производственную ситуацию, приведшую к задаче обслуживания станков. Имеется группа станков, причем в общем случае в связи с различной степенью износа станки обладают неодинаковыми характеристиками. Станок может находиться как в рабочем, так и в выключенном состоянии в связи с его ремонтом, наладкой.

Каждый станок является источником заданий. Всего в результате анализа было выделено 5 типов заданий, например, установка новой заготовки, ремонт станка и т. д.

Каждое задание характеризуется периодичностью появления, длительностью выполнения, резервом времени на выполнение, минимальным и максимальным размером бригады, которая может его выполнять. Периодичность (время между двумя последовательными появлениями одного задания) может зависеть как от времени работы станка, так и от реального времени. Например, моменты установки заготовок в станок определяются темпом работы станка, тогда как моменты занесения рабочим записей в журнал учета есть функция реального времени. Обычно предполагают, что интервалы между соседними появлениями одного и того же события (задания) подчинены экспоненциальному закону распределения или распределению Эрланга, или какому-то другому стандартному распределению. Такое предположение должно быть

---

[1] Пример взят из монографии Р. Шеннона «Имитационное моделирование систем —искусство и наука». Мир, М., 1978.

Таблица 1.6

Пример описания характеристик заданий

Номер задания	Специальность рабочего	Требуемое состояние станка	Может прерывать другие задания	Может быть прервано	Закон распределения времени поступления		Закон распределения интервалов резервного времени		Закон распределения времени выполнения задания		Численность бригады [2]	Коэффициент эффективности бригады	Метод планирования
					тип [1]	математическое ожидание (параметры)	тип [1]	математическое ожидание (параметры)	тип [1]	математическое ожидание (параметры)			
1	Станочник	Выключен	Да	Нет	Э	25,0	Д	0,0	Н	2,5 (среднеквад. отклонение 0,5)	1 2	1,6	Назначается на станок, выполнивший предыдущее задание
2	Станочник или подсобный рабочий	Включен	Да	Нет	Э	40,0	Д	30,0	Т	4,0 (пределы 3,0; 5,0)	1 1	1,0	Назначается по готовности к выполнению
3	Подсобный рабочий	Включен или выключен	Нет	Нет	Д	18,0	Д	20,0	Т	2,0 (пределы 1,5; 2,5)	1 1	1,0	То же
4	Станочник или подсобный рабочий	То же	Нет	Да	Р	8,0 (пределы 3,0; 13,0)	Р	3,0 пределы	Р	1,0 (пределы 0,5; 1,5)	1 1	1,0	Ставится в очередь на станок, где начато выполнение другого задания
5	Слесарь-ремонтник	Выключен	Да	Нет	Э	80,0	Д	0,0	Э	10,0	1 1	1,0	Назначается на станок, выполнивший предыдущее задание

---

1) Д — *детерминированная* величина; Н — нормальное; Р — равномерное; Э — Эрланга; Т — треугольное распределение.

2) Трудоемкость задания (в чел.-мин) при максимальной численности бригады рассчитывается путем умножения времени, необходимого для бригады минимального состава, на коэффициент эффективности. В случае бригады переменной численности это время определяется линейной интерполяцией между временем для наибольшей и для наименьшей численности.

обязательно проверено на реальных данных. Некоторые задания могут или должны выполняться не одним рабочим, а целой бригадой. В таком случае должно быть задано время для выполнения всех возможных вариантов задания.

Кроме того, для некоторых заданий может вводиться резерв времени. Он означает, что задание может быть готово к выполнению и ожидать обслуживания в течение заданного резерва времени и это не приведет к простою.

Задания выполняются рабочими — операторами разных профессий, различной квалификации. Некоторые задания выполняются рабочими-подсобниками. Наконец,

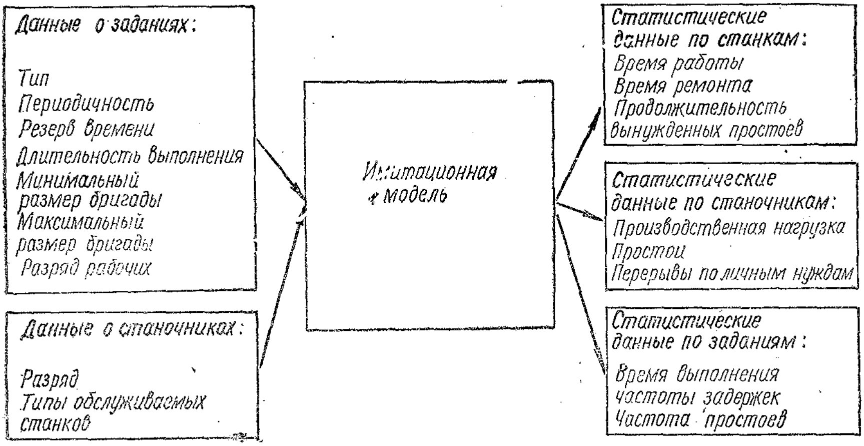


Рис. 1.4.

имеются и такие задания, которые могут выполняться рабочими нескольких профессий. Время обслуживания задания представляет собой случайную величину с известным законом распределения. Основные характеристики заданий и распределения длительности их обслуживания приводятся в табл. 1.6.

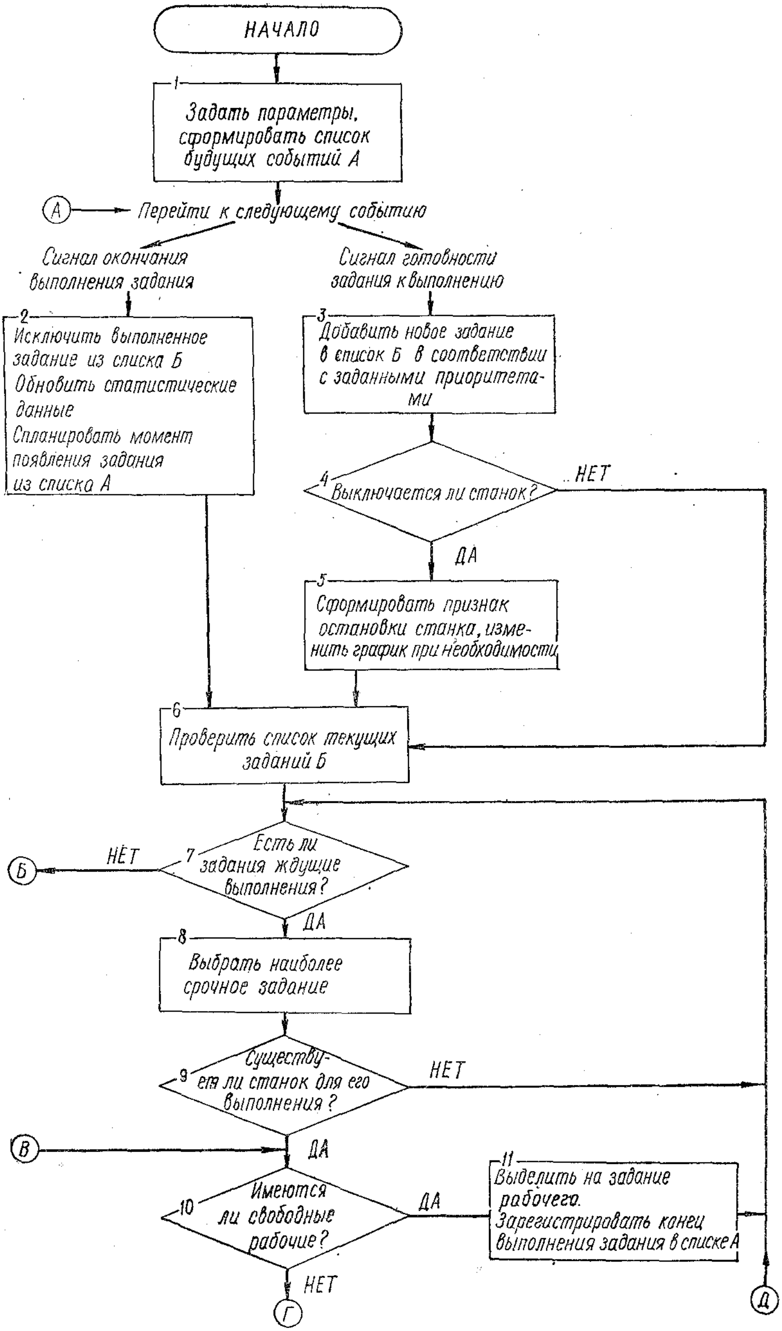
Вводятся приоритеты в обслуживании заданий. Каждое задание может иметь один из трех приоритетов: 1) может прерывать другие задания, но само не может быть прервано; 2) не может прерывать и не может быть прервано; 3) не может прерывать, но может быть прервано.

Кроме того, для несрочных заданий (т. е. таких, выполнение которых может быть отложено при отсутствии рабочей силы), вводится дополнительная категория. Эти задания не вызывают наложения операций обслуживания, но увеличивают рабочую нагрузку.

Предположим, что предприятие начинает выпуск нового типа изделий. Имеется шесть станков, которые изготавливают эти изделия и требуют периодического обслуживания со стороны рабочих-станочников. Некоторые виды заданий на обслуживание появляются по графику, тогда как другие возникают в случайные моменты времени. Станки обслуживаются рабочими трех специальностей — станочником, подсобным рабочим и слесарем-ремонтником, имеющими разную часовую тарифную ставку. Различия в уровне квалификации не позволяют каждому рабочему выполнять все операции. Известны потери на производстве за единицу времени простоя станка. Требуется определить оптимальное число рабочих каждой категории, при котором минимизируется сумма затрат на рабочую силу и потерь вследствие простоев станков от несвоевременного обслуживания.

Выбор метода решения. Решение указанной задачи определения необходимого числа рабочих может быть получено одним из следующих традиционных методов.

1. Использование таблиц режимов обслуживания. Недостаток этого метода — необходимость в слишком упрощенной постановке задачи, так как при составлении таблиц действуют следующие ограничения: однотипность станков, одинаковая квалификация операторов, способность каждого из них работать на любом из станков, выполнение одного задания только одним оператором, недопустимость прерывания заданий низкого приоритета при появлении более приоритетных заданий.


НАЧАЛО
1
Задать параметры, сформировать список будущих событий А
А
Перейти к следующему событию
Сигнал окончания выполнения задания
Сигнал готовности задания к выполнению
2
Исключить выполненное задание из списка Б
Обновить статистические данные
Спланировать момент появления задания из списка А
3
Добавить новое задание в список Б в соответствии с заданными приоритетами
4
Выключается ли станок?
НЕТ
ДА
5
Сформировать признак остановки станка, изменить график при необходимости
6
Проверить список текущих заданий Б
7
Есть ли задания ждущие выполнения?
НЕТ
Б
ДА
8
Выбрать наиболее срочное задание
9
Существует ли станок для его выполнения?
НЕТ
ДА
В
10
Имеются ли свободные рабочие?
ДА
НЕТ
Г
11
Выделить на задание рабочего.
Зарегистрировать конец выполнения задания в списке А
Д

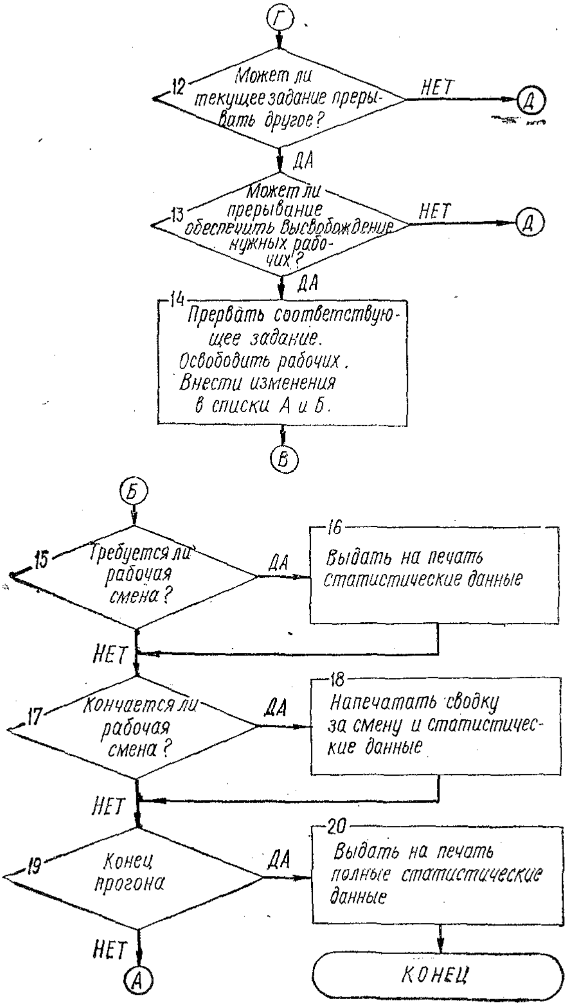

Г
12
Может ли текущее задание прерывать другое?
НЕТ
Д
ДА
13
Может ли прерывание обеспечить высвобождение нужных рабочих?
НЕТ
Д
ДА
14
Прервать соответствующее задание. Освободить рабочих. Внести изменения в списки А и Б.
В
Б
15
Требуется ли рабочая смена?
ДА
16
Выдать на печать статистические данные
НЕТ
17
Кончается ли рабочая смена?
ДА
18
Напечатать сводку за смену и статистические данные
НЕТ
19
Конец прогона
ДА
20
Выдать на печать полные статистические данные
НЕТ
А
КОНЕЦ

Рис. 1.5.

Таблица 1.7

Сводка результатов имитационного моделирования

Статистические данные по станкам

Среднее время беспростойной работы	377,689	Процент времени беспростойной работы	78,685
Минимальная граница	278,944	Максимальная граница	408,563
Среднее время простоев	102,311	Процент времени простоев	21,315
Среднее время ремонта	83,835	Процент времени ремонта	17,466
Минимальная граница	45,655	Максимальная граница	139,464
Среднее время ожидания	18,454	Процент времени ожидания	3,845
Минимальная граница	5,180	Максимальная граница	53,656

Статистические данные по станочникам

Номер станочника	Разряд	Нагрузка, ч	Нагрузка, %	Личное время, ч	Личное время, %	Простои, ч	Простои, %
1	1	287,95	56,46	90,00	17,65	132,03	25,89
2	1	255,71	50,14	90,00	17,65	164,27	32,21
3	2	263,38	51,84	90,00	17,65	156,60	30,71
4	2	208,59	40,90	90,00	17,65	211,39	41,45
5	3	173,57	34,03	90,00	17,65	246,41	48,32
6	3	90,67	17,78	90,00	17,65	329,31	64,57

Статистические данные по заданиям

Номер задания	Тип	Частота появления за смену	Среднее время выполнения	Время на задание, %	Максимальный интервал между заданиями	Процент заданий, выполненных с задержкой	Максимальная длительность задержки	Частота поступления заявок во время выполнения предыдущего задания
1	2	15,45	2,30	18,90	3,90	32,61	6,86	4
2	1	9,79	3,99	20,74	4,97	17,45	4,33	0
3	3	21,25	2,00	22,60	2,46	44,90	18,17	1
4	3	34,42	0,79	14,37	1,49	30,99	17,52	3
5	2	4,62	9,52	23,38	50,82	18,92	39,74	0

Таблица 1.8

Результаты моделирования

Состав бригады	Относительное время простоев	Затраты на рабочую силу в расчете на час, ед. ст.	Потери от простоев в расчете на час, ед. ст.	Суммарные издержки в расчете на час, ед. ст.
Размер группы: 6 станков				
2 Ст, 2 П, 1 Сл	0,280	26,0	50,40	76,40
1 Ст, 2 П, 1 Сл	0,345	21,0	62,10	83,10
3 Ст, 2 П, 1 Сл	0,280	31,0	50,40	81,40

Продолжение табл. 1.8

Состав бригады	Относительное время простоев	Затраты на рабочую силу в расчете на час., ед. ст	Потери от простоев в расчете на час, ед. ст.	Суммарные издержки в расчете на час, ед. ст.
2 Ст, 1 П, 1 Сл	0,320	22,0	57,60	79,60
2 Ст, 3 П, 1 Сл	0,280	30,0	50,40	80,40
3 Ст, 1 П, 1 Сл	0,325	27,0	58,50	85,50
1 Ст, 3 П, 1 Сл	0,300	25,0	54,00	79,00
1 Ст, 2 П, 2 Сл	0,290	29,0	52,20	81,20
2 Ст, 2 П, 2 Сл	0,213	34,0	38,34	72,34
2 Ст, 1 П, 2 Сл	0,286	30,0	51,48	81,48
1 Ст, 3 П, 2 Сл	0,280	33,0	50,40	83,40
3 Ст, 1 П, 2 Сл	0,261	35,0	46,98	81,98
3 Ст, 3 П, 1 Сл	0,255	35,0	45,90	80,90
3 Ст, 2 П, 2 Сл	0,205	39,0	36,90	75,90
2 Ст, 3 П, 2 Сл	0,203	38,0	36,54	74,54
3 Ст, 3 П, 2 Сл	0,195	43,0	35,10	78,10

Прогоны имитационной модели повторялись до получения устойчивой оценки для относительного времени простоев с точностью до 0,05.

Затраты на рабочую силу в расчете на час, ед. ст.:

Станочник (Ст)	5
Подсобный рабочий (П)	4
Слесарь-ремонтник (Сл)	8

Потери от простоев в расчете на станок за час — 30 ед. ст.

2. Теория очередей. В случае, когда потоки заданий являются пуассоновскими, время их обслуживания распределено по экспоненциальному закону, а принятая дисциплина обслуживания — «первым пришел — первым обслуживается», для решения указанной задачи возможно применить аппарат теории массового обслуживания. К сожалению, большинство из этих предположений на практике не выполняются. Кроме того, такой подход не позволяет исследовать другие дисциплины обслуживания заданий.

Указанные недостатки и ограничения не позволяют применить вышеперечисленные методы для решения поставленной задачи. Однако они были использованы для получения первого приближения оценок численности рабочих.

Описание моделирующей программы. Для решения задачи была разработана имитационная модель, входные и выходные характеристики которой приведены на рис. 1.4. Вход включает в себя данные о выполняемых заданиях и конфигурации человеко-машинной системы. На выходе определяются продолжительность работы, ремонта и вынужденных простоев для каждого станка, нагрузка и степень использования каждого станочника и накапливаются статистические данные, получаемые при работе модели.

На рис. 1.5 приведена структурная схема моделирующей программы.

Модель работает в различных режимах, обеспечивающих различный уровень детализации выходных данных.

В табл.1.7 приведены все выходные параметры по результатам моделирования. Полученная на модели сводка по станкам содержит средние значения и диапазон изменения времени работы, ремонта и вынужденного простоя станков за моделируемый период времени. Сводка по станочникам показывает рабочую нагрузку, время работы и время простоев каждого из них. Сводка по заданиям дает частоту появления, временные задержки и вынужденные простои по каждому заданию.

Приоритеты для обслуживания заданий назначались следующим образом. Относительные приоритеты определяются по данным о резервах времени моделирующей программой. Задание с минимальным резервом должно выдаваться оператору первым. Среди заданий с равными резервами времени первый приоритет имеет задание с *наименьшим временем выполнения.* При поиске путей сокращения времени выполнения заданий эти правила можно усложнить и проанализировать более сложные выходные результаты.

Данная имитационная модель была применена для решения задачи определения оптимального числа рабочих для обслуживания группы из шести станков. В результате анализа ожидаемого интервала поступления заданий и времени их выполнения была определена рабочая нагрузка и сделан вывод о целесообразности использования от трех до шести рабочих для обслуживания этой группы станков. Применение теории массового обслуживания позволило найти оценку оптимального числа рабочих: три станочника или три подсобных рабочих и два слесаря-ремонтника. При этом общее ожидаемое время, в течение которого станки не работают, составило по оценке 51%.

Для уточнения требуемой численности рабочих для обслуживания станков была использована имитационная модель. Имитационное моделирование выполнялось для групп из трех и шести станков. Результаты анализа приведены в табл. 1.8. Легко видеть, что оптимальная численность работников такова: два станочника (Ст), два подсобника (П) и два слесаря (Сл), при этом относительное время простоя 21,3%, а суммарные издержки в час составляют 72, 34 ед. ст.

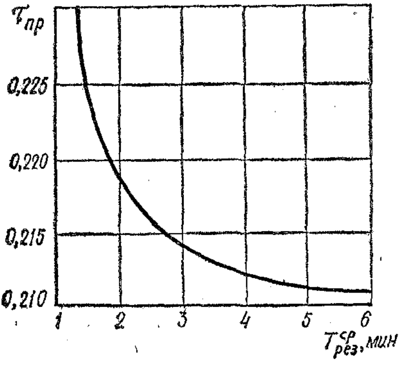


Рис. 1.6.

Имитационная модель позволила не только определить оптимальную численность рабочих, но и провести исследование различных факторов и организационных мероприятий, влияющих на критерий эффективности системы. В частности, требовалось выяснить, какой эффект может быть получен, если снять ограничение, не позволяющее ремонтникам помогать в работе станочникам. С этой целью в модель было введено условие, что ремонтник, не занятый непосредственно ремонтом станков, помогает станочнику. В результате простои снизились с 21,3 до 19,7%, а затраты уменьшились на 23 единицы в смену.

Требовалось выяснить влияние изменения резерва времени на его выполнение. Очевидно,уменьшение резерва времени $T^{ср}_{рез}$ может повлечь за собой увеличение времени простоев $\tau_{пр}$, и наоборот. В имитационной модели были проведены прогоны при различных резервах времени. На рис. 1.6 приведен график соответствующей зависимости, из которого следует, что увеличение резервов *времени свыше 3 мин уже почти* не изменяет вынужденных простоев.

Таким образом, использование имитационной модели в данном примере позволило не только ответить на основной вопрос задачи, но и провести анализ влияния различных факторов на результаты решения. Проведенные эксперименты подтвердили достаточно высокую гибкость имитационной модели. Данная модель может применяться и в задачах, непосредственно не связанных с обслуживанием станков, для анализа систем с конечным числом источников заявок и конечным числом приборов обслуживания.

§ 6. ДЕЛОВЫЕ ИГРЫ КАК МОДЕЛИ

Особой разновидностью моделирования в исследовании операций являются так называемые операционные, или деловые, игры. Деловая игра представляет человеко-машинный процесс моделирования, где человек-руководитель или группа людей участвуют в процессе принятия управленческих решений, а последствия этих решений оцениваются путем многократного проигрывания на ЭВМ.

Деловые игры, таким образом, играют роль тренажа для отработки у будущих организаторов производства умения ориентироваться в сложной производственной обстановке и принимать своевременные и разумные решения в критических ситуациях.

Конструктивными элементами любой деловой игры являются:

1) участники игры, выступающие руководителями структурных хозяйственных единиц, цехов, предприятий и других подсистем организации, называемых «распорядительными центрами» (РЦ);

2) правила, ограничивающие и направляющие проявления интересов соответственно представлениям конструктора игры о моделируемом объекте;

3) информационные базы, или фонд, отражающий состояние и движение ресурсов моделируемой хозяйственной системы и стимулирующий проявление интересов у людей — участников деловой игры.

Очевидно, что замена информационной базы на действительные ресурсы превратит любую игру в натуральный эксперимент со всеми вытекающими отсюда последствиями.

Основными направлениями использования деловых игр являются следующие:

для обучения управляющих и выработки у них умения принимать решения в сложных хозяйственных ситуациях;

для аттестации кадров и проверки их компетентности; такие игры проводятся следующим образом. Подбирается группа участников, выполняющая контрольную роль. С некоторого обусловленного момента контрольные игроки начинают принимать заведомо неэффективные решения. Проверяемый участник должен это заметить и выступить против них, добиваться перемены их политики в игре. Такие игры очень полезны, например, на курсах резерва руководителей той или иной хозяйственной системы;

для исследовательских целей. В этом случае участники игры и их интересы являются той средой, в которой варьируются правила взаимодействия и стратегия распределения ресурсов между хозяйственными единицами.

В такого рода играх можно проверять и совершенствовать принципы хозяйствования, новые виды показателей, документов и т. п.

Полезные результаты такая игра может дать лишь в том случае, если те же люди, которые участвуют в игре, потом на практике будут пользоваться проверяемыми предложениями для отработки игровых методов принятия решений, касающихся разработки плана.

Каждое из перечисленных направлений использования деловых игр имеет свои особенности конструирования, предъявляет свои специфические требования к формированию команд участников, выбору игровой структуры, правил принятия решений, состава и содержания алгоритмов обработки данных.

Для игр, применяемых в реальном планировании, наиболее существенными элементами являются: игровая структура и правила распределения ресурсов и взаимодействия хозяйственных единиц; для игр на аттестацию — выбор участников и алгоритмов обработки данных; для игр исследовательского типа — детализация правил и структура данных; для учебных игр — правила и алгоритмы обработки сведений, используемых в принятии решений.

В качестве примера рассмотрим следующую ситуацию. Имеется некоторый производственный объект, который характеризуется следующими особенностями:

а) определены производственные мощности предприятия, причем динамика их изменения представляет некоторый неконтролируемый процесс вследствие выхода станков из строя;

б) задан месячный план предприятия в виде объемов выпуска по заказам, при этом технологические маршруты деталей считаются известными;

в) длительности детале-операций представляют собой случайные величины, распределенные по известному закону с заданными статистическими характеристиками — средним, дисперсией и т. д.

Процесс функционирования предприятия моделируется на ЭВМ в виде сложной вероятностной системы, причем результаты каждого цикла моделирования — итоги выпуска за день — выводятся на печать и сопоставляются с плановыми показателями. В случае некоторой аварийной ситуации (например, длительной поломки станка) и резкого отставания от плана испытуемый руководитель должен принять некоторое организационное решение, например, о переброске менее загруженных станков на прорыв или об изменении очередности выпускаемых заказов, перекомпоновке плана и т. д. Затем принятое решение реализуется в модели, после чего продолжается весь процесс моделирования. Последствия принятых решений в конечном счете оцениваются по окончательному результату — итогам выполнения плана за месяц. Многократно проиграв процесс функционирования предприятия при участии человека — руководителя, можно дать оценку его профессиональной подготовленности.

В операционной игре может участвовать не один, а несколько испытуемых одновременно. Возможна постановка конфликтной операционной игры, где интересы участников противоположны.

В частности, весьма полезно применение операционных игр при моделировании боевых операций.

Интересный опыт по использованию деловых игр для обучения управленческого персонала и исследовательских целей накоплен в Ленинградском финансово-экономическом институте. Здесь разработаны и успешно используются на факультете повышения квалификации руководящих кадров такие деловые игры, как «ИМПУЛЬС», «ЭПОС», «АСТРА» [16].

Глава 2. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Большинство задач, решаемых методами исследования операций, может быть сформулировано так.

Требуется найти

$$\max f(x_1, \ldots, x_n) \tag{1}$$

при ограничениях вида

$$\begin{gathered} g_1(x_1, \ldots, x_n) \leqslant b_1; \\ g_2(x_1, \ldots, x_n) \leqslant b_2; \\ \cdots\cdots\cdots\cdots \\ g_m(x_1, \ldots, x_n) \leqslant b_m, \end{gathered}$$

где $f(x_1, \ldots, x_n)$ — целевая функция или эффективность системы (например, доход от производства каких-то изделий, стоимость перевозок и пр.); $\mathbf{X} = x_1, \ldots, x_n$ — варьируемые параметры; $g_1(\mathbf{X}), \ldots$ $\ldots, g_m(\mathbf{X})$ — функции, которые задают ограничения на имеющиеся ресурсы.

Среди известных разделов математического программирования наиболее развитым и законченным является линейное программирование (ЛП). Несмотря на требование линейности целевой функции и ограничений, в рамки линейного программирования укладываются задачи распределения ресурсов, управления запасами, сетевого и календарного планирования, транспортные задачи, задачи теории расписаний и т. д.

Рассмотрим некоторые примеры задач линейного программирования.

Определение оптимального ассортимента. Имеется p видов ресурсов в количествах $a_1, a_2, \ldots, a_i, \ldots, a_p$ и q видов изделий. Задана матрица $\mathbf{A} = \| a_{ik} \|$, где a_{ik} характеризует нормы расхода i-го ресурса на единицу k-го изделия ($k = 1, 2, \ldots, q$).

Эффективность выпуска единицы k-го изделия характеризуется показателем c_k, удовлетворяющим условию линейности.

Определить план выпуска изделий (оптимальный ассортимент), при котором суммарный показатель эффективности принимает наибольшее значение.

Количество единиц k-го изделия, выпускаемых предприятием, обозначим x_k. Тогда математическая модель задачи имеет такой вид:

найти
$$\max z = \sum_k c_k x_k, \tag{2}$$

при ограничении

$$\sum_k a_{ik} \leqslant a_i, \quad i = 1, 2, \ \ldots, \ p. \tag{3}$$

Кроме ограничения по ресурсам (3), в модель могут быть введены дополнительные ограничения на планируемый выпуск продукции $x_j \geqslant x_{j0}$, условия комплектности для сборки $x_i : x_j : x_k = b_i : b_j : b_k$ для всех i, j, k и т. д.

Оптимальное распределение взаимозаменяемых ресурсов. Имеются m видов взаимозаменяемых ресурсов a_1, a_2, ..., a_i, ..., a_m, используемых при выполнении n различных работ в объеме b_1, b_2, ..., b_n.

Заданы числа λ_{ij}, указывающие, сколько единиц j-й работы можно получить из единицы i-го ресурса, а также c_{ij} — затраты при изготовлении единицы j-го продукта из i-го ресурса.

Требуется распределить ресурсы по работам таким образом, чтобы суммарная эффективность была наибольшей (или суммарные затраты — наименьшими).

Данная задача называется *общей распределительной задачей.*

Количество единиц i-го ресурса, которое выделено для выполнения работ j-го вида, обозначим x_{ij}.

Математическая модель задачи такова:

найти

$$\min \sum_i \sum_j c_{ij} x_{ij} \tag{4}$$

при ограничениях

$$\sum_{i=1}^{n} \lambda_{ij} x_{ij} \geqslant b_j, \quad j = 1, \ \ldots, \ n; \tag{5}$$

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, \ \ldots, \ m. \tag{6}$$

Ограничение (5) означает, что план всех работ должен быть выполнен полностью, а ограничение (6) — что ресурсы должны быть израсходованы целиком.

В качестве примера такой задачи может служить известная задача о распределении самолетов по авиалиниям.

Задача о смесях. Имеется p компонентов ($i = 1, 2, ..., p$), при сочетании которых в разных пропорциях получают различные смеси. В каждый компонент, а следовательно, и в смесь входит q веществ. Количество k-го вещества ($k = 1, 2, ..., q$), входящее в состав единицы i-го компонента и в состав единицы смеси, обозначим соответственно a_{ik} и a_k. Полагают, что a_k зависит от a_{ik} линейно, т. е. если смесь состоит из x_1 единиц первого компонента, x_2 — единиц второго компонента и т. д., то

$$a_k = \sum_i a_{ik} x_i.$$

Задано p величин c_i, характеризующих цену, массу или калорийность единицы i-го компонента, и q величин b_k, указывающих минимально необходимое процентное содержание k-го вещества в смеси.

Необходимо определить состав смеси, при котором суммарная характеристика (цена, масса или калорийность) окажется наилучшей.

Обозначим через $x_1, x_2, \ldots, x_p$ величину компонента p-го вида, входящего в смесь.

Математическая модель имеет такой вид:

найти

$$\min \sum_{i=1}^{p} c_i x_i \tag{7}$$

при условии

$$\sum_{i=1}^{p} a_{ik} x_i \geqslant b_k \quad (k = 1, 2, \ldots, q). \tag{8}$$

Условие (8) означает, что процентное содержание k-го вещества в единице смеси должно быть не меньше величины b_k.

К этой же модели сводится, например, задача определения оптимального рациона кормления скота [26].

Задача о раскрое материалов. На раскрой поступает m различных материалов. Требуется изготовить из них k разных комплектующих изделий в количествах, пропорциональных $b_1, b_2, \ldots, b_k$ (условие комплектности).

Пусть каждая единица j-го материала ($j = 1, 2, \ldots, m$) может быть раскроена n различными способами, так что при использовании i-го способа раскроя ($i = 1, 2, \ldots, n$) получится $a_{ij}^{(k)}$ единиц k-го изделия.

Определить план раскроя, обеспечивающий максимальное количество комплектов, если известно, что объем запаса j-го материала равен a_j единиц.

Количество единиц j-го материала, раскраиваемых i-м способом, обозначим x_{ij}, а количество изготавливаемых комплектов изделий — x.

Математическая модель задачи такова:

найти

$$\max x \tag{9}$$

при условиях

$$\sum_{i=1}^{n} x_{ij} \leqslant a_j, \tag{10}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} a_{ij}^{(k)} = b_k x. \tag{11}$$

Условие (10) означает ограничение запаса j-го материала, а (11) — условие комплектности.

Оптимальные балансовые модели. Рассматривают n-отраслевую балансовую модель с постоянными технологическими коэффициентами, задаваемыми матрицей затрат $\mathbf{A} = \| a_{ij} \|$, где a_{ij} — затраты продукции i-й отрасли на производство единицы продукции j-й отрасли. Производственные мощности i-й отрасли ограничивают ее валовой выпуск величиной d_i ($i = 1, \ldots, n$) и цена конечного продукта i-й отрасли составляет c_i.

Определить оптимальный валовой выпуск продукции каждой отрасли, при котором будет достигнут максимальный суммарный выпуск конечного продукта в денежном выражении.

Обозначим вектор валовой продукции всех отраслей $\mathbf{X} = [x_1, x_2, \ldots \ldots, x_n]$, а вектор конечного продукта $\mathbf{Y} = [y_1, y_2, \ldots, y_n]$. Тогда y_i — объем продукта i-й отрасли, идущего на накопление.

Между векторами **X** и **Y** существует следующая связь:

$$\mathbf{X} = \mathbf{AX} + \mathbf{Y}, \tag{12}$$

где **AX** — продукт, расходуемый на потребление.

Отсюда $\mathbf{Y} = \mathbf{X}[\mathbf{E} - \mathbf{A}]$, $\mathbf{X} = [\mathbf{E} - \mathbf{A}]^{-1}\mathbf{Y}$.

Математическая модель задачи имеет следующий вид: найти

$$\max \mathbf{C}^T\mathbf{Y} = \sum_i c_i y_i \tag{13}$$

при условиях

$$\mathbf{X} = [\mathbf{E} - \mathbf{A}]^{-1}\mathbf{Y} \leqslant \mathbf{d}, \quad \mathbf{Y} \geqslant 0.$$

Кроме того, в задаче могут быть дополнительно указаны следующие ограничения, налагаемые на конечные продукты, например: *а*) $y_1 : : y_2 : \ldots : y_n = b_1 : b_2 : \ldots : b_n$ — условие комплектности; *б*) $d_i \geqslant \geqslant y_i \geqslant b_i$ — условие ограниченности выпуска конечного продукта.

§ 1. ПОСТАНОВКА ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ И ИССЛЕДОВАНИЯ ЕЕ СТРУКТУРЫ

Задачу линейного программирования можно сформулировать так. Найти

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} c_i x_i \tag{1.1}$$

при условии:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leqslant b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leqslant b_2; \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leqslant b_m \end{aligned} \tag{1.2}$$

и

$$x_1 \geqslant 0, \; x_2 \geqslant 0, \; \ldots, \; x_n \geqslant 0. \tag{1.3}$$

Ограничения (1.3) называют условиями неотрицательности.

В данном случае все условия имеют вид неравенств. Иногда они могут быть смешанными, т. е. неравенства и равенства:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leqslant b_1; \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leqslant b_m; \\ a_{m+1,1}x_1 + a_{m+1,2}x_2 + \cdots + a_{m+1,n} &= b_{m+1}; \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n &= b_k. \end{aligned} \tag{1.4}$$

Если все ограничения задачи ЛП заданы в виде строгих равенств:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2; \\ \dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m, \end{array} \tag{1.5}$$

то данная форма называется *канонической*.

В матричной форме задачу ЛП записывают следующим образом. Найти

$$\max \mathbf{c}^T \mathbf{x} \tag{1.6}$$

при условии

$$\begin{array}{c} \mathbf{A}\mathbf{x} \leqslant \mathbf{b}; \\ \mathbf{x} \geqslant 0, \end{array} \tag{1.7}$$

где $\mathbf{A}$ — матрица ограничений размером $(m \times n)$, $\mathbf{b}^{(m \times 1)}$— вектор-столбец свободных членов, $\mathbf{x}^{(n \times 1)}$ — вектор переменных, $\mathbf{c}^T = [c_1, c_2, \ldots, c_n]$ — вектор-строка коэффициентов целевой функции.

Допустимым множеством решений задачи (1.1) — (1.3) *называется множество* R (x) *всех векторов* x, *удовлетворяющих условиям* (1.2) и (1.3).

Очевидно множество R (x) представляет собой выпуклое многогранное множество или выпуклый многогранник.

Решение $\mathbf{x}_0$ называется *оптимальным*, если для него выполняется условие $\mathbf{c}^T\mathbf{x}_0 \geqslant \mathbf{c}^T\mathbf{x}$, для всех $\mathbf{x} \in R$ (x).

Отметим, что поскольку min f(x) эквивалентен max $[-f(\mathrm{x})]$, то задачу ЛП всегда можно свести к эквивалентной задаче максимизации.

Геометрическая интерпретация задачи ЛП. Рассмотрим следующий пример.

Найти max $(2x_1 + 5x_2) = z$ при условиях $x_1 \leqslant 400$, $x_2 \leqslant 300$, $x_1 + x_2 \leqslant 500$, $x_1 \geqslant 0$, $x_2 \geqslant 0$.

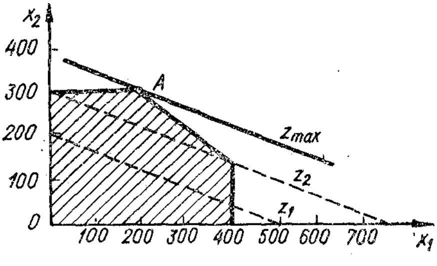


Рис. 2.1.

Каждое из этих неравенств-ограничений определяет полуплоскости, пересечение которых дает многоугольник, который заштрихован на рис. 2.1. Этот многоугольник (выпуклый многогранник) и представляет собой допустимое множество решений R задачи ЛП.

Теперь рассмотрим целевую функцию $f(x_1, x_2) = 2x_1 + 5x_2$.

Пусть $f(x_1, x_2) = 1000 = z_1$. График уравнения $2x_1 + 5x_2 = 1000$ представляет собой прямую с отрезками на осях $x_1 = 500$ единиц, а $x_2 = 200$ единиц.

При $f(x_1, x_2) = 1500$ получим прямую z_2, имеющую уравнение

$$\frac{2x_1}{1500} + \frac{5x_2}{1500} = \frac{x_1}{750} + \frac{x_2}{300} = 1.$$

Прямая z_2 параллельна прямой z_1, но расположена выше ее. Двигая прямую вверх параллельно самой себе, приходим к такому положению $z_{\max}$, когда прямая и множество R будут иметь только одну общую точку A. Очевидно, что точка A ($x_1 = 200$; $x_2 = 300$) — оптимальное решение, так как она лежит на прямой с максимально возможным значением $z_{\max}$. Заметим, что эта точка оказалась крайней точкой множества R.

При векторной форме записи ограничения задачи ЛП записывают так:

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_n x_n \leqslant \mathbf{b}, \tag{1.8}$$

где

$$\mathbf{A}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \cdots \\ a_{m1} \end{bmatrix}; \ \mathbf{A}_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \cdots \\ a_{m2} \end{bmatrix}; \ \ldots; \ \mathbf{A}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \cdots \\ a_{mn} \end{bmatrix}; \ \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix}.$$

Векторы $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$ называются требованиями задачи. Рассмотрим допустимое множество R в пространстве данных векторов. Так как в формуле (1.8) $x_i \geqslant 0$ ($i = 1, 2, \ldots, n$), то все положительные комбинации векторов $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$ образуют конус (см. приложение 1). Поэтому вопрос о существовании допустимого решения равнозначен вопросу о принадлежности вектора $\mathbf{b}$ к этому конусу. Поскольку $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$ m-мерные векторы ($n > m$), то среди них всегда найдется m линейно независимых векторов, образующих базис m-мерного пространства и содержащих конус, образованный векторами $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$.

Поэтому справедливо следующее утверждение. Если задача ЛП содержит n переменных и m ограничений ($n > m$), записанных в форме неравенств, не считая ограничений неотрицательности $x_j \geqslant 0$, то в оптимальное решение входит не более, чем m, ненулевых компонент вектора $\mathbf{x}$.

Расширенная форма задачи ЛП. Для решения задач ЛП необходимо уметь переходить от ограничений в форме неравенств к равенствам. Для этого вводят свободные переменные $x_{n+1} \geqslant 0$, $x_{n+2} \geqslant 0$, ..., $x_{n+m} \geqslant 0$, которые превращают неравенства в равенства. В таком виде задачу ЛП называют *расширенной* и записывают так:

найти

$$\begin{aligned} \max f(x_1, x_2, \ldots, x_n) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + \\ + 0 \cdot x_{n+1} + \cdots + 0 \cdot x_{n+m} \end{aligned} \tag{1.9}$$

при ограничениях

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + 1 \cdot x_{n+1} + 0 \cdot x_{n+2} + \\ + \cdots + 0 \cdot x_{n+m} = b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + 0 \cdot x_{n+1} + 1 \cdot x_{n+2} + \\ + \cdots + 0 \cdot x_{n+m} = b_2; \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{aligned}$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + 0 \cdot x_{n+1} + 0 \cdot x_{n+2} + \\ + \dots + 1 \cdot x_{n+m} = b_m.$$

В матричной записи она имеет следующий вид.
Найти

$$\max \mathbf{c}^T \mathbf{x}$$

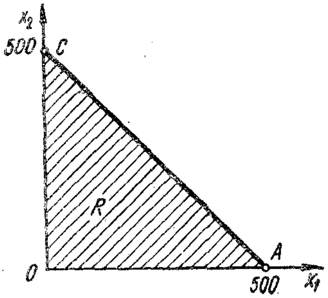


Рис. 2.2.

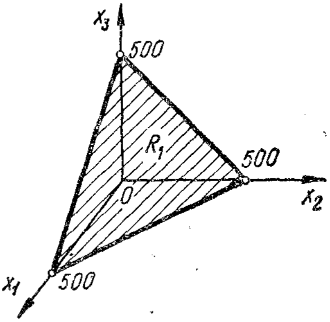


Рис. 2.3.

при ограничениях

$$\mathbf{A}^{m\times n}\mathbf{x}_1 + \mathbf{E}^{m\times m}\mathbf{x}_2 = \mathbf{b};$$

$$\mathbf{E}^{m\times m} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 1 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} x_{n+1} \\ x_{n+2} \\ \dots\dots \\ x_{n+m} \end{pmatrix}. \tag{1.10}$$

Наконец, векторная запись расширенной формы задачи ЛП: найти $\max \mathbf{c}^T \mathbf{x}$ при ограничениях

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \dots + \mathbf{A}_n x_n + \mathbf{A}_{n+1} x_{n+1} + \dots + \mathbf{A}_{n+m} x_{n+m} = \mathbf{b}. \tag{1.11}$$

Пусть R и R_1 — допустимое множество решений соответственно исходной и расширенной задач. Тогда любой точке допустимого множества R_1 соответствует единственная точка R и наоборот.

Установим отношение между элементами этих множеств R и R_1:

исходная задача	расширенная задача
$x_1 + x_2 \leqslant 500$	$x_1 + x_2 + x_3 = 500.$

На рис. 2.2 и 2.3 графически изображены допустимые множества решений обеих задач. *Видно, что треугольник OCA допустимого множества R есть проекция допустимого множества R_1 на подпространство $x_1 0 x_2$.*

В общем случае допустимое множество R решений исходной задачи есть проекция допустимого множества R_1 решений расширенной задачи на подпространство исходных переменных ($x_1 0 x_2$).

Пример 2.1. Рассмотрим задачу, для которой исходные ограничения и ограничения в расширенной форме имеют соответственно вид:

$$1x_1 + 1x_2 \leqslant 5; \quad 1x_1 + 1x_2 + 1x_3 = 5;$$

$$\frac{3}{2}x_1 + 1x_2 \leqslant 6; \quad \frac{3}{2}x_1 + 1x_2 + 0x_3 + 1x_4 = 6;$$

$$x_1 \geqslant 0; \quad x_2 \geqslant 0; \quad x_1, x_2, x_3, x_4 \geqslant 0.$$

Обозначим

$$A_1 = \begin{bmatrix} 1 \\ \frac{3}{2} \end{bmatrix}; \quad A_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad A_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad A_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad A_0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}.$$

Из векторов A_1, A_2 можно составить шесть базисов: $\{A_1, A_2\}$; $\{A_1, A_3\}$; $\{A_4, A_3\}$ и т. д.

Для каждой из этих матриц находим обратные матрицы B^{-1}. Умножив B^{-1} на A_0, получим базисные решения:

$$(x_1 = 2,\ x_2 = 3); \quad (x_1 = 4,\ x_3 = 1); \quad \left(x_1 = 5,\ x_4 = -\frac{3}{2}\right);$$

$$(x_2 = 6,\ x_3 = -1); \quad (x_2 = 5,\ x_4 = 1); \quad (x_3 = 5,\ x_4 = 6).$$

3 Допустимые базисные решения

Пусть ограничения задачи ЛП заданы в форме равенств

$$A^{(m\times n)}x^{(n\times 1)} = b^{(m\times 1)}. \tag{1.12}$$

Предположим, что $m < n$ и ранг матрицы A равен m. Выберем из матрицы $A = [A_1 A_2 \ldots A_n]$ m линейно-независимых столбцов, которые обозначим через $B^{(m\times m)}$. Очевидно, матрица B образует базис системы. Совокупность оставшихся столбцов матрицы A обозначим через D. Тогда $A = [B, D]$.

Совокупность переменных, связанных с матрицей B, обозначим через x_B, а связанных с матрицей D — через x_D. Тогда

$$Ax = Bx_B + Dx_D. \tag{1.13}$$

Так как B—невырожденная квадратная матрица, то существует обратная к ней B^{-1}. Умножив обе части (1.13) на B^{-1}, получим

$$B^{-1}Bx_B + B^{-1}Dx_D = B^{-1}b.$$

Отсюда

$$x_B = B^{-1}b - B^{-1}Dx_D. \tag{1.14}$$

Переменные x_B — базисные, а x_D — небазисные.

Соотношения (1.14) определяют полное множество решений системы линейных уравнений (1.12). В развернутом виде оно может быть записано так

$$\left.\begin{aligned} x_i &= a_{i0} - \sum_{j\in J_{\text{неб}}} a_{ij}x_j, \quad i \in I_{\text{б}} \\ x_j &= \alpha_j, \end{aligned}\right\} \tag{1.15}$$

где $I_{\text{б}}$ — множество индексов базисных векторов; $J_{\text{неб}}$ — множество индексов небазисных векторов; a_{i0} — i-я — компонента вектора $B^{-1}b$; a_{ij} $(j \in J_{\text{неб}})$ — i-я — строка матрицы $B^{-1}D$. В (1.15) переменные x_j могут принимать произвольные значения α_j.

Если положить для всех небазисных переменных нулевые значения, то получим базисное решение системы (1.12). Очевидно, в этом случае $x_i = a_{i0}$, $i \in I_б$.

Если базисное решение удовлетворяет условию неотрицательности, то оно называется допустимым (или сокращенно д. б. р.).

Если среди компонент x_i $(i \in I_б)$ нет нулевых, то базисное решение называется *невырожденным.*

Основные теоремы линейного программирования

В этом разделе мы познакомимся с некоторыми важными теоретическими вопросами, которые лежат в основе методов решения задач линейного программирования.

Справедлива следующая основная теорема линейного программирования, устанавливающая место нахождения оптимальных решений.

Теорема 2.1. *Если целевая функция принимает максимальное значение в некоторой точке допустимого множества R_1, то она принимает это значение в крайней точке R_1. Если целевая функция принимает максимальное значение более, чем в одной крайней точке, то она принимает это же значение в любой их выпуклой комбинации.*

Будем считать, что R_1 — выпуклый многогранник.

Доказательство основано на следующей лемме: если R_1 — замкнутое ограниченное выпуклое множество, имеющее конечное число крайних точек, то любая точка $\mathrm{x} \in R_1$ может быть представлена в виде выпуклой комбинации крайних точек R_1.

(Доказательство леммы см. в приложении 1.)

Обозначим крайние точки через $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m$, а точку, в которой $f(\mathrm{x})$ достигает максимума, через $\mathrm{x}_{\text{опт}}$.

Предположим, что $\mathrm{x}_{\text{опт}}$ — не крайняя точка множества. Тогда по лемме

$$\mathrm{x}_{\text{опт}} = \sum_{i=1}^{m} k_i \mathrm{x}_i, \quad \sum_{i=1}^{m} k_i = 1, \; k_i \geqslant 0, \; i = \overline{1, m}.$$

Так как $\mathrm{x}_{\text{опт}}$ — оптимальное решение, то $\mathrm{c}^T\mathrm{x}_{\text{опт}} \geqslant \mathrm{c}^T\mathrm{x}$. Функция $f(\mathrm{x})$ — линейная, поэтому $f\left(\sum\limits_{i=1}^{m} k_i \mathrm{x}_i\right) = \sum\limits_{i=1}^{m} k_i f(\mathrm{x}_i)$. Поэтому

$$\mathrm{c}^T\mathrm{x}_{\text{опт}} = \sum_{i=1}^{m} \mathrm{c}^T k_i \mathrm{x}_i \leqslant \sum_{i=1}^{m} k_i \mathrm{c}^T \mathrm{x}_r = \mathrm{c}^T\mathrm{x}_r \sum_{i=1}^{m} k_i = \mathrm{c}^T\mathrm{x}_r, \tag{1.16}$$

где через $\mathrm{c}^T\mathrm{x}_r$ обозначено $\max\limits_{1 \leqslant i \leqslant m} \{\mathrm{c}^T \mathrm{x}_i\}$. Итак,

$$\mathrm{c}^T\mathrm{x}_{\text{опт}} \leqslant \mathrm{c}^T\mathrm{x}_r. \tag{1.17}$$

Однако $\mathrm{x}_{\text{опт}}$ — оптимальное решение, и потому

$$\mathrm{c}^T\mathrm{x}_{\text{опт}} \geqslant \mathrm{c}^T\mathrm{x}_r. \tag{1.18}$$

Сравнив (1.18) с (1.17), получим $\mathrm{c}^T\mathrm{x}_r = \mathrm{c}^T\mathrm{x}_{\text{опт}}$, т. е. существует по крайней мере одна крайняя точка x_r, где целевая функция принимает максимальное значение.

Итак, первая часть теоремы доказана. Перейдем к доказательству второй части. Допустим, что оптимальные решения находятся в крайних точках $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s$. Тогда их произвольная выпуклая комбинация определится как

$$\mathbf{x}^* = \sum_{i=1}^{s} \mathbf{x}_i k_i, \tag{1.19}$$

где

$$\sum_{i=1}^{s} k_i = 1, \ k_i \geqslant 0.$$

Найдем значения целевой функции

$$f(\mathbf{x}^*) = \mathbf{c}^T \mathbf{x}^* = \sum_{i=1}^{s} \mathbf{c}^T \mathbf{x}_i k_i. \tag{1.20}$$

Так как точки $\mathbf{x}_i$, $i = \overline{1, s}$ отвечают оптимальным решениям, то

$$\mathbf{c}^T \mathbf{x}_i = \mathbf{c}^T \mathbf{x}_{\text{опт}}, \text{ для всех } i = \overline{1, s}. \tag{1.21}$$

Подставляя (1.21) в (1.20), получим

$$f(\mathbf{x}^*) = \mathbf{c}^T \mathbf{x}_{\text{опт}} \sum_{i=1}^{s} k_i = \mathbf{c}^T \mathbf{x}_{\text{опт}}. \tag{1.22}$$

Теорема 2.1 доказана.

Из теоремы 2.1 следует, что *при отыскании оптимального решения достаточно просмотреть только крайние точки допустимого множества решений* R_1.

Докажем теперь, что всегда можно найти крайние точки, выделяя только допустимые базисные решения.

Теорема 2.2. *Если существует такое независимое множество m-мерных векторов* $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_K$ $(K \leqslant m)$, *что* $\mathbf{A}_1 x + \mathbf{A}_2 x_2 + \ldots + \mathbf{A}_K x_K = \mathbf{A}_0$ $(x_i \geqslant 0, i = 1, 2, \ldots, m)$, *то n-мерный вектор* $\mathbf{x}^T = (x_1, x_2, x_K, \underbrace{0, \ldots, 0}_{n-K})$ *есть крайняя точка допустимого множества* R_1, *другими словами, каждое допустимое базисное решение соответствует крайней точке* R_1.

Доказательство проводим от противного, т. е. предположим, что $\mathbf{x}$ — не крайняя точка R_1.

Тогда $\mathbf{x} = k\mathbf{x}_1 + (1 - k)\ \mathbf{x}_2$, $0 < k < 1$, где $\mathbf{x}_1$ и $\mathbf{x}_2$ — крайние точки R_1.

Поскольку последние $(n - K)$ компоненты $\mathbf{x}$ по предположению равны нулю, то эти же компоненты векторов $\mathbf{x}_1$ и $\mathbf{x}_2$ также равны нулю. Поэтому

$$\mathbf{A}_1 x_{11} + \mathbf{A}_2 x_{21} + \cdots + \mathbf{A}_K x_{K1} = \mathbf{A}_0; \tag{1.23}$$

$$\mathbf{A}_1 x_{12} + \mathbf{A}_2 x_{22} + \cdots + \mathbf{A}_K x_{K2} = \mathbf{A}_0. \tag{1.24}$$

Вычитая из уравнения (1.23) уравнение (1.24), получим

$$\mathbf{A}_1 (x_{11} - x_{12}) + \mathbf{A}_2 (x_{21} - x_{22}) + \cdots + \mathbf{A}_K (x_{K1} - x_{K2}) = 0. \tag{1.25}$$

Так как $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_K$ — независимы, то $x_{11} - x_{12} = 0$, $x_{21} - x_{22} = 0$, ..., $x_{K1} - x_{K2} = 0$, что возможно только в том случае, если

$\mathbf{x}_1 = \mathbf{x}_2$. Это противоречит предположению, что $\mathbf{x}_1$ и $\mathbf{x}_2$ — разные крайние точки множества R_1.

Поскольку **x** не может быть представлена в виде выпуклой комбинации двух других крайних точек множества R_1, то **x** является крайней точкой множества R_1.

Справедлива также следующая теорема, обратная к теореме 2.2.

Теорема 2.3. *Если* $\mathbf{x}_0^T = (x_1^{(0)}, x_2^{(0)}, \dots, x_k^{(0)}, 0 \dots 0)$ — *крайняя точка допустимого множества решений R, то соответствующее решение* $\mathbf{x}_0$ — *является допустимым базисным решением системы ограничений* (1.12).

Доказательство теоремы несложно и приводится в [24].

Следствие. Используя результаты теорем 2.1 и 2.2, можно сделать вывод, что для *отыскания оптимального решения ЛП-задачи достаточно перебрать лишь допустимые базисные решения.* Этот вывод лежит в основе многих методов решения задач линейного программирования.

Графический метод решения ЛП-задач

Если общее число переменных ЛП-задачи $n = 2$ или она может быть сведена к соответствующей задаче с числом независимых переменных $k = 2$, то такая задача может быть легко решена графическим методом.

Итак, пусть ЛП-задача имеет вид

$$\max f(x_1, x_2) = \max (c_1x_1 + c_2x_2); \tag{1.26}$$

$$\begin{gathered} g_1(x_1, x_2) \leqslant b_1; \\ g_2(x_1, x_2) \leqslant b_2; \\ \dots\dots\dots \\ g_m(x_1, x_2) \leqslant b_m; \end{gathered} \tag{1.27}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0. \tag{1.28}$$

Графический метод решения заключается в следующем.

1. Строим допустимое множество решений R, определяемое (1.27) и (1.28).

2. Далее строим вектор нормали **N** к целевой функции $f(x_1, x_2)$. Его проекция на ось Ox_1 равна kc_1, а на ось Ox_2 — kc_2, где k — произвольный положительный скаляр.

Заметим, что **N** указывает направление возрастания $f(x_1, x_2)$.

3. Перемещаем прямую $f(x_1, x_2) = \text{const}$ в направлении **N** так, чтобы она оставалась перпендикулярной **N** до тех пор, пока эта прямая не выйдет на границу множества R.

При этом возможен один из следующих случаев:

а) $f(x_1, x_2)$ и R будут иметь лишь одну общую точку (крайнюю точку R); эта точка определяет единственное оптимальное решение;

б) $f(x_1, x_2)$ и R имеют целое множество общих точек, это будет в том случае, когда вектор **N** окажется нормален к соответствующей грани множества R, данное множество общих точек представляет собой множество оптимальных решений задачи;

в) прямая $f(x_1, x_2) = \mathrm{const}$ не выходит на границу множества R, сколько бы ее не перемещали (это будет в случае, если множество R — неограниченно), тогда целевая функция $f(x_1, x_2)$ оказывается также неограниченной.

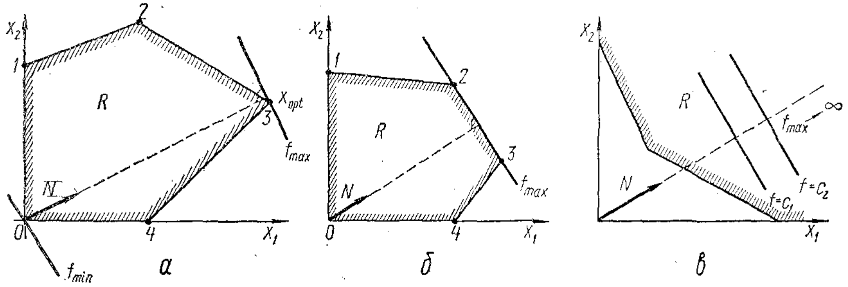


Рис. 2.4.

Соответствующие случаи иллюстрируются рис. 2.4, *а*, *б*, *в*.

Заметим, что при решении задачи минимизации $f(x_1, x_2)$ перемещают в направлении, противоположном **N**.

Рассмотрим теперь общий случай ЛП-задачи:

$$\max \sum_{j=1}^{n} c_j x_j;$$

$$\sum_{j} a_{ij} x_j = b_i, \quad i = \overline{1, m}; \; x_j \geqslant 0.$$

Если $k = n - r = 2$, где r — ранг матрицы **A**, то данная задача может быть также решена графически.

Действительно, на основании общих соотношений (1.15) все базисные переменные x_i могут быть выражены через две небазисные переменные x_{j_1} и x_{j_2}:

$$\begin{aligned} x_1 &= \varphi_1(x_{j_1}, x_{j_2}) \geqslant 0; \\ x_2 &= \varphi_2(x_{j_1}, x_{j_2}) \geqslant 0; \\ &\dots\dots\dots\dots \\ x_r &= \varphi_r(x_{j_1}, x_{j_2}) \geqslant 0. \end{aligned} \tag{1.29}$$

Подставляя в целевую функцию выражения (1.29), мы придем к следующей задаче:

$$\max f(x_1, x_2, \ldots, x_n) = \max f(\varphi_i(x_{j_1}, x_{j_2})_{i \in I_б}, x_{j_1}, x_{j_2}) \tag{1.30}$$

при условиях (1.29) и

$$x_{j_1} \geqslant 0, \; x_{j_2} \geqslant 0. \tag{1.31}$$

Как видим, задача *(1.30)* легко решается графически.

§ 2. СИМПЛЕКС-МЕТОД

Симплекс-метод, известный в отечественной литературе под названием *«метод последовательного улучшения плана»*, впервые был разработан Данцигом в 1947 г. [9, 24, 57]. Этот метод позволяет переходить от одного допустимого базисного решения к другому, причем так, что значения целевой функции непрерывно возрастают. В результате оптимальное решение находят за конечное число шагов. Алгоритмы симплекс-метода позволяют также установить, является ли задача ЛП разрешимой.

Запишем ограничения задачи ЛП в таком виде:

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_{n+m} x_{n+m} = \mathbf{A}_0.$$

Пусть $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$ — множество линейно независимых векторов. Тогда уравнение

$$\mathbf{A}_1 x_1^* + \mathbf{A}_2 x_2^* + \cdots + \mathbf{A}_m x_m^* = \mathbf{A}_0 \tag{2.1}$$

определяет базисное решение $x_1^*, x_2^*, \ldots, x_m^*$.

Предположим, что это решение допустимо, т. е. $x_1^* \geqslant 0, \ldots, x_m^* \geqslant 0$. Базис $\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m\}$ образует m-мерное пространство, а потому каждый из векторов $\mathbf{A}_{m+1}, \mathbf{A}_{m+2}, \ldots, \mathbf{A}_{m+n}$ единственным образом выражается через этот базис.

Если $\mathbf{A}_r$ не входит в базис, то

$$\mathbf{A}_r = x_{1r}\mathbf{A}_1 + x_{2r}\mathbf{A}_2 + \cdots + x_{mr}\mathbf{A}_m, \tag{2.2}$$

где x_{ir} — соответствующие коэффициенты ($i = 1, 2, \ldots, m$).

Теперь предположим, что хотя бы одна из величин x_{ir} больше нуля. Решение уравнения

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_m x_m + \mathbf{A}_r x_r = \mathbf{A}_0 \tag{2.3}$$

обозначим как $\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m, x_r\}$.

Тогда очевидно

$$\mathbf{A}_1 \tilde{x}_1 + \mathbf{A}_2 \tilde{x}_2 + \cdots + \mathbf{A}_m \tilde{x}_m + \mathbf{A}_r x_r = \mathbf{A}_0. \tag{2.4}$$

Умножив уравнение (2.2) на x_r и вычтя полученное уравнение из уравнения (2.1), получим

$$\mathbf{A}_1 (x_1^* - x_r x_{1r}) + \mathbf{A}_2 (x_2^* - x_r x_{2r}) + \cdots + \mathbf{A}_m (x_m^* - x_r x_{mr}) = \mathbf{A}_0 - x_r \mathbf{A}_r. \tag{2.5}$$

Сравнив уравнения (2.5) и (2.4), находим связь нового решения $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m, x_r$ со старым базисным $x_1^*, x_2^*, \ldots, x_m^*$:

$$\tilde{x}_1 = x_1^* - x_r x_{1r}, \quad \tilde{x}_2 = x_2^* - x_r x_{2r}, \quad \ldots, \quad \tilde{x}_m = x_m^* - x_r x_{mr}, x_r. \tag{2.6}$$

Это решение (2.6), во-первых, не будет базисным, так как содержит $m + 1$ переменную, а во-вторых, будет допустимым не для всех значений x_r.

Чтобы новое решение оставалось допустимым, нужно выбрать x_r таким, чтобы ни одна из величин $\tilde{x}_i = x_i^* - x_r x_{ir}$ ($i = 1, 2, \ldots$ $\ldots, m$) не стала меньше нуля. Следовательно, максимальное значение переменной x_r определяется соотношением

$$x_{r\max} = \min_i \left\{ \frac{x_i^*}{x_{ir}} \right\}, \tag{2.7}$$

где $x_{ir} > 0$.

Чтобы сделать новое допустимое решение базисным, нужно удалить одну переменную x_j из решения, а соответствующий вектор $\mathbf{A}_j$ из базиса. В этом случае новый базис будет содержать также m векторов.

Для этого выбираем x_r в соответствии с (2.7). Тогда новое базисное решение имеет вид

$$x_1^* - x_{r\,\max} x_{1r};$$

$$x_2^* - x_{r\,\max} x_{2r};$$

$$x_j \text{ (опущен)};$$

$$x_{r\max},$$

а новый базис $(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{j+1}, \ldots, \mathbf{A}_m, \mathbf{A}_r)$.

Такой переход от одного базиса к другому позволяет находить решения почти для всех задач ЛП. Найдя все крайние точки, можно вычислить целевую функцию и найти экстремальное решение. Однако при больших m и n это практически невозможно. Для перехода к таким допустимым базисным решениям, которым отвечают бо́льшие значения целевой функции, предложен критерий симплекс-метода. Новому базисному решению $\{x^*{}_1 - x_r x_{1r},\ x^*{}_2 - x_r x_{2r}, \ldots,\ x^*{}_m - x_r x_{mr},\ x_r\}$ соответствует следующее значение целевой функции:

$$z_1 = c_1(x_1^* - x_r x_{1r}) + c_2(x_2^* - x_r x_{2r}) + \cdots + c_r x_r =$$

$$= \underbrace{(c_1 x_1^* + c_2 x_2^* + \cdots + c_m x_m^*)}_{z_0} + x_r(c_r - x_{1r}c_1 - \cdots - c_m x_{mr}), \quad (2.8)$$

где z_0 — значение целевой функции, отвечающее исходному базисному решению;

$c_r - c_1 x_{1_r} - c_r x_{2_r} - \ldots - c_m x_{mr}$ *-симплекс-разность* для x_r.

Симплекс-разность вычисляют для каждой переменной, не входящей в базисное решение, и выбирают такую свободную переменную x_r, для которой симплекс-разность максимальна и положительна.

Таким образом, алгоритм симплекс-метода состоит из следующих этапов:

1) находят начальный базис и связанное с ним допустимое решение;

2) вычисляют симплекс-разность для каждой переменной, не входящей в базисное решение;

3) вводят в базис самую выгодную переменную, с максимальной симплекс-разностью, причем ее значение $x_{r\,\max}$ определяют из условия $x_{r\,\max} = \min\limits_i \left\{\dfrac{x_i^*}{x_{ir}}\right\}$ для всех i, для которых $x_{ir} > 0$;

4) выводят из базисного решения переменную x_j, соответствующую $\min\left\{\dfrac{x_i^*}{x_{ir}}\right\} = \dfrac{x_j^*}{x_{jr}}$, а из базиса — вектор $\mathbf{A}_j$;

5) этот цикл повторяют до тех пор, пока симплекс-разности для всех переменных, не входящих в базис, не станут отрицательными — это признак оптимальности базисного решения.

Пример 2.2. Решение задачи симплекс-методом.

Найти $\max\limits_{x_1 x_2} (2x_1 + 5x_2)$ при ограничениях $x_1 \leqslant 400$, $x_2 \leqslant 300$, $x_1 + x_2 \leqslant 500$.

Расширенная форма задачи:

$$\begin{cases} 1x_1 + 0x_2 + 1x_3 + 0x_4 + 0x_5 = 400; \\ 0x_1 + 1x_2 + 0x_3 + 1x_4 + 0x_5 = 300; \\ 1x_1 + 1x_2 + 0x_3 + 0x_4 + 1x_5 = 500. \end{cases}$$

Условия задачи запишем в виде табл. 2.1:

Таблица 2.1

A_1	A_2	A_3	A_4	A_5	A_0
1	0	1	0	0	400
0	1	0	1	0	300
1	1	0	0	1	500

Первый шаг. 1. Выбрав в качестве начального базиса $\{A_3, A_4, A_5\}$, находим первое допустимое базисное решение

$$A_3 x_3^* + A_4 x_4^* + A_5 x_5^* = A_0,$$

откуда $x_3^* = 400$; $x_4^* = 300$; $x_5^* = 500$.

2. Представим каждый из векторов A_1, A_2 в виде линейной комбинации базисных векторов $\{A_3, A_4, A_5\}$:

$$A_1 = A_3 x_{31} + A_4 x_{41} + A_5 x_{51};$$

$$A_2 = A_3 x_{32} + A_4 x_{42} + A_5 x_{52}.$$

Решая эти уравнения, получим

$$x_{31} = 1; \quad x_{41} = 0; \quad x_{51} = 1; \quad x_{32} = 0; \quad x_{42} = 1; \quad x_{52} = 1.$$

3. Находим симплекс-разности соответственно для переменных x_1 и x_2:

$$c_1 - c_3 x_{31} - c_4 x_{41} - c_5 x_{51} = c_1 = 2;$$

$$c_2 - c_3 x_{32} - c_4 x_{42} - c_5 x_{52} = c_2 = 5.$$

Так как $c_2 > c_1$, вводим переменную x_2 в базис.

4. Определим, какая переменная выводится из базиса.

Новые значения переменных следующие:

$$x_3 = x_3^* - x_2 x_{32} = x_3^*; \quad x_4 = x_4^* - x_2 x_{42} = 300 - 1x_2;$$

$$x_5 = x_5^* - x_2 x_{52} = 500 - 1x_2.$$

$$\max x_2 = \min \left\{ \frac{x_3^*}{x_{32}}, \frac{x_4^*}{x_{42}}, \frac{x_5^*}{x_{52}} \right\} = \left\{ \frac{400}{0}; \frac{300}{1}; \frac{500}{1} \right\} = 300.$$

Итак, x_2 вводится в базис со значением $x_2^* = 300$, переменная x_4 выводится из базисного решения, а вектор A_4 — из базиса.

Второй шаг. 1. Новый базис $\{A_2, A_3, A_5\}$, соответствующее базисное решение: $x_2^* = 300$; $x_3^* = 400$; $x_5^* = 500 - 300 \cdot 1 = 200$.

Представим каждый из векторов A_1, A_4, не вошедших в базис, в виде линейной комбинации A_2, A_3, A_5. Так как вектор A_4 был выведен из базиса, рассмотрим только вектор A_1.

Составим уравнение

$$A_2 x_{21} + A_3 x_{31} + A_5 x_{51} = A_1,$$

откуда $x_{21} = 0$; $x_{31} = 1$; $x_{51} = 1$.

2. Находим симплекс-разность для x_1:

$$c_1 - c_2x_{21} - c_3x_{31} - c_5x_{51} = c_1 - 0 - 0 - 0 = c_1 = 2 > 0.$$

Итак, вводим переменную x_1 в базисное решение.

3. Определим, какую переменную следует вывести из базисного решения:

$$x_2 = x_2^* - x_1x_{21} = 300 - 0x_1; \quad x_3 = x_3^* - x_1x_{31} = 400 - 1x_1;$$

$$x_5 = x_5^* - x_1x_{51} = 200 - 1x_1.$$

$$x_1^* = \max x_1 = \min \left\{\frac{300}{0}; \ \frac{400}{1}; \ \frac{200}{1}\right\} = 200.$$

Следовательно, выводим переменную x_5 и вектор A_5 из базиса.

Третий шаг. 1. Новый базис $\{A_1, A_2, A_3\}$, соответствующее базисное решение: $x_1^* = 200$; $x_2^* = 300$; $x_3^* = 400 - x_1x_{31} = 400 - 200 \cdot 1 = 200$.

Так как вектор A_5 был только что выведен из базиса, то рассмотрим только вектор A_4: $A_4 = A_1x_{14} + A_2x_{24} + A_3x_{34}$, откуда $x_{24} = 1$, $x_{14} = -1$, $x_{34} = 1$.

2. Найдем симплекс-разность для переменной x_4:

$$c_4 - c_1x_{14} - c_2c_{24} - c_3x_{34} = 0 - 2(-1) - 5 \cdot 1 = -3 < 0.$$

Поскольку симплекс-разность отрицательна, то данное базисное решение — оптимальное: $x_1^* = 200$; $x_2^* = 300$.

Переменная $x_3^* = 200$ является свободной и потому в окончательном решении опущена.

Рассмотренный выше симплекс-метод вполне пригоден для ручного счета, но плохо приспособлен для программирования и решения задач на ЭЦВМ. Потребовалась его рационализация как по форме представления данных, так и в способе организации вычислений, чтобы сделать симплекс-метод пригодным для машинного счета. С этой целью разработан *табличный вариант симплекс-метода, так называемый метод симплекс-таблиц*. Вычислительной основой для метода является *алгоритм полного исключения* [57].

Метод полного исключения. Пусть задана система линейных алгебраических уравнений

$$\sum_{j=1}^{n} a_{ij}x_j = a_{i0}, \qquad i = 1, \ldots, m.$$

В матричной форме данная система уравнений имеет следующий вид:

$$Ax = A_0.$$

Матрица $A_p = [A, A_0]$ *называется расширенной матрицей.* Метод полного исключения Жордана-Гаусса состоит из конечного числа однотипных шагов и заключается в проведении матрицы A к единичному виду. Он основан на следующих двух операциях: 1) одну из строк расширенной матрицы умножают на число, отличное от нуля; 2) из каждой строки расширенной матрицы вычитают одну строку, умноженную на некоторое число.

Каждое из таких элементарных преобразований, называемых преобразованием Гаусса, приводит к новой системе линейных уравнений, которая эквивалентна исходной.

Первый шаг метода полного исключения:

1. Среди элементов **A** выбирают произвольный элемент, отличный от нуля. Его называют *направляющим элементом шага. Строку и столбец,* содержащие направляющий элемент, называют *направляющими.*

2. Все элементы направляющей строки расширенной матрицы делят на направляющий элемент. В результате получают направляющую строку с направляющим элементом, равным единице. Далее из каждой строки матрицы **A** вычитают новую направляющую строку, умноженную на элемент, который расположен на пересечении преобразуемой строки и направляющего столбца.

Матрицу, в которую преобразовалась расширенная матрица A_p после первого шага, обозначим $A_p^{(1)}$. В этой матрице все элементы направляющего столбца, отличные от направляющего элемента, стали нулями. Совокупность элементов первых n столбцов матрицы A_p, лежащих вне направляющей строки и столбца предыдущего шага, называют *главной частью матрицы* $A_p^{(1)}$.

Направляющий элемент второго шага разыскивают среди ненулевых элементов главной части матрицы $A_p^{(1)}$.

Второй и дальнейшие шаги метода проводят аналогично первому шагу, причем последовательные шаги проводят до тех пор, пока имеется возможность выбора направляющего элемента. Если после k-го шага главная часть матрицы $A_p^{(k)}$ не содержит ни одного элемента или содержит только нули, то процесс исключения переменных заканчивается.

Пусть процесс оборвался после шага l. Предположим вначале, что среди строк матрицы $A^{(l)}$ есть такие, которые не были направляющими ни в одном из предыдущих шагов, к примеру, строка с номером i. Тогда, очевидно, $a_{ij} = 0$, $j = 1, 2, \ldots, n$.

Поскольку для любой строки справедливо

$$A^{(i)}x = a_{i0}, \tag{2.9}$$

то уравнение для i-й строки имеет вид:

$$0x_1 + 0x_2 + \cdots + 0x_n = a_{i0}^{(l)}. \tag{2.10}$$

Если $a_{i0}^{(l)} \neq 0$, то уравнение (2.10) противоречиво, и исходная система уравнений неразрешима.

Если $a_{i0}^{(l)} = 0$, то уравнение (2.10) представляет тождество, и i-я строка может быть отброшена.

Перебрав одну за другой строки матрицы $A^{(l)}$, которые не являлись направляющими, либо устанавливают неразрешимость системы уравнений, либо отбрасывают все нулевые строки.

Таким образом, в системе окажется l уравнений. Примем для определенности, что это первые (по порядку) l уравнений.

Тогда полученная система уравнений может быть записана в виде

$$\sum_{j=1}^{n} a_{ij}^{(l)}x_j = a_{i0}^{(l)}, \qquad i = 1, 2, \ldots, l. \tag{2.11}$$

Примем, что i-й направляющей строке соответствует i-й направляющий столбец (вследствие соответствующего выбора направляющего элемента).

Тогда

$$a_{ij}^{(l)} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \qquad j = 1, 2, \ldots, l. \tag{2.12}$$

Следовательно, (2.11) можно записать в виде:

$$x_i = a_{i0}^{(l)} - \sum_{j=l+1}^{n} a_{ij}^{(l)} \cdot x_j, \qquad i = 1, 2, \ldots, l, \tag{2.13}$$

причем переменные x_i $(i = 1, 2, \ldots, l)$ являются базисными, а переменные $(x_j,\ j = l + 1, \ldots, n)$ — небазисными.

При $x_j = 0$ $(j = l + 1, \ldots, n)$ получим одно из базисных решений системы уравнений:

$$x_i = a_{i_0}^{(l)}, \quad i = 1, 2, \ldots, l; \quad x_j = 0, \quad j = l + 1, \ldots, n.$$

Задавая для x_j произвольные значения α_j, получим все множество решений, которое образует $(n - l)$-мерное подпространство решений.

Если x_i-i-я компонента этого решения, то

$$x_i = \begin{cases} a_{i0} - \sum\limits_{j=l+1}^{n} a_{ij}\alpha_j, & \text{если } i = 1, \ldots, l; \\ \alpha_i, & \text{если } i = l + 1, l + 2, \ldots, n. \end{cases} \tag{2.14}$$

Пусть

$$\mathrm{x}_0 = (a_{10}, a_{20}, \ldots, a_{l0}, 0, \ldots, 0)$$

и

$$\mathrm{x}_j = \underbrace{\underbrace{(-a_{1j}, -a_{2j}, \ldots, -a_{lj}, 0, \ldots, 0, 1}_{j}, 0, \ldots, 0)}_{n}, \qquad l < j \leqslant n,$$

тогда общее решение системы линейных уравнений определяется соотношением, аналогичным (2.14):

$$\mathrm{x}_{\text{общ}} = \mathrm{x}_0 + \sum_{j=l+1}^{n} \alpha_j \mathrm{x}_j, \tag{2.15}$$

где x_0 — базисное решение исходной системы; $\sum\limits_{j=l+1}^{n} \alpha_j \mathrm{x}_j$ — полное решение соответствующей однородной системы уравнений, т. е. при $\mathrm{A}_0 = 0$.

Расширенную матрицу условий после k-й итерации обозначим

$$\mathrm{A}_\mathrm{p}^{(k)} = [a_{i0}^{(k)}, a_{i1}^{(k)}, \ldots, a_{in}^{(k)}], \qquad i = 1, 2, \ldots, m.$$

Пусть $a_{ij}^{(k)}$ — направляющий элемент преобразования на $(k + 1)$-й итерации. Тогда в результате $(k + 1)$-й итерации метода полного исключения Гаусса получим матрицу $\mathrm{A}_\mathrm{p}^{(k+1)}$, элементы которой определяются следующими соотношениями:

1) для всех элементов направляющей строки

$$a_{il}^{(k+1)} = \frac{a_{il}^{(k)}}{a_{ij}^{(k)}}, \qquad l = 0,\ 1,\ \ldots,\ n; \qquad (2.16)$$

2) для элементов направляющего столбца

$$a_{rj}^{(k+1)} = 0, \quad r = 1,\ 2,\ \ldots,\ n, \text{ причем } r \neq i; \quad a_{ij}^{(k+1)} = 1; \qquad (2.17)$$

3) для всех остальных элементов матрицы $A_p^{(k+1)}$

$$a_{rl}^{(k+1)} = a_{rl}^{(k)} - \frac{a_{il}^{(k)}}{a_{ij}^{(k)}} a_{rj}^{(k)}, \qquad l \neq j, \quad r \neq i. \qquad (2.18)$$

Пример 2.3. Применим метод полного исключения Гаусса для исследования системы уравнений:

$$\begin{cases} x_1 + 2x_2 + x_3 + x_4 = 3, \\ 2x_1 + x_2 + x_3 + 3x_4 = 3, \\ 4x_1 + 5x_2 + 3x_3 + 5x_4 = 9. \end{cases}$$

Расширенная матрица имеет вид

$$\begin{array}{cc} & \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_0 \end{array} \\ A_p = & \begin{bmatrix} 1 & 2 & 1 & 1 & 3 \\ 2 & 1 & 1 & 3 & 3 \\ 4 & 5 & 3 & 5 & 9 \end{bmatrix}. \end{array}$$

Первый шаг. В качестве первого направляющего элемента возьмем $a_{11} = 1$. Умножив первую строку матрицы A на 2 и на 4, затем вычитая результаты из второй и третьей строк, получим

$$\begin{array}{cc} & \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_0 \end{array} \\ A_p^{(1)} = & \begin{bmatrix} 1 & 2 & 1 & 1 & 3 \\ 0 & -3 & -1 & 1 & -3 \\ 0 & -3 & -1 & 1 & -3 \end{bmatrix}. \end{array}$$

Второй шаг. Поскольку главная часть матрицы $A_p^{(1)}$ содержит отличные от нуля элементы, продолжим процесс исключения. Выберем элемент $a_{22}^{(1)} = -3$. После аналогичных преобразований получим

$$\begin{array}{cc} & \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_0 \end{array} \\ A_p^{(2)} = & \begin{bmatrix} 1 & 0 & ^1/_3 & 1^2/_3 & 1 \\ 0 & 1 & ^1/_3 & -^1/_3 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{array}$$

Как видим, главная часть матрицы $A_p^{(2)}$, состоящая из элементов $a_{33}^{(2)}$ и $a_{34}^{(2)}$, содержит только нули. Следовательно, процесс исключения заканчивается.

Исследуем матрицу $A^{(2)}$. Так как третья строка содержит только нулевые элементы, то она может быть отброшена. Тогда эквивалентная матрица системы уравнений имеет вид

$$\begin{array}{cc} & \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_0 \end{array} \\ A_p^{(3)} = & \begin{bmatrix} 1 & 0 & ^1/_3 & 1^2/_3 & 1 \\ 0 & 1 & ^1/_3 & -^1/_3 & 1 \end{bmatrix}. \end{array}$$

В соответствии с формулами (2.13) и (2.14) получим базисное решение: $x_1^* = 1$, $x_2^* = 1$, $x_3 = 0$, $x_4 = 0$.

Общее решение данной системы имеет вид:

$$x_1 = 1 - \frac{1}{3}\alpha_3 - 1\frac{2}{3}\alpha_4, \quad x_2 = 1 - \frac{1}{3}\alpha_3 + \frac{1}{3}\alpha_4, \quad x_3 = \alpha_3, \quad x_4 = \alpha_4,$$

где α_3, α_4 — произвольные скаляры.

Решение задач линейного программирования методом симплекс-таблиц

Основная идея симплекс-метода состоит в переходе от одного допустимого базисного решения к другому, так что значения целевой функции при этом непрерывно увеличиваются (задача максимизации).

Предположим, что ограничения задачи сведены к такому виду, что в матрице **A** имеется единичная подматрица и все свободные члены положительны. Другими словами, пусть матрица ограничений имеет, например, такой вид:

$$\mathbf{A}_1 x_1 + \cdots + \mathbf{A}_n x_n + \mathbf{e}_1 x_n + \mathbf{e}_1 x_{n+1} + \cdots + \mathbf{e}_m x_{n+m} =$$

$$= \mathbf{A}_0 = \begin{bmatrix} a_{10} \\ \vdots \\ a_{m0} \end{bmatrix},$$

где $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \mathbf{e}_m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$ — единичный базис и $a_{i0} \geqslant 0$ для всех $i = 1, 2, \ldots, m$.

Применим первый шаг метода полного исключения к расширенной матрице ограничений $\mathbf{A}_p = [\mathbf{A}_1, \ldots, \mathbf{A}_n, \mathbf{e}_1, \ldots, \mathbf{e}_m, \mathbf{A}_0]$.

Пусть a_{ij} — направляющий элемент преобразования на данной итерации. Тогда в результате преобразований в соответствии с (2.18) получим новые значения свободных членов

$$a_{l0}^{(k+1)} = a_{l0}^{(k)} - \frac{a_{i0}^{(k)} a_{lj}^{(k)}}{a_{ij}^{(k)}}, \quad \text{если } l \neq i, \quad l = 1, 2, \ldots, m,$$

$$a_{i0}^{(k+1)} = \frac{a_{i0}^{(k)}}{a_{ij}^{(k)}}. \tag{2.19}$$

Исследуем выражения (2.19) и выясним условия, при которых $a_{l0}^{(k+1)} > 0$ для всех l, т. е. новое базисное решение будет допустимым. По предположению, $a_{l0}^{(k)} \geqslant 0$, $l = 1, 2, \ldots, m$.

Допустим, что $a_{ij}^{(k)} > 0$, тогда $a_{i0}^{(k+1)} = \dfrac{a_{i0}^{(k)}}{a_{ij}^{(k)}} \geqslant 0$.

Если $a_{lj}^{(k)} < 0$, то, очевидно, $a_{l0}^{(k+1)} > 0$, так как $a_{i0}^{(k)} > 0$, $a_{ij}^{(k)} > 0$. Если $a_{lj}^{(k)} > 0$, то $a_{l0}^{(k+1)} = a_{lj}^{(k)} \left(\frac{a_{l0}^{(k)}}{a_{lj}^{(k)}} - \frac{a_{i0}^{(k)}}{a_{ij}^{(k)}} \right)$ будет больше нуля при всех значениях $l = 1, 2, \ldots, m$ тогда и только тогда, когда

$$\frac{a_{i0}^{(k)}}{a_{ij}^{(k)}} = \min_{l} \left\{ \frac{a_{l0}^{(k)}}{a_{lj}^{(k)}} \,\middle|\, a_{lj}^{(k)} > 0 \right\}. \qquad (2.20)$$

Таблица 2.2

c	—	—	c_1	c_2	c_3	...	c_j	...	c_n
—	B_x	a_{i0}	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	...	$\mathbf{A}_j$	...	$\mathbf{A}_n$
c_1	x_1	a_{10}	a_{11}	a_{12}	a_{13}	...	a_{1j}	...	a_{1n}
c_2	x_2	a_{20}	a_{21}	a_{22}	a_{23}	...	a_{2j}	...	a_{2n}
...	...	...	...	...	...	...	...	...	...
c_i	x_i	a_{i0}	a_{i1}	a_{i2}	a_{i3}	...	$\boxed{a_{ij}}$	...	a_{in}
...	...	...	...	...	...	...	...	...	...
c_m	x_m	a_{m0}	a_{m1}	a_{m2}	a_{m3}	...	a_{mj}	...	a_{mn}
—	Δ	a_{00}	Δ_1	Δ_2	Δ_3	...	Δ_j	...	Δ_n

Преобразование Гаусса называется *симплексным преобразованием*, когда направляющий элемент определяют по следующим правилам:

а) направляющий столбец j выбирают из условия, что в нем имеется хотя бы один положительный элемент;

б) направляющую строку i выбирают так, чтобы отношение $\frac{a_{i0}}{a_{ij}}$ было минимально при условии, что $a_{ij} > 0$.

Очевидно, при таком преобразовании в базис вводится вектор $\mathbf{A}_j$ и выводится вектор $\mathbf{A}_i$. Теперь надо определить, как выбрать вектор, вводимый в базис, чтобы при этом значение целевой функции увеличилось.

Для этого используют так называемые *оценки векторов* Δ_j:

$$\Delta_j = \sum_{i \in I_б} c_i x_{ij} - c_j = a_{0j}, \quad j \notin I_б, \qquad (2.21)$$

где $I_б$ — множество индексов базисных векторов; x_{ij} — определяют из условий:

$$\sum_{i \in I_б} \mathbf{A}_i x_{ij} = \mathbf{A}_j. \qquad (2.22)$$

Величины $\{\Delta_j\}$ равны симплекс-разностям для переменных $\{x_j\}$ с обратным знаком. Следовательно, для того чтобы значение целевой функции увеличилось, необходимо выбрать направляющий столбец $\mathbf{A}_j$ с наибольшей отрицательной оценкой, т. е.

$$\Delta_j = a_{0j} = \min_k \{a_{0k}/a_{0k} < 0\}.$$

Для решения задачи симплекс-методом на каждой итерации заполняют симплекс-таблицу 2.2.

Последняя строка таблицы — индексная служит для определения направляющего столбца. Ее элементы Δ_j определяют по формуле (2.21). Очевидно, для всех базисных векторов $\{\mathbf{A}_i\}$, $i = 1, 2, \ldots, m$ оценки $\Delta_i = a_{0i} = 0$.

Значение целевой функции a_{00} определяется соотношением

$$a_{00} = \sum_{i=1}^{m} c_i x_i^{(k)}.$$

В столбце B_x записывают базисные переменные $\{x_i\}$, $i = 1, \ldots, m$. Их значения определяют столбцом свободных членов a_{i0}, т. е.

$$x_i = a_{i0}, \qquad i = 1, \ldots, m.$$

Направляющие столбец $\mathbf{A}_j$ и строка $\mathbf{A}_i$ указываются стрелками.

Если выбран направляющий элемент a_{ij}, то переход от данной таблицы к следующей задается соотношениями (2.16), (2.17) и (2.18).

Итак, алгоритм решения задачи ЛП (задача максимизации) методом симплекс-таблиц состоит в следующем:

1. Рассчитывают и заполняют начальную таблицу с допустимым единичным базисом, включая индексную строку.

2. В качестве направляющего столбца выбирают столбец $\mathbf{A}_j$, для которого

$$a_{0j} = \min_{1 \leqslant l \leqslant n} \{a_{0l}/a_{0l} < 0\}.$$

3. Направляющая строка $\mathbf{A}_i$ выбирается из условия:

$$\frac{a_{i0}}{a_{ij}} = \min_{1 \leqslant r \leqslant m} \left\{ \frac{a_{r0}}{a_{rj}} \,\middle|\, a_{rj} > 0 \right\}.$$

4. Делают один шаг симплекс-преобразования с направляющим элементом a_{ij}, для чего используют соотношения (2.16), (2.17) и (2.18).

В частности, элементы индексной строки новой таблицы вычисляют по формулам:

$$a_{00}^{(k+1)} = a_{00}^{(k)} - \frac{a_{i0}^{(k)} a_{0j}^{(k)}}{a_{ij}^{(k)}}, \quad a_{0l}^{(k+1)} = a_{0l}^{(k)} - \frac{a_{il}^{(k)} a_{0j}^{(k)}}{a_{ij}^{(k)}}, \qquad l = 1, 2, \ldots, n.$$

Правильность вычислений контролируют по формулам непосредственного счета:

$$a_{00}^{(k+1)} = \sum_{i \in I_б^{(k+1)}} c_i a_{i0}^{(k+1)}; \tag{2.23}$$

$$a_{0l}^{(k+1)} = \sum_{i \in I_б^{(k+1)}} c_i a_{il}^{(k+1)} - c_l. \tag{2.24}$$

В столбце B_x заменяют x_i на x_l, а в столбце c c_i на c_l.

5. Если все $a_{0l}^{k+1} \geqslant 0$, $l = 1, 2, \ldots, n$, то новое базисное решение $x_i = a_{i0}^{k+1}$, $i \in I_б^{(k+1)}$ оптимально. В противном случае переходят к этапу 2 и выполняют очередную итерацию.

6. Второй, третий и четвертый этапы повторяют до тех пор, пока одна из итераций не закончится одним из двух исходов:

а) все $a_{0l} \geqslant 0$, $l = 1, 2, \ldots, n$. Это условие оптимальности базиса последней таблицы;

б) найдется такой $a_{0j} = \Delta_j < 0$, что все элементы этого столбца $a_{rj} \leqslant 0$ $(r = 1, 2, \ldots, m)$. Это признак неограниченности целевой функции $z = \sum_j c_j x_j$ на множестве допустимых решений.

Особенности применения табличного симплекс-метода.

1. *Если в качестве начального базиса выбирают базис из свободных переменных, для которых* $c_i = 0$, *то оценки для всех небазисных переменных равны* $\Delta_j = a_{0j} = -c_j$, *а соответствующее значение целевой функции* $a_{00} = \sum_{i \in I_б} c_i x_i = 0$.

2. *Отсутствие векторов с отрицательными оценками (при решении задачи максимизации) является признаком оптимальности соответствующего базисного решения.*

3. *Если имеется хотя бы одна отрицательная оценка для небазисного вектора, а его столбец содержит только отрицательные элементы, то в области допустимых решений целевая функция не ограничена.*

4. *При решении задачи минимизации в базис вводится вектор с наибольшей положительной оценкой.*

Пример 2.4. Найти max $4x_1 + 3x_2$ при ограничениях

$$x_1 \leqslant 4000;$$

$$x_2 \leqslant 6000;$$

$$x_1 + \frac{2}{3} x_2 \leqslant 6000;$$

$$x_1, x_2 \geqslant 0.$$

Расширенная форма задачи имеет следующий вид:
найти max $4x_1 + 3x_2$

$$\begin{cases} 1x_1 + 0x_2 + 1x_3 + 0x_4 + 0x_5 = 4000; \\ 0x_1 + 1x_2 + 0x_3 + 1x_4 + 0x_5 = 6000; \\ 1x_1 + \frac{2}{3} x_2 + 0x_3 + 0x_4 + 1x_5 = 6000. \end{cases}$$

Таблица 2.3

	c_i			↓ 4	3	0	0	0
		B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5
←	0	x_3	4000	$\boxed{1}$	0	1	0	0
	0	x_4	6000	0	1	0	1	0
	0	x_5	6000	1	$^2/_3$	0	0	1
		Δ	0	—4	—3	0	0	0

Так как $A_0 > 0$, а векторы A_3, A_4, A_5 образуют единичный базис, то задачу можно решать методом симплекс-таблиц.

Направляющие столбец и строку укажем стрелками.

Составляем исходную симплекс-таблицу 2.3.

Так как $-4 < -3 < 0$, то направляющий столбец — первый. Составив отношение вида $\left\{\frac{a_{i0}}{a_{i1}}\right\}$, определим направляющую строку. Для этого находим $\min\left\{\frac{4000}{1}; \frac{6000}{0}; \frac{6000}{1}\right\} = 4000.$

Итак, направляющая строка — первая, направляющий элемент $a_{11} = 1$.Выполнив первый шаг симплекс-преобразования, получим табл. 2.4.

Направляющий столбец — второй, направляющая строка — третья, так как

Таблица 2.4

	c_i			4	↓ 3	0	0	0
			a_{i0}	A_1	A_2	A_3	A_4	A_5
	4	x_1	4000	1	0	1	0	0
	0	x_4	6000	0	1	0	1	0
←		x_5	2000	0	$\boxed{^2/_3}$	—1	0	1
		Δ	16 000	0	—3	4	0	0

$\frac{2000}{\frac{2}{3}} < \frac{6000}{1} < \frac{4000}{0}$. Применив теперь шаг симплекс-преобразования, получаем табл. 2.5.

Таблица 2.5

c_i			4	3	0	0	0
		a_{i0}	A_1	A_2	A_3	A_4	A_5
4	x_1	4000	1	0	1	0	0
0	x_4	3000	0	0	$\boxed{3/2}$	1	$-3/2$
3	x_2	3000	0	1	$-3/2$	0	$3/2$
	Δ	25 000	0	0	$-1/2$	0	$9/2$

Так как $a_{03} = -\frac{1}{2} < 0$, то направляющий столбец $\mathbf{A}_3$, направляющая строка — вторая, направляющий элемент $a_{23} = \frac{3}{2}$. Выполнив очередной шаг, получаем табл. 2.6.

Поскольку в индексной строке все оценки положительны, то найдено оптимальное решение: $x_{1\text{опт}} = 2000$, $x_{2\text{опт}} = 6000$, $x_{3\text{опт}} = 2000$.

Искомое значение целевой функции

$$\max (4x_1 + 3x_2) = a_{00} = 26\,000 = 4x_{1\text{опт}} + 3x_{2\text{опт}}.$$

Таблица 2.6

c_i			4	3	0	0	0
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5
4	x_1	2000	1	0	0	$-2/3$	1
0	x_3	2000	0	0	1	$2/3$	-1
3	x_2	6000	0	1	0	1	0
	Δ	26 000	0	0	0	$1/3$	4

§ 3. НАХОЖДЕНИЕ ДОПУСТИМЫХ БАЗИСНЫХ РЕШЕНИЙ

Определение начального допустимого базиса в общем случае представляет значительные трудности. Поэтому для нахождения допустимых базисных решений разработаны специальные методы.

Метод искусственных переменных. Пусть ограничения имеют вид $\mathbf{A}\mathbf{x} \leqslant \mathbf{A}_0$.

Если все $b_i \geqslant 0$, $i = 1, 2, \ldots, m$, то свободные векторы, образующие единичную подматрицу, составляют начальный базис, а соответствующие им переменные — начальное базисное решение.

В более общем случае, когда ряд неравенств имеет знак больше или равно, например, $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geqslant b_i$, $i = 1, 2, \ldots, m$, то для приведения их к стандартной форме равенств свободные переменные надо вычесть. Тогда расширенная форма задачи имеет вид:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{1n}x_n - 1x_{n+1} + 0x_{n+2} + \cdots + 0x_{n+m} = b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + 0x_{n+1} - 1x_{n+2} + \cdots + 0x_{n+m} = b_2; \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + 0x_{n+1} + \cdots - 1x_{n+m} = b_m. \end{array} \quad (3.1)$$

Свободные переменные $\{x_{n+1}, x_{n+2}, \ldots, x_{n+m}\}$ уже нельзя использовать в качестве начального базиса, так как $x_{n+1} < 0$, $x_{n+2} < 0$, ..., $x_{n+m} < 0$. Поэтому в уравнения (3.1) дополнительно вводят искусственные переменные $x_{n+m+1}, x_{n+m+2}, \ldots, x_{n+m+k}$. Эти переменные не имеют ничего общего с реальной задачей и должны быть выведены из базиса как можно скорее. Чтобы гарантировать их быстрое выведение после начала итераций, искусственным переменным в целевой функции приписывают очень большие по величине отрицательные коэффициенты ($-M$) для задач максимизаций, где $M \gg c_i$ $(i = 1, 2, \ldots, n)$.

В случае решения задач минимизации искусственные переменные вводят в целевую функцию с большими по величине положительными коэффициентами ($+M$).

Знаки вводимых в ограничения искусственных переменных x_{n+m+1}, $x_{n+m+2}, \ldots, x_{n+m+k}$ должны совпадать со знаками соответствующих свободных членов.

Искусственные переменные образуют начальное базисное решение. Применив симплекс-метод, необходимо вывести из базиса все искусственные переменные. Если доказано, что от искусственных переменных избавиться нельзя, то это означает, что задача не имеет решения, т. е. ее ограничения противоречивы.

Пример 2.5. Найти $\min f(\mathbf{x}) = \min(15x_1 + 33x_2)$ при ограничениях

$$\begin{cases} 3x_1 + 2x_2 \geqslant 6; \\ 6x_1 + x_2 \geqslant 6; \\ x_2 \geqslant 1; \\ x_1,\ x_2 \geqslant 0. \end{cases}$$

Вводя свободные переменные x_3, x_4, x_5, приходим к расширенной форме задачи:

$$\begin{cases} 3x_1 + 2x_2 - x_3 = 6; \\ 6x_1 + x_2 - x_4 = 6; \\ x_2 - x_5 = 1. \end{cases}$$

Переменные x_3, x_4, x_5 образуют недопустимое базисное решение:

$$x_3 = -6 < 0, \quad x_4 = -6 < 0, \quad x_5 = -1 < 0.$$

Поэтому вводим в ограничения и в целевую функцию искусственные переменные x_6, x_7, x_8:

$$\min \{15x_1 + 33x_2 + Mx_6 + Mx_7 + Mx_8\};$$
$$3x_1 + 2x_2 - x_3 + x_6 = 6;$$
$$6x_1 + x_2 - x_4 + x_7 = 6;$$
$$x_2 - x_5 + x_8 = 1.$$

Очевидно, начальное базисное решение: $x_6^* = 6$; $x_7^* = 6$; $x_8^* = 1$. Так как A_6, A_7, A_8 образуют единичный базис, а все $a_{i0} > 0$, то для решения применим метод симплекс-таблиц (табл. 2.7).

Таблица 2.7

c_i			15	33	0	0	0	M	M	M
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
M	x_6	6	3	2	—1	0	0	1	0	0
M	x_7	6	$\boxed{6}$	1	0	—1	0	0	1	0
M	x_8	1	0	1	0	0	—1	0	0	1
		13M	9M—15	4M—33	—M	—M	—M	0	0	0

Первая итерация. Элементы индексной строки вычисляем следующим образом: $a_{06} = a_{07} = a_{08} = 0$;

$$a_{01} = \sum_{i \in J_б} c_i x_{i1} - c_1 = \sum_{i=6} c_i x_{i1} - c_1 = 3M + 6M + 0M - 15 = 9M - 15;$$

$$a_{02} = \sum_{i \in J_б} c_i x_{i2} - c_2 = 2M + 1M + 1M - 33 = 4M - 33;$$

$$a_{03} = \sum_{i \in J_б} c_i x_{i3} - c_3 = -M; \quad a_{04} = -M; \quad a_{05} = -M;$$

$$a_{00} = \sum_{i \in J_б} c_i x_i = 6M + 6M + 1M = 13M.$$

Поскольку это задача минимизации, то направляющий столбец определяют по наибольшему положительному элементу индексной строки:

$$a_{0l} = \max_{1 \leqslant k \leqslant n} \{a_{0k}/a_{0k} > 0\}.$$

Направляющий столбец — A_1; направляющая строка — вторая.

Выполним один шаг преобразования. Разделим направляющую строку на направляющий элемент $a_{71} = 6$. Умножив преобразованную направляющую строку на 3, вычтем ее из первой. Затем, умножив преобразованную направляющую строку на (9M — 15), вычтем ее из индексной. В результате получим табл. 2.8.

Таблица 2.8

↓

c_i			15	33	0	0	0	M	M	M
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
M	x_6	3	0	$1\frac{1}{2}$	-1	$\frac{1}{2}$	0	1	$-\frac{1}{2}$	0
15	x_1	1	1	$\frac{1}{6}$	0	$-\frac{1}{6}$	0	0	$\frac{1}{6}$	0
← M	x_8	1	0	$\boxed{1}$	0	0	-1	0	0	1
		$4M+15$	0	$\frac{5}{2}M-30\frac{1}{2}$	$-M$	$\frac{M}{2}-\frac{15}{6}$	$-M$	0	$\frac{5}{2}-\frac{3}{2}M$	0

Вторая итерация. Так как $a_{02} = \frac{5}{2}M - 30\frac{1}{2} > a_{0j}$ $(j \neq 2)$, то направляющий столбец второго шага — A_2. Направляющая строка — третья, направляющий элемент $a_{82} = 1$.

Выполним шаг симплекс-преобразования. Умножив направляющую строку на $1\frac{1}{2}$, вычтем ее из первой строки. Затем умножив направляющую строку на $\frac{1}{6}$, вычтем ее из второй и, наконец, умножив направляющую строку на $\left(\frac{5}{2}M - 30\frac{1}{2}\right)$, вычтем ее из последней строки. Получим табл. 2.9.

Таблица 2.9

↓

c_i			15	33	0	0	0	M	M	M
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
← M	x_6	$1\frac{1}{2}$	0	0	-1	$\frac{1}{2}$	$\boxed{1\frac{1}{2}}$	1	$-\frac{1}{2}$	$-1\frac{1}{2}$
15	x_1	$\frac{5}{6}$	1	0	0	$-\frac{1}{6}$	$\frac{1}{6}$	0	$\frac{1}{2}$	$-\frac{1}{6}$
33	x_2	1	0	1	0	0	-1	0	0	1
		$\frac{3}{2}M+\frac{91}{2}$	0	0	$-M$	$\frac{M}{2}-\frac{15}{6}$	$\frac{3}{2}M-30\frac{1}{2}$	0	$\frac{15}{2}-\frac{3M}{2}$	$-\frac{5}{2}M+30\frac{1}{2}$

Третья итерация. Так как $a_{05} = \frac{3}{2}M - 30\frac{1}{2} > a_{0j}$ $(j \neq 5)$, направляющий столбец — A_5. Направляющая строка—первая, направляющий элемент $a_{65} = 1\frac{1}{2}$.

Выполнив очередной шаг симплекс-преобразования, выведем из базиса последнюю искусственную переменную x_6 и введем x_5. Таким образом, приходим к табл. 2.10.

Таблица 2.10

c_i			15	33	0	0 ↓	0	M	M	M
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
← 0	x_5	1	0	0	$-\frac{2}{3}$	$\boxed{\frac{1}{3}}$	1	$\frac{2}{3}$	$-\frac{1}{3}$	-1
15	x_1	$\frac{4}{6}$	1	0	$\frac{1}{9}$	$-\frac{2}{9}$	0	$-\frac{1}{9}$	$\frac{2}{9}$	0
33	x_2	2	0	1	$-\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$-\frac{1}{3}$	0
		76	0	0	$-20\frac{1}{3}$	$\frac{46}{6}$	0	$\frac{61}{3}-M$	$\frac{46}{6}-M$	$-M$

Четвертая итерация. Обратим внимание, что в этой таблице все искусственные переменные выведены из базиса. Направляющий столбец — A_4, направляющая строка — первая, направляющий элемент $a_{54} = \frac{1}{3}$.

Таблица 2.11

c_i			15	33	0	0	0	M	M	M
	B_x	a_{i0}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
0	x_4	3	0	0	-2	1	3	2	-1	-3
15	x_1	$\frac{4}{3}$	1	0	$-\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$-\frac{2}{3}$
33	x_2	1	0	1	0	0	-1	0	0	1
		53	0	0	-5	0	-23	$5-M$	$-M$	$-M+23$

Выполнив еще один шаг симплекс-преобразования, получим табл. 2.11.

Поскольку в индексной строке все оценки $a_{0j} \leqslant 0$, то найдено оптимальное решение: $x_{1\,\text{опт}} = \frac{4}{3}$, $x_{2\,\text{опт}} = 1$, $x_{4\,\text{опт}} = 3$.

Искомое значение целевой функции $a_{00} = \min z = 53$.

Проверим это:

$$\min (15x_1 + 33x_2) = 15x_{1\text{опт}} + 33x_{2\text{опт}} = 53.$$

Метод Данцига для нахождения начального допустимого базиса [29]. Этот метод позволяет получить за конечное число шагов допустимый базис либо удостовериться в отсутствии допустимых базисных решений.

Рассмотрим случай ограничений в виде строгих равенств:

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_n x_n = \mathbf{A}_0.$$

Выбираем произвольный базис, образованный вектором $\mathbf{A}_0$, и, например, векторами $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_{m-1}$. Тогда все остальные векторы могут быть разложены по базису

$$\mathbf{A}_j = y_{0j}\mathbf{A}_0 + y_{1j}\mathbf{A}_1 + y_{2j}\mathbf{A}_2 + \cdots + y_{m-1,j}\mathbf{A}_{m-1} = \sum_{i=0}^{m-1} y_{ij}\mathbf{A}_i, \qquad (3.2)$$

где $j = m, m+1, ..., n$.

Условие $y_{0j} \leqslant 0$ для всех $j = m, m+1, ..., n$ является достаточным для отсутствия допустимых базисных решений.

Нахождение начального допустимого базиса начинают с выбора произвольных величин $w_1 > 0$, $w_2 > 0$, ..., $w_{m-1} > 0$, $\rho_0 > 0$.

Образуем вспомогательный вектор G:

$$\mathbf{G} = w_1\mathbf{A}_1 + w_2\mathbf{A}_2 + \cdots + w_{m-1}\mathbf{A}_{m-1} - \rho_0\mathbf{A}_0$$

или

$$\mathbf{G} + \rho_0\mathbf{A}_0 = \sum_{i=1}^{m-1} w_i\mathbf{A}_i. \qquad (3.3)$$

Для существования допустимого базиса необходимо, чтобы существовал такой вектор $\mathbf{A}_j$, для которого $y_{0j} > 0$.

Выберем такой вектор и умножим (3.2) на θ. Затем вычтем результат из уравнения (3.3):

$$\mathbf{G} + (\rho_0 + \theta y_{0j})\,\mathbf{A}_0 = \theta\mathbf{A}_j + \sum_{i=1}^{m-1} (w_i - \theta y_{ij})\,\mathbf{A}_i. \qquad (3.4)$$

По предположению $\rho_0 + \theta y_{0j} = \rho > 0$. Если в уравнении (3.2) все $y_{ij} < 0$ ($i = 1, 2, ..., m-1$), то в уравнении (3.4) все коэффициенты вида $(w_i - \theta y_{ij})$ будут больше нуля.

Решив тогда уравнение (3.2) относительно $\mathbf{A}_0$, получим

$$\mathbf{A}_0 = \frac{1}{y_{0j}}\mathbf{A}_j + \left(-\frac{y_{1j}}{y_{0j}}\right)\mathbf{A}_1 + \left(-\frac{y_{2j}}{y_{0j}}\right)\mathbf{A}_2 + \cdots + \left(-\frac{y_{m-1,j}}{y_{0j}}\right)\mathbf{A}_{m-1}. \qquad (3.5)$$

Если все коэффициенты вида $\left(-\frac{y_{kj}}{y_{0j}}\right) > 0$ ($k = 1, 2, \ldots, m-1$), то $\mathbf{A}_j, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{m-1}$ образуют допустимый базис, причем базисные решения определяются автоматически: $\left(x_j^* = \frac{1}{y_{0j}}\right)$, $\left(x_1^* = -\frac{y_{1j}}{y_{0j}}\right)$ и т. д.

Если по крайней мере один из $y_{ij} > 0$ ($i = 1, 2, \ldots, m-1$), то соответствующий вектор $\mathbf{A}_i$ должен быть выведен из базиса.

Для этого выбирают $\theta_0 = \min \left\{\frac{w_i}{y_{ij}}\right\}$, где минимум ищут по всем i, для которых $y_{ij} > 0$.

Если принять $\theta = \theta_0$, то в соотношении (3.4) один из коэффициентов перед $\mathbf{A}_i$ обратится в нуль. Новый вектор $\mathbf{G}_1 + \rho_1\mathbf{A}_0$ находят из уравнения (3.4), причем $\rho_1 = \theta_0 y_{0j} + \rho_0$.

Разложив по базису все векторы $\mathbf{A}_j$, описанный выше процесс повторяют. После конечного числа итераций приходят к одному из двух исходов:

1) если $y_{0j} \leqslant 0$ для всех $j = m+1, \ldots n$, тогда нет допустимых базисных решений;

2) если $y_{0j} > 0$, а $y_{ij} < 0$ для всех $i = 1, 2, \ldots, m$, то в этом случае, решив уравнение (3.2) относительно $\mathbf{A}_0$, получают искомое допустимое базисное решение.

Итак, алгоритм нахождения допустимого базиса состоит из следующих этапов:

1. Выбирают произвольный базис вида $\{\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{m-1}\}$. В соответствии с уравнением (3.4) определяют y_{ij} ($i = 0, 1, 2, \ldots, m-1$).

2. Выбирают произвольные положительные значения вспомогательных переменных ρ_0, w_i ($i = 1, 2, \ldots, m-1$) и записывают выражение (3.4).

3. Используя уравнение (3.2), отыскивают вектор $\mathbf{A}_j$, для которого $y_{0j} > 0$. Этот вектор должен быть введен в базис.

4. Отыскивают вектор, который должен быть выведен из базиса. Его индекс определяют из условия $\min\limits_i \left\{\frac{w_i}{y_{ij}}\right\} = \theta_0$, где минимум ищут по всем i, для которых $y_{ij} > 0$.

5. Образуют соотношение $\mathbf{G} + (\rho_0 + \theta_0 y_{0j})\,\mathbf{A}_0 = \theta_0\mathbf{A}_j + \sum\limits_{i=1}^{m-1} (w_i - \theta_0 y_{ij})\,\mathbf{A}_j$, откуда получают новые значения переменных $\rho' = \rho_0 + \theta_0 y_{0j}$; $w'_i = w_i - \theta_0 y_{ij}$.

6. Третий, четвертый, пятый шаги повторяют до тех пор, пока в равенстве $\mathbf{A}_j = y_{0j}\mathbf{A}_0 + \sum\limits_{i=1}^{m-1} y_{ij}\mathbf{A}_i$ все y_{ij} не станут отрицательными.

В этом случае допустимое решение имеет вид:

$$\frac{1}{y_{0j}}\,\mathbf{A}_j + \sum_{i=1}^{m-1} \frac{(-y_{ij})\mathbf{A}_i}{y_{0j}} = \mathbf{A}_0. \tag{3.6}$$

Таким образом, нахождение допустимого начального базиса осуществляют одной из модификаций симплекс-метода.

Пример 2.6. Найти допустимое базисное решение системы уравнений:

$$\begin{cases} 3x_1 + x_3 - x_5 = 3; \\ x_1 + x_2 - 3x_4 = -12; \\ x_2 + x_3 + x_5 = 4. \end{cases}$$

Данная система уравнений в матричном виде

$$\begin{bmatrix} 3 & 0 & 1 & 0 & -1 \\ 1 & \cancel{0}^{1} & 0 & -3 & 0 \\ 0 & 1 & \overset{1}{\cancel{0}} & 0 & 1 \\ A_1 & A_2 & A_3 & A_4 & A_5 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ \cdots \\ \cdots \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 \\ -12 \\ 4 \end{bmatrix}$$

и в векторной записи

$$A_1x_1 + A_2x_2 + A_3x_3 + A_4x_4 + A_5x_5 = A_0.$$

Первый шаг. Зададимся произвольным базисом A_0, A_1, A_3 и найдем значения y_{ij}.

Решая уравнение $A_2 = A_0y_{02} + A_1y_{12} + A_3y_{32}$, получим $y_{02} = -\frac{4}{35}$, $y_{12} = -\frac{13}{35}$, $x_{32} = \frac{51}{35}$.

Решая уравнение $A_4 = A_0y_{04} + A_1y_{14} + A_3y_{34}$, получим $y_{04} = \frac{9}{35}$, $y_{14} = \frac{3}{35}$, $y_{34} = -\frac{36}{35}$.

Наконец, решая уравнение $A_5 = A_0y_{05} + A_1y_{15} + A_3y_{35}$, получим $y_{05} = -\frac{2}{35}$, $y_{15} = -\frac{24}{35}$; $y_{35} = -\frac{43}{35}$.

Сравнивая величины $y_{02} = -\frac{4}{35}$, $\underline{y_{04}} = \frac{9}{35}$, $y_{05} = -\frac{2}{35}$, устанавливаем, что в базис должен быть введен вектор A_4, так как $y_{04} > 0$.

Запишем уравнение для A_4:

$$A_4 = y_{04}A_0 + y_{14}A_1 + y_{34}A_3 = \frac{9}{35}A_0 + \frac{3}{35}A_1 - \frac{36}{35}A_3.$$

Дадим следующие значения вспомогательным переменным: $\rho_0 = 2$, $w_1 = 4$, $w_3 = 3$.

Образуем вспомогательный вектор

$$G + \rho_0A_0 = w_1A_1 + w_3A_3, \quad G + 2A_0 = 4A_1 + 3A_3.$$

Определим, какой вектор необходимо вывести из базиса:

$$\frac{w_1}{y_{14}} = \frac{4}{\frac{3}{35}} = \frac{140}{3} > 0; \quad \frac{w_3}{y_{34}} = \frac{3}{-\frac{36}{35}} = -\frac{35}{12} < 0.$$

Следовательно, выводим A_1, при этом $\theta_0 = \frac{140}{3}$.

Отыскиваем новые значения для переменных ρ и w_1 $(i = 1, \ldots, m - 1)$:

$$G + (\rho_0 + \theta_0y_{0j})A_0 = \underline{(w_1 - \theta_0y_{1j})}A_1 + (w_3 - \theta_0y_{3j})A_3 + \theta_0A_j;$$

$$G + \left(2 + \frac{140^{4}}{\cancel{3}} \cdot \frac{\cancel{9}^{3}}{\cancel{35}}\right)A_0 \overset{0}{=} \frac{140}{3}A_4 + \left[3 - \frac{140}{3} \cdot \left(-\frac{36}{3}\right)\right]A_3;$$

$$G + 14A_0 = \frac{140}{3}A_4 + 51A_3.$$

$\theta = \frac{\omega_1}{y_{14}}$

$\omega_1 - \theta y_{14} = \frac{\omega}{y_{14}}$

Отсюда

$$\rho_0' = 14, \quad w_3' = 51, \quad w_4' = \frac{140}{3}.$$

Второй шаг. Исходный базис $\{\mathbf{A}_0, \mathbf{A}_3, \mathbf{A}_4\}$. Разложим по базису небазисные векторы.

Решив уравнение $\mathbf{A}_1 = y_{01}\mathbf{A}_0 + y_{31}\mathbf{A}_3 + y_{41}\mathbf{A}_4$, получим $y_{01} = -3$; $y_{31} = 12$, $y_{41} = \frac{35}{3}$.

Решив уравнение $\mathbf{A}_2 + y_{02}\mathbf{A}_0 + y_{32}\mathbf{A}_3 + y_{42}\mathbf{A}_4$, получим $y_{02} = 1$, $y_{32} = -3$, $y_{42} = -\frac{13}{3}$.

Наконец, решив уравнение $\mathbf{A}_5 = y_{05}\mathbf{A}_0 + y_{35}\mathbf{A}_3 + y_{45}\mathbf{A}_4$, получим $y_{05} = 2$, $y_{35} = -7$, $y_{45} = -8$.

Так как $y_{02} > 0$ и $y_{05} > 0$, то в базис можно ввести как вектор $\mathbf{A}_2$, так и вектор $\mathbf{A}_5$. Введем в базис вектор $\mathbf{A}_5$:

$$\mathbf{A}_5 = y_{05}\mathbf{A}_0 + y_{35}\mathbf{A}_3 + y_{45}\mathbf{A}_4 = 2\mathbf{A}_0 - 7\mathbf{A}_3 - 8\mathbf{A}_4.$$

Поскольку $y_{35} = -7 < 0$ и $y_{45} = -8 < 0$, то векторы $\mathbf{A}_3$, $\mathbf{A}_4$, $\mathbf{A}_5$ образуют допустимый базис. Чтобы выделить допустимое базисное решение, приведем последнее уравнение к следующему виду:

$$\mathbf{A}_0 = \frac{1}{y_{05}}\mathbf{A}_5 - \frac{y_{35}}{y_{05}}\mathbf{A}_3 - \frac{y_{45}}{y_{05}}\mathbf{A}_4 = \frac{1}{2}\mathbf{A}_5 + \frac{7}{2}\mathbf{A}_3 + 4\mathbf{A}_4.$$

Следовательно, допустимое базисное решение:

$$x_1^* = 0, \quad x_2^* = 0, \quad x_3^* = \frac{7}{2}, \quad x_4^* = 4, \quad x_5^* = \frac{1}{2}.$$

§ 4. ДВОЙСТВЕННАЯ ЗАДАЧА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Структура и свойства двойственной задачи

Любую задачу максимизации ЛП с экономической точки зрения можно рассматривать как задачу о распределении ограниченных ресурсов $b_1, b_2, \ldots, b_m$ между различными потребителями, например, между некоторыми технологическими процессами, которые представляются столбцами $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ матрицы ограничений задачи. Любое допустимое решение задачи ЛП $x_1, x_2, \ldots, x_n$ дает конкретное распределение, указывающее ту долю каждого из ресурсов, которая должна быть использована при осуществлении соответствующего технологического процесса.

Рассмотрим пример. Завод производит три вида продукции x_1, x_2 и x_3, каждый из которых требует затрат времени на обработку на токарном, фрезерном и сверлильном станках. Количество машинного времени для каждого из станков ограничено. Пусть c_1, c_2, c_3 — прибыль от реализации единицы соответствующего вида продукции. Необходимо определить, какое количество каждого вида продукции необходимо производить в течение недели, чтобы получить максимальную прибыль.

Формально эта задача записывается так:
найти

$$\max_{x_1, x_2, x_3} (c_1x_1 + c_2x_2 + c_3x_3) \tag{4.1}$$

при ограничениях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leqslant b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leqslant b_2; \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leqslant b_3, \end{cases} \qquad (4.2)$$

где a_{1j}, a_{2j}, a_{3j} — время, необходимое для обработки единицы j-го вида продукции соответственно на токарном, фрезерном и сверлильном станках ($j = 1, 2, 3$); b_1, b_2, b_3 — недельный ресурс машинного времени соответственно для токарного, фрезерного и сверлильного станков.

Обозначим y_1, y_2 и y_3 — цену единицы времени работы на токарном, фрезерном и сверлильном станках. Тогда $a_{11}y_1 + a_{21}y_2 + a_{31}y_3$ — можно трактовать как расходы на изготовление единицы продукции первого вида, $a_{12}y_1 + a_{22}y_2 + a_{23}y_3$ — расходы на изготовление единицы продукции второго вида и т. д.

Предположим, что цены ресурсов y_1, y_2, ..., y_m выбраны так, что выполняются следующие соотношения:

$$\left.\begin{aligned} a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \geqslant c_1; \\ a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \geqslant c_2; \\ a_{13}y_1 + a_{23}y_2 + a_{33}x_3 \geqslant c_3. \end{aligned}\right\} \qquad (4.3)$$

Поскольку b_1, b_2 и b_3 — использованный ресурс машинного времени для каждого из станков, то $b_1y_1 + b_2y_2 + b_3y_3$ — суммарные расходы на производство.

Требуется найти такие y_1, y_2 и y_3, удовлетворяющие условиям (4.3), при которых минимизируются суммарные расходы на производство:

$$\min g(y_1, y_2, y_3) = b_1y_1 + b_2y_2 + b_3y_3, \qquad (4.4)$$

$$y_1 \geqslant 0, \quad y_2 \geqslant 0, \quad y_3 \geqslant 0.$$

Такую задачу называют *двойственной задачей* по отношению к задаче (4.1), *называемой прямой*.

Запишем теперь прямую и двойственную задачи в общем случае.

Прямая задача

$$\max_{\{x_j\}} \sum_{j=1}^{n} c_jx_j \qquad (4.5)$$

при условиях

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \qquad i = \overline{1, m}; \qquad (4.6)$$

$$x_j \geqslant 0, \qquad j = \overline{1, n}. \qquad (4.7)$$

Двойственная задача

$$\min \sum_{i=1}^{m} b_iy_i \qquad (4.8)$$

при условиях

$$\sum_{i=1}^{m} a_{ij}y_i \geqslant c_j, \qquad j = \overline{1, n}; \qquad (4.9)$$

$$y_i \geqslant 0, \qquad i = \overline{1, m}. \qquad (4.10)$$

В матричном виде пара двойственных задач записывается следующим образом:

$$\max \mathbf{c}^T\mathbf{x}; \tag{4.11}$$

$$\mathbf{A}\mathbf{x} \leqslant \mathbf{b}; \tag{4.12}$$

$$\mathbf{x} \geqslant 0; \tag{4.13}$$

$$\min \mathbf{b}^T\mathbf{y}; \tag{4.14}$$

$$\mathbf{A}^T\mathbf{y} \geqslant \mathbf{c}; \tag{4.15}$$

$$\mathbf{y} \geqslant 0. \tag{4.16}$$

Сопоставляя формы записи прямой и двойственной задач, можно установить между ними следующие взаимосвязи:

1) если *прямая задача* является *задачей максимизации*, то *двойственная* будет *задачей минимизации*, и наоборот;

2) *коэффициенты целевой функции прямой задачи* $c_1, c_2, \ldots, c_n$ становятся *свободными членами ограничений двойственной задачи*;

3) *свободные члены ограничений прямой задачи* $b_1, b_2, \ldots, b_m$ *становятся коэффициентами целевой функции двойственной задачи*;

4) *матрицу ограничений двойственной задачи получают транспонированием матрицы ограничений прямой задачи*;

5) *знаки неравенств в ограничениях изменяются на обратные*;

6) число ограничений прямой задачи равно числу переменных двойственной задачи, а число ограничений двойственной задачи равно числу переменных прямой задачи.

Переменные $y_1, y_2, \ldots, y_m$ двойственной задачи иногда называют *«теневыми ценами»*.

Двойственную задачу выгоднее решать, чем исходную прямую, если в прямой задаче при малом количестве переменных имеется большое количество ограничений ($m > n$).

Связь между оптимальными решениями прямой и двойственной задач устанавливают посредством следующих теорем теории двойственности.

Теорема 2.4. *Если* $\mathbf{x}_0$ *и* $\mathbf{y}_0$ — *допустимые решения прямой и двойственной задач, т. е. если* $\mathbf{A}\mathbf{x}_0 \leqslant \mathbf{b}$ и $\mathbf{A}^T\mathbf{y}_0 \geqslant \mathbf{c}$, *то*

$$\mathbf{c}^T\mathbf{x}_0 \leqslant \mathbf{b}^T\mathbf{y}_0, \tag{4.17}$$

т. е. значения целевой функции прямой задачи никогда не превышают значений целевой функции двойственной задачи.

Доказательство. Умножим выражение (4.12) на $\mathbf{y}_0^T$ и получим

$$\mathbf{y}_0^T\mathbf{A}\mathbf{x}_0 \leqslant \mathbf{y}_0^T\mathbf{b}. \tag{4.18}$$

Аналогичным образом умножим (4.15) на $\mathbf{x}_0^T$

$$\mathbf{x}_0^T\mathbf{A}^T\mathbf{y}_0 \geqslant \mathbf{x}_0^T\mathbf{c}. \tag{4.19}$$

Но $\mathbf{y}_0^T\mathbf{A}\mathbf{x}_0 = (\mathbf{y}_0^T\mathbf{A}\mathbf{x}_0)^T = \mathbf{x}_0^T\mathbf{A}^T\mathbf{y}_0$, а кроме того $\mathbf{x}_0^T\mathbf{c} = \mathbf{c}^T\mathbf{x}_0$.

Поэтому, сравнив (4.18) с (4.19), получим

$$y_0^T b \geqslant y_0^T A x_0 \geqslant x_0^T c \text{ ил } c^T x_0 \leqslant b^T y_0.$$

Теорема доказана.

Теорема 2.5. (основная теорема двойственности). *Если* x_0 *и* y_0 — *допустимые решения прямой и двойственной задач и если* $c^T x_0 = b^T y_0$, *то* x_0 *и* y_0 — *оптимальные решения пары двойственных задач.*

Доказательство. Согласно теореме 2.4 для всех допустимых решений x и y справедливо неравенство (4.17). В частности, для всех допустимых решений x справедливо $c^T x \leqslant b^T y_0$. Но из условия $c^T x_0 = b^T y_0$ следует $c^T x \leqslant c^T x_0$. Следовательно, x_0 — оптимальное решение.

Вторая часть теоремы доказывается аналогично.

В силу теоремы 2.4 для всех допустимых y справедливо $c^T x_0 \leqslant b^T y$. Но из условия $c^T x_0 = b^T y_0$ следует $b^T y \geqslant b^T y_0$ для всех $y \geqslant 0$.

Таким образом y_0 — оптимальное решение.

Теорема 2.6. *Если в оптимальном решении прямой задачи (4.5) — (4.7) i-е ограничение выполняется как строгое неравенство, то оптимальное значение соответствующей двойственной переменной равно нулю, т. е. если*

$$\sum_{j=1}^{n} a_{ij} x_{j\text{опт}} = A^{(i)} x_{\text{опт}} < b_i, \text{ то } y_{i\text{опт}} = 0, \tag{4.20}$$

где A^i — i- я строка матрицы A.

Смысл теоремы 2.6 состоит в следующем. Если некоторый ресурс b_i имеется в избытке и i-е ограничение при оптимальном решении выполняется как строгое неравенство, то оно становится несущественным, и оптимальная цена соответствующего ресурса равна 0.

Теорему 2.6 дополняет теорема 2.7, устанавливающая взаимосвязь между оптимальным решением прямой задачи и ограничениями двойственной.

Теорема 2.7. *Если в оптимальном решении двойственной задачи ограничение j выполняется как строгое неравенство, то оптимальное значение соответствующей переменной прямой задачи должно быть равно нулю, т. е. если* $A_j^T y_{\text{опт}} - c_j > 0$, *то*

$$x_{j\text{опт}} = 0. \tag{4.21}$$

Дадим экономическую интерпретацию теоремы 2.7.

Поскольку величины y_i $(i = \overline{1, m})$ представляют собой цены соответствующих ресурсов, то $A_j^T y = \sum_{i=1}^{m} a_{ij} y_i$ — это затраты на j-й технологический процесс, величина c_j — прибыль от реализации на единицу изделия. Поэтому с экономической точки зрения теорема 2.7 означает следующее: если j-й технологический процесс оказывается строго невыгодным с точки зрения оптимальных цен ресурсов $y_{\text{опт}}$, то в оптимальном решении прямой задачи интенсивность использования данного технологического процесса x_j должна быть равна 0.

Таким образом, теорема 2.7 выражает принцип рентабельности оптимального организованного производства.

Из нее вытекает также, что если $x_{j\text{опт}} > 0$, то

$$\mathbf{A}_j^T \mathbf{y}_{\text{опт}} - c_j = 0. \tag{4.22}$$

Предположим, что среди переменных $x_1, x_2, \ldots, x_n$ прямой задачи есть множество из m переменных, которые в оптимальном решении имеют ненулевое значение. Пусть, например, таковыми оказались первые по порядку m переменных.

Тогда на основании уравнения (4.22) получают m условий рентабельности:

$$\begin{aligned} \mathbf{A}_1^T \mathbf{y}_{\text{опт}} - c_1 &= 0; \\ \mathbf{A}_2^T \mathbf{y}_{\text{опт}} - c_2 &= 0; \\ \mathbf{A}_m^T \mathbf{y}_{\text{опт}} - c_m &= 0, \end{aligned} \tag{4.23}$$

где

$$\mathbf{A}_1^T = (a_{11}, a_{21}, \ldots, a_{m1});$$

$$\mathbf{A}_m^T = (a_{1m}, a_{2m}, \ldots, a_{mm}).$$

Доказательства теорем 2.6 и 2.7 проведем поочередно.

Пусть $\mathbf{x}_{\text{опт}}$ и $\mathbf{y}_{\text{опт}}$ — оптимальные решения прямой и двойственной задач. В силу допустимости этих решений имеем

$$\mathbf{A}\mathbf{x}_{\text{опт}} - \mathbf{b} \leqslant \mathbf{0}, \tag{4.24}$$

$$\mathbf{A}^T \mathbf{y}_{\text{опт}} - \mathbf{c} \geqslant \mathbf{0}. \tag{4.25}$$

Умножим неравенство (4.24) на $\mathbf{y}_{\text{опт}}$, а неравенство (4.25) на $\mathbf{x}_{\text{опт}}^T$:

$$\mathbf{y}_{\text{опт}}^T \mathbf{A}\mathbf{x}_{\text{опт}} - \mathbf{y}_{\text{опт}}^T \mathbf{b} \leqslant 0; \tag{4.26}$$

$$\mathbf{x}_{\text{опт}}^T \mathbf{A}^T \mathbf{y}_{\text{опт}} - \mathbf{x}_{\text{опт}}^T \mathbf{c} \geqslant 0. \tag{4.27}$$

Так как в силу теоремы 2.5 $\mathbf{y}_{\text{опт}}^T \mathbf{b} = \mathbf{x}_{\text{опт}}^T \mathbf{c}$ и $\mathbf{y}_{\text{опт}}^T \mathbf{A}\mathbf{x}_{\text{опт}} = \mathbf{x}_{\text{опт}}^T \mathbf{A}^T \mathbf{y}_{\text{опт}}$, то выражения (4.26) и (4.27) строго равны нулю.

Расписав левую часть неравенства (4.26), получим

$$\mathbf{y}_{\text{опт}}^T (\mathbf{A}\mathbf{x}_{\text{опт}} - \mathbf{b}) = y_{1\text{опт}} (\mathbf{A}^{(1)}\mathbf{x}_{\text{опт}} - b_1) + \\ + y_{2\text{опт}} (\mathbf{A}^{(2)}\mathbf{x}_{\text{опт}} - b_2) + \cdots + y_{m\text{опт}} (\mathbf{A}^{(m)}\mathbf{x}_{\text{опт}} - b_m) = 0. \tag{4.28}$$

Поскольку все $y_{i\text{опт}} > 0$ и $\mathbf{A}^{(i)}\mathbf{x}_{\text{опт}} - b_i \leqslant 0$ для всех $i = 1, 2, \ldots$ $\ldots, m$, левая часть уравнения (4.28) может быть равна нулю только в том случае, если каждое слагаемое в отдельности равно нулю.

Таким образом, для каждого i, при котором $\mathbf{A}^{(i)}\mathbf{x}_{\text{опт}} - b_i < 0$, имеем $y_{i\text{опт}} = 0$, что и требовалось доказать в теореме 2.6.

Рассмотрим теперь левую часть неравенства 4.27, предварительно расписав ее:

$$\mathbf{x}_{\text{опт}}^T \mathbf{A}^T \mathbf{y}_{\text{опт}} - \mathbf{x}_{\text{опт}}^T \mathbf{c} = \mathbf{x}_{\text{опт}}^T (\mathbf{A}^T \mathbf{y}_{\text{опт}} - \mathbf{c}) = \\ = x_{1\text{опт}} (\mathbf{A}_1^T \mathbf{y}_{\text{опт}} - c_1) + \cdots + x_{n\text{опт}} (\mathbf{A}_n^T \mathbf{y}_{\text{опт}} - c_n) = 0, \tag{4.29}$$

где $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n]$.

Так как все $x_{j\text{опт}} \geqslant 0$ и $\mathbf{A}_j^T \mathbf{y}_{\text{опт}} - c_j \geqslant 0$ для всех $j = 1, 2, \ldots, n$, то уравнение (4.29) строго равно нулю, если для каждого j, при котором $A_j^T y_{опт} - c_j > 0$, соответствующая переменная $x_{jопт}$ равна нулю.

Приведем еще две важные теоремы теории двойственности [24, 57].

Теорема 2.8 (теорема существования). *Прямая и двойственная задачи имеют оптимальные решения тогда и только тогда, когда обе они имеют допустимые решения.*

Теорема 2.9 (теорема двойственности). *Допустимый вектор* x_0 *оптимален тогда и только тогда, когда в двойственной задаче имеется такое допустимое решение* y_0, *что*

$$c^T x_0 = A_0^T y_0. \tag{4.30}$$

Доказательство теорем 2.8 и 2.9 несложно и предлагается читателю.

Между оптимальными решениями прямой и двойственной задач и элементами индексных строк симплекс-таблиц, соответствующих этим решениям, существует следующая взаимосвязь:

$$\Delta_{n+i}^{пр} = y_{iопт}; \quad -\Delta_{m+j}^{дв} = x_{jопт}, \tag{4.31}$$
$$i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n,$$

где n — количество переменных прямой задачи; m — количество ее ограничений;

$\Delta_{n+i}^{(пр)}$, $\Delta_{m+j}^{(дв)}$ — соответствующие элементы индексной строки прямой и двойственной задач соответственно. При этом, если $n + i$ (где $1 \leqslant i \leqslant m$) больше числа векторов-столбцов матрицы ограничений расширенной формы соответствующей задачи, то элементы Δ_{n+i} (и Δ_{m+j}) находятся путем циклической перестановки элементов индексной строки, начиная с элемента Δ_1.

Пример 2.7. Рассмотрим следующую задачу ЛП: найти

$$\max f(x_1, x_2) = c_1x_1 + c_2x_2 = 4x_1 + 3x_2$$

при ограничениях

$$1x_1 + 0x_2 \leqslant 4000;$$
$$0x_1 + 1x_2 \leqslant 6000;$$
$$1x_1 + \frac{2}{3}x_2 \leqslant 6000.$$

Двойственная задача записывается так:
найти $\min g(y_1, y_2, y_3) = \min(4000y_1 + 6000y_2 + 6000y_3)$ при ограничениях

$$1y_1 + 0y_2 + 1y_3 \geqslant c_1 = 4;$$
$$0y_1 + 1y_2 + \frac{2}{3}y_3 \geqslant c_2 = 3.$$

В §2 настоящей главы было найдено решение прямой задачи: $x_{1\,опт} = 2000$, $x_{2\,опт} = 6000$, $x_{3\,опт} = 2000$, $\max f = 26\,000$.

Для решения двойственной задачи симплекс-методом введем свободные переменные u_4 и u_5 и запишем ограничения в следующем виде:

$$1y_1 + 0y_2 + 1y_3 - y_4 = 4;$$
$$0y_1 + 1y_2 + \frac{2}{3}y_3 - y_5 = 3.$$

Решаем задачу табличным симплекс-методом, исходная задача записывается в табл. 2.12.

Таблица 2.12

c_i			4000	6000	6000	0	0
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
4000	y_1	4	1	0	$\boxed{1}$	—1	0
6000	y_2	3	0	1	2/3	0	—1
	Δ	34 000	0	0	2000	—4000	—6000

Так как $\Delta_3 = 2000 > 0$, то в качестве направляющего выбираем столбец A_3. Выполнив первую итерацию симплекс-метода с направляющим элементом $a_{13} = 1$, получим табл. 2.13.

Таблица 2.13

c_i			4000	6000	6000		
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
6000	y_3	4	1	0	1	—1	0
6000	y_2	1/3	—2/3	1	0	2/3	—1
	Δ	26 000	—2000	0	0	—2000	—6000

Так как в индексной строке этой таблицы все оценки неположительны, то текущее базисное решение — оптимально.

Итак, мы получили $y_{1\text{опт}} = 0$; $y_{2\text{опт}} = \frac{1}{3}$; $y_{3\text{опт}} = 4$. Очевидно, min ($4000 y_1 +$ $+ 6000y_2 + 6000y_3) = 26\,000$. Следовательно, $\min g(\mathbf{y}) = \max f(\mathbf{x}) = 26\,000$, т. е. теорема 2.5 выполняется.

Проверим теперь, будут ли выполняться теоремы 2.6 и 2.7

$$A_1x_{\text{опт}} = |1 \ \ 0| \times \begin{bmatrix} 2000 \\ 6000 \end{bmatrix} = 2000 \cdot 1 + 6000 \cdot 0 = 2000 < b_1 = 4000.$$

Следовательно, $y_{1\text{опт}} = 0$, что совпадает с полученным результатом.

$$A_2x_{\text{опт}} = a_{21}x_{1\text{опт}} + a_{22}x_{2\text{опт}} = 0x_{1\text{опт}} + 1x_{2\text{опт}} = 0 \cdot 2000 + 1 \cdot 6000 = 6000 = b_2.$$

Поэтому

$$y_{2\text{опт}} \neq 0.$$

Наконец,

$$A_3x_{\text{опт}} = a_{31}x_{1\text{опт}} + a_{32}x_{2\text{опт}} = 1x_{1\text{опт}} + \frac{2}{3}x_{2\text{опт}} = 6000 = b_3;$$

$$y_{3\text{опт}} \neq 0.$$

Как видим, теорема 2.6 выполняется полностью.

Для проверки теоремы 2.7 выпишем ограничения двойственной задачи при оптимальных значениях переменных:

$$a_{11}y_{1\text{опт}} + a_{21}y_{2\text{опт}} + a_{31}y_{3\text{опт}} - c_1 = 1 \cdot 0 + 0 \cdot \frac{1}{3} + 1 \cdot 4 - 4 = 0.$$

Поэтому в силу теоремы 2.7 $x_{1\text{опт}}$ должно быть больше нуля.

Нами найдено решение (см. пример 2.4) $x_{1\text{опт}} = 2000 > 0$. Аналогичным образом для второго ограничения

$$a_{12}y_{1\text{опт}} + a_{22}y_{2\text{опт}} + a_{32}y_{3\text{опт}} - c_2 = 0 + 1 \cdot \frac{1}{3} + \frac{2}{3} \cdot 4 - 3 = 0.$$

В силу теоремы 2.7 $x_{2\text{опт}}$ должно быть больше нуля. Было получено $x_{2\text{опт}} = 6000$.

Следовательно, теорема 2.7 полностью подтверждается.

Проверим справедливость соотношений (4.31) для рассмотренной пары двойственных задач (см. табл. 2.6 и 2.13). Рассмотрим индексную строку табл. 2.6. Учтем, что $n = 2$; $m = 3$.

$$y_{1\text{опт}} = \Delta_{n+1}^{(\text{пр})} = \Delta_{2+1}^{(\text{пр})} = \Delta_3 = 0;$$

$$y_{3\text{опт}} = \Delta_{n+3}^{(\text{пр})} = \Delta_5^{(\text{пр})} = 4; \qquad y_{2\text{опт}} = \Delta_{n+2}^{(\text{пр})} = \Delta_4 = \frac{1}{3}.$$

Эти значения совпадают с найденным решением двойственной задачи в табл. 2.13.

Рассмотрим теперь индексную строку табл. 2, 13.

Согласно соотношений (4.31) имеем

$$x_{1\text{опт}} = -\Delta_{m+1}^{(\text{дв})} = -\Delta_4^{(\text{дв})} = 2000, \quad x_{2\text{опт}} = -\Delta_{m+2}^{(\text{дв})} = -\Delta_5 = 6000.$$

$x_{3\text{опт}} = -\Delta_6^{(\text{дв})}$, и так как матрица ограничений табл. 2.13 состоит из 6 столбцов, то по правилу циклической перестановки

$$x_{3\text{опт}} = -\Delta_6^{(\text{дв})} = -\Delta_1^{(\text{дв})} = 2000.$$

Это решение полностью совпадает с решением прямой задачи.

Таким образом, пользуясь соотношениями (4.31), мы, решив прямую задачу симплекс-методом, одновременно находим и решение двойственной задачи, и наоборот.

Этот вывод дает обоснование возможности перехода от прямой задачи к двойственной при $m > n$, и наоборот.

Общий случай двойственности

В предыдущем разделе были установлены основные соотношения для пары двойственных ЛП-задач при ограничениях в форме неравенств. Обобщим теперь эти результаты на случай произвольных ограничений.

Пусть прямая ЛП-задача задана в виде:
найти

$$\max \sum_{j=1}^{n} c_j x_j = \max L(\mathbf{x}), \tag{4.32}$$

при условии

$$\left.\begin{aligned} &\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i; \quad i = 1, 2, \ldots, m_1 \leqslant m, \\ &\sum_{j=1}^{n} a_{ij}x_j = b_i; \quad i = m_1 + 1, m_1 + 2, \ldots, m, \end{aligned}\right\} \tag{4.33}$$

$$x_j \geqslant 0; \quad j = 1; 2, \ldots, n_1 \leqslant n. \tag{4.34}$$

Тогда двойственная задача по отношению к (4.32) — (4.34) (или сопряженная с ней) состоит в минимизации линейной формы

$$\sum_{i=1}^{m} b_i y_i \tag{4.35}$$

при условии

$$\left.\begin{aligned} &\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \quad j = 1, 2, \ldots, n_1 \leqslant n, \\ &\sum_{i=1}^{m} a_{ij} y_i = c_j, \quad j = n_1 + 1, n_1 + 2, \ldots, n, \end{aligned}\right\} \tag{4.36}$$

$$y_i \geqslant 0; \quad i = 1, 2, \ldots, m_1 \leqslant m. \tag{4.37}$$

Таким образом, задача, сопряженная с задачей со смешанными условиями, составляется согласно следующим правилам:

1. Если переменная x_j прямой задачи предполагается неотрицательной, то j-е условие системы ограничений (4.36) является неравенством.

Если на x_j не накладывается такое ограничение, то j-е ограничение двойственной задачи будет равенством.

Аналогичным образом связаны знаки переменных двойственной задачи y_i и соответствующие им ограничения прямой задачи. Заметим, что если положить $m_1 = m$ и $n_1 = n$, то получим частный случай пары двойственных задач с ограничениями в форме неравенств.

Докажем справедливость соотношений (4.32) — (4.34) и (4.35) — (4.37), связывающих прямую и двойственную задачи.

Свяжем с каждой ЛП-задачей вида (4.32) — (4.34) следующую задачу с ограничениями в форме неравенств.

Требуется найти

$$\max \sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^{n} c_j (x_j' - x_{j+n_2}'), \tag{4.38}$$

при условии

$$\sum_{j=1}^{n_1} a_{ij} x_j' + \sum_{j=n_1+1}^{n} a_{ij} (x_j' - x_{j+n_2}') \leqslant b_i; \qquad i = \overline{1, m}, \tag{4.39}$$

$$-\sum_{j=1}^{n_1} a_{ij} x_j' - \sum_{j=n_1+1}^{n} a_{ij} (x_j' - x_{j+n_2}') \leqslant -b_i, \tag{4.40}$$

$$i = m_1 + 1, \quad m_1 + 2, \ldots, m,$$

$$x_j' \geqslant 0; \qquad j = 1, 2, \ldots, n + n_2, \tag{4.41}$$

где $n_2 = n - n_1$ — *число переменных задачи (4.32) — (4.34)*, на которые не наложено условие неотрицательности.

Установим соответствие между переменными задач (4.32) — (4.34) и (4.38) — (4.41). Непосредственным сопоставлением форм записи убеждаемся, что n-мерный вектор $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ и $(n + n_2)$-

мерный вектор $\mathbf{x}' = \{x'_1, x'_2, \dots, x'_{n+2}\}$ связаны соотношением:

$$x_j = \begin{cases} x'_j, & j = 1, 2, \dots, n_1; \\ x'_j - x'_{j+n_2}, & j = n_1 + 1, n_1 + 2, \dots, n. \end{cases} \tag{4.42}$$

Очевидно, каждому $(n + n_2)$-мерному вектору $\mathbf{x}'$ соответствует единственный n-мерный вектор $\mathbf{x}$, и вместе с тем любому n-мерному $\mathbf{x}$ соответствует целое семейство $(n + n_2)$-мерных векторов $\mathbf{x}'$.

Таким образом, соответствие, устанавливаемое формулой (4.42), является однозначным только в одну сторону. Вместе с тем, из семейства векторов $\mathbf{x}'$, соответствующих $\mathbf{x}$, всегда существуют векторы с неотрицательными компонентами.

Пусть $\mathbf{x}' = \{x'_1, x'_2, \dots, x'_{n+n_2}\}$ план задачи (4.38)—(4.41). Используя соотношение (4.42), можно легко получить, что соответствующий вектор $\mathbf{x}$ является планом задачи (4.32) — (4.34). И наоборот, если $\mathbf{x}$ — план задачи, то существует целое семейство планов $\mathbf{x}'$ задачи (4.38) — (4.41), среди которых имеются заведомо неотрицательные.

Одним из них является вектор $\tilde{\mathbf{x}}'$, где

$$\tilde{x}'_j = \begin{cases} x_j, & j = \overline{1, n_1}, \\ \max(0, x_j), & j = n_1 + 1, n_1 + 2, \dots, n, \\ \max(0, -x_{j-n_2}), & j = n + 1, n + 2, \dots, n + n_2. \end{cases} \tag{4.43}$$

Неотрицательность всех компонентов $\tilde{\mathbf{x}}'$ очевидна, а соответствие векторов $\mathbf{x}$ и $\tilde{\mathbf{x}}'$ следует из равенства

$$\tilde{x}'_j - \tilde{x}'_{j+n_2} =$$

$$= \max(0, x_j) - \max(0, -x_j) = \begin{cases} x_j - 0 = x_j, & \text{если } x_j \geqslant 0, \\ 0 - (-x_j) = x_j, & \text{если } x_j < 0, \end{cases}$$

где $j = n_1 + 1, n_1 + 2, \dots, n$.

Рассмотрим теперь задачу (4.35) — (4.37), двойственную к задаче (4.32) — (4.34). Нетрудно показать, что она легко приводится к виду (4.32) — (4.34).

Для этого достаточно положить $\bar{c}_j = -c_j$; $\bar{a}_{ij} = -a_{ij}$; $\bar{b}_i = -b_i$.

При этом задача (4.35) — (4.37) переходит в задачу максимизации линейной формы

$$\sum_{i=1}^{m} \bar{b}_i y_i \tag{4.44}$$

при условии

$$\sum_{i=1}^{m} \bar{a}_{ij} y_i \leqslant \bar{c}_j; \qquad j = 1, 2, \dots, n_1; \tag{4.45}$$

$$\sum_{i=1}^{m} \bar{a}_{ij} y_i = \bar{c}_j; \qquad j = n_1 + 1, \dots, n; \tag{4.46}$$

$$y_i \geqslant 0; \qquad i = \overline{1, m_1}. \tag{4.47}$$

Поэтому задаче (4.35) — (4.37) соответствует следующая задача с ограничениями в форме неравенств:

требуется обратить в минимум линейную форму

$$\sum_{i=1}^{m_1} b_i y_i' + \sum_{i=m_1+1}^{m} b_i (y_i' - y_{i+m_2}') \tag{4.48}$$

при условии

$$\sum_{i=1}^{m_1} a_{ij} y_i' + \sum_{i=m_1+1}^{m} a_{ij} (y_i' - y_{i+m_2}') \geqslant c_j; \qquad j = 1, 2, \ldots, n, \tag{4.49}$$

$$-\sum_{i=1}^{m_1} a_{ij} y_i' - \sum_{i=m_1+1}^{m} a_{ij} (y_i' - y_{i+m_2}') \geqslant -c_j; \tag{4.50}$$

$$j = n_1 + 1, \ldots, n,$$

$$y_i' \geqslant 0; \qquad i = 1, 2, \ldots, m + m_2, \tag{4.51}$$

где $m_2 = m - m_1$ — число переменных y_i, на которые не наложено условие неотрицательности.

Вектор $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ и соответствующий ему $(m+m_2)$-мерный вектор $\mathbf{y}' = \{y_1', y_2', \ldots, y_{m+m_2}'\}$ связаны соотношением

$$y_i = \begin{cases} y_i', & i = 1, 2, \ldots, m_1; \\ y_i' - y_{i+m_2}', & i = m_1 + 1, \ m_1 + 2, \ \ldots, m. \end{cases} \tag{4.52}$$

Следовательно, каждому плану $\mathbf{y}'$ задачи (4.48) — (4.51) соответствует план $\mathbf{y}$ задачи (4.35) — (4.37), и наоборот, любой неотрицательный вектор $\mathbf{y}'$, соответствующий плану $\mathbf{y}$ задачи (4.35) — (4.37), является планом задачи (4.48) — (4.51). При этом, если $\mathbf{y}$ и $\mathbf{y}'$ — два соответствующих друг другу плана, то из оптимальности одного из планов следует оптимальность другого плана. Запишем теперь задачу, двойственную (4.38) — (4.41). Непосредственной проверкой можно убедиться, что получим задачу в форме (4.48) — (4.51). Таким образом, задачи (4.38) — (4.41) и (4.48) — (4.51) составляют пару двойственных задач с ограничениями-неравенствами. Отсюда следует, что соответствующие им задачи (4.32) — (4.34) и (4.35) — (4.37) с произвольными ограничениями также представляют собой двойственную пару.

Заметим, что все теоремы двойственности, доказанные для задач с ограничениями в форме неравенств, легко распространяются на общий случай задач с произвольными ограничениями. Рассмотрим для примера теорему 2.5.

Теорема 2.5 (основная теорема двойственности).

Если $\mathbf{x}$ *и* $\mathbf{y}$ *— допустимые решения прямой* (4.32) *и двойственной задачи* (4.35) *и если при этом* $\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i$, *то* $\mathbf{x}$ и $\mathbf{y}$ *оптимальные решения этих задач.*

Доказательство. Допустим, что задача (4.32) — (4.34) разрешима и $\mathbf{x}$ — ее допустимый план, а $\mathbf{y}$ — допустимый план задачи (4.35) — (4.37). Введем в рассмотрение вектор $\mathbf{x}' = \{x_1', \ldots, x_{n+n_2}'\}$, связанный с $\mathbf{x}$ соотношениями (4.42) с неотрицательными компонентами. По доказанному выше $\mathbf{x}'$ является решением задачи (4.38).

Воспользуемся тогда теоремой 2.5 двойственности для задач с ограничениями-неравенствами.

Согласно этой теореме, если $\mathbf{x}'$ и $\mathbf{y}'$ — допустимые решения пары двойственных и имеет место равенство

$$\sum_{j=1}^{n_1} c_j x'_j + \sum_{j=n_1+1}^{n} c_j (x'_j - x'_{j+n_2}) = \sum_{i=1}^{m_1} b_i y'_i + \sum_{i=m_1+1}^{m} b_i (y'_i - y'_{i+m_2}), \qquad (4.53)$$

то $\mathbf{x}'$ и $\mathbf{y}'$ — оптимальные решения этой пары.

Используя соотношения (4.42), (4.52), связывающие соответствующие планы $\mathbf{x}$ и $\mathbf{x}'$, $\mathbf{y}$ и $\mathbf{y}'$, получим

$$\sum_{j=1}^{n_1} c_j x'_j + \sum_{j=n_1+1}^{n} c_j (x'_j - x'_{j+n_2}) = \sum_{j=1}^{n} c_j x_j; \qquad (4.54)$$

$$\sum_{i=1}^{m_1} b_i y'_i + \sum_{i=m_1+1}^{m} b_i (y'_i - y'_{i+m_2}) = \sum_{i=1}^{m} b_i y_i. \qquad (4.55)$$

И, таким образом, соотношения (4.53) и (4.54—55) эквивалентны, и поэтому планы $\mathbf{x}'$ и $\mathbf{y}'$ — оптимальны. Но по доказанному выше каждому оптимальному $\mathbf{x}'$ соответствует единственный оптимальный план $\mathbf{x}$, а каждому оптимальному плану $\mathbf{y}'$ соответствует оптимальный план $\mathbf{y}$. Теорема доказана.

Аналогичным образом могут быть доказаны остальные теоремы двойственности для произвольных ограничений.

Исследование моделей задач линейного программирования на чувствительность

Теория двойственности позволяет проводить анализ моделей задач линейного программирования на чувствительность.

Рассматривается обычная задача ЛП в виде:
найти

$$\max_{x_j} \sum_{j=1}^{n} c_j x_j = \max L(\mathbf{x}), \qquad (4.56)$$

при условии

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i; \qquad i = \overline{1, m}, \qquad (4.57)$$

$$x_j \geqslant 0. \qquad (4.58)$$

Напомним ее экономическую интерпретацию.

Целевая функция $L(\mathbf{x})$ — доход от реализации плана производства $\mathbf{x}$; a_{ij} — интенсивность расходования i-го ресурса при j-м способе производства; b_i — имеющийся уровень i-го ресурса.

Предположим, что величины ресурсов $\mathbf{b} = \| b_i \|$ варьируются. Тогда возникают следующие вопросы. При каких вариациях правых частей ограничений найденный оптимальный план x_0 не изменяется? Как эти вариации сказываются на величине максимального дохода

$L_{\max}$? Ответ на эти вопросы дает *анализ* соответствующей ЛП-задачи *на чувствительность.*

Пусть ограничения b_i получают некоторые вариации Δb_i, что приводит соответственно к вариациям плана $\mathbf{x}_0$, $\mathbf{x}_0 = \mathbf{x}_0 (\mathbf{b}+\Delta\mathbf{b})$, и функции $L_{\max} (\mathbf{x}_0 (\mathbf{b} + \Delta\mathbf{b}))$.

Пусть эти вариации $\Delta\mathbf{b}$ таковы, что план $\mathbf{x}_0 (\mathbf{b} + \Delta\mathbf{b})$ остается допустимым (т. е. удовлетворяет условию неотрицательности). Найдем отношение приращения $L_{\max} (\mathbf{b}) = L_{\max} (\mathbf{x}_0 (\mathbf{b} + \Delta\mathbf{b}))—L_{\max} (\mathbf{x}_0 (\mathbf{b}))$ к $\Delta\mathbf{b}$. Устремляя $\Delta b_i \to 0$, получим

$$\lim_{\Delta b_i \to 0} \frac{\Delta L_{\max}(\mathbf{b})}{\Delta b_i} = \frac{\partial L_{\max}(\mathbf{b})}{\partial b_i}, \qquad (4.59)$$

где $\mathbf{b}$ — рассматривается как варьируемый параметр.

Вспомнив, что в соответствии с основной теоремой двойственности

$$L_{\max} (\mathbf{x}_0) = \sum_{j} c_j x_j^0 = \sum_{i} b_i y_i^0 \qquad (4.60)$$

и подставляя (4.60) в (4.59), получим

$$\frac{\partial L_{\max}(\mathbf{b})}{\partial b_i} = y_i^0 \qquad (i = 1,\ 2,\ \ldots,\ m). \qquad (4.61)$$

Таким образом, *оптимальные значения двойственных переменных* y_i^0 *определяют вклад каждого ресурса в доход* $\mathrm{L}_{\max}$ *при оптимальном решении* $\mathbf{x}_0$. Эта величина численно равна дополнительному доходу при увеличении i-го ресурса b_i на единицу и условии, что ресурсы используются оптимальным образом.

Итак, величины y_i^0 служат показателями важности соответствующих ресурсов для системы. Чем больше значение y_i^0 при некотором i, тем существеннее вклад i-го ресурса в функцию максимального дохода $L_{\max}$ и тем выгоднее его увеличение. Если для некоторого i $y_i^0 = 0$, то i-й ресурс не является существенным ограничением для системы (поскольку он имеется в избытке, то $b_i - \sum_{j=1}^{n} a_{ij}x_j^0 > 0$).

Заметим, что этот вывод хорошо согласуется с результатами теоремы (2.6) двойственности, которая связывает величины y_i^0 с ограничениями прямой задачи.

Величины y_i^0 двойственной задачи часто называют «скрытыми» доходами [24] или маргинальными оценками ресурсов системы. Как мы выяснили, они могут рассматриваться как потенциальная возможность получения дополнительной прибыли за счет увеличения соответствующего ресурса при условии, что система функционирует оптимально.

Варьирование целевой функции. Теперь рассмотрим случай, когда варьируются коэффициенты $\{c_j\}$, $j = \overline{1,\ n}$. Попытаемся выяснить условия, при которых найденный ранее оптимальный план останется оптимальным при таких вариациях.

Пусть вариациям δ_{c_r} подвергся коэффициент c_r: $c_r^{\text{н}} = c_r + \delta_{c_r}$. Обозначим через $I_{\text{б}}$, $J_{\text{неб}}$ множество индексов базисных и небазисных векторов в оптимальном решении $\mathbf{x}_0$.

Найдем величины оценок $\Delta_j^{\text{н}}$ после вариации c_r для двух случаев:

1) $r \in J_{\text{неб}}$, тогда $\Delta_j^{(\text{н})} = \Delta_j$ для всех $j \neq r$;

и
$$\Delta_r^{\text{н}} = \sum_{i\in I_{\text{б}}} c_i a_{ir} - (c_r + \delta_{c_r}) \quad \text{для } j = r; \tag{4.62}$$

2) $r \in I_{\text{б}}$
$$\Delta_j^{(\text{н})} = \sum_{i\in I_{\text{б}}} c_i^{(\text{н})} a_{ij} - c_j = \sum_{i\in I_{\text{б}}} c_i a_{ij} + \delta_{c_r} a_{rj} - c_j; \quad j \in J_{\text{неб}}. \tag{4.63}$$

Очевидно, что для сохранения оптимальности прежнего плана при вариациях коэффициента c_r необходимо и достаточно сохранения знаков оценок $\Delta_j^{(\text{н})}$ для всех небазисных переменных. Поэтому из условий $\Delta_j^{(\text{н})} \geqslant 0$, определяемых через (4.62) или (4.63), можно определить допустимые вариации коэффициента δ_{c_r}, при которых сохраняется прежнее оптимальное решение.

До сих пор мы рассматривали вариации лишь одного коэффициента целевой функции. Этот же подход можно применить, когда варьируются одновременно несколько коэффициентов c_i. В таком случае получим соотношения, аналогичные (4.63), в которых оценки Δ_j будут функциями уже нескольких варьируемых параметров $(\delta_{c_1}, \delta_{c_2}, \ldots, \delta_{c_r})$. Решая совместно систему неравенств вида $\Delta_j\,(\delta_{c_1}, \delta_{c_2}, \ldots, \delta_{c_r}) \geqslant 0$, $j \in J_{\text{неб}}$, получим искомые условия для вариаций δc_r, при которых прежний оптимальный базис сохраняется при указанных вариациях.

Заметим, что данная задача относится к классу задач параметрического программирования и для ее решения разработаны специальные стандартные программы на ЭВМ [18, 24].

Пример 2.8. Рассмотрим задачу ЛП при произвольных ограничениях
$$\max\,(5x_1 + 3x_2 + 6x_3) \tag{1}$$
при условии
$$x_1 + 2x_2 + x_3 \leqslant 18;$$
$$2x_1 + 1x_2 + 3x_3 \leqslant 16; \tag{2}$$
$$x_1 + x_2 + x_3 = 10;$$
$$x_1 \geqslant 0, \quad x_2 \geqslant 0. \tag{3}$$

Составляем задачу, двойственную к (1) — (3), согласно общим правилам
$$\min\,(18y_1 + 16y_2 + 10y_3) \tag{4}$$
при условиях
$$y_1 + 2y_2 + y_3 \geqslant 5;$$
$$2y_1 + y_2 + y_3 \geqslant 3; \tag{5}$$
$$y_1 + 3y_2 + y_3 = 6;$$
$$y_1 \geqslant 0; \quad y_2 \geqslant 0. \tag{6}$$

Найдем решение задачи (1) — (3). Так как на x_3 не наложено условие неотрицательности, то заменим x_3 на свободные переменные
$$x_3 = x_4 - x_5. \tag{7}$$

Таблица 2.14

c_i			5	3	6	—6			—M
	B_x	A_0	A_1	A_2	A_4	A_5	A_6	A_7	A_8
0	x_6	18	1	2	1	—1	1		
0	x_7	16	$\boxed{2}$	1	3	—3		1	
—M	x_8	10	1	1	1	—1			1
	Δ		—M—5	—M—3	—M—6	6+M	0	0	0

Таблица 2.15

c_i			5	3	6	—6			—M
	B_x	A_0	A_1	A_2	A_4	A_5	A_6	A_7	A_8
	x_6	10	0	3/2	—1/2	1/2	1	—1/2	
5	x_1	8	1	1/2	3/2	—3/2		1/2	
—M	x_8	2	0	$\boxed{1/2}$	—1/2	1/2		—1/2	1
	Δ		0	$-\frac{M}{2}-\frac{1}{2}$	$\frac{M}{2}+\frac{3}{2}$	$-\frac{M}{2}-\frac{3}{2}$	0	$M/2$	0

Таблица 2.16

c_i			5	3	6	—6			—M
	B_x	A_0	A_1	A_2	A_4	A_5	A_6	A_7	A_8
	x_6	4	0	0	1	—1	1	1	—3
5	x_1	6	1	0	2	—2		1	—1
3	x_2	4	0	1	—1	$\boxed{1}$		—1	2
	Δ	42	0	0	1	—1	0	2	$M+1$

Таблица 2.17

c_i			5	3	6	—6			—M
	B_x	A_0	A_1	A_2	A_4	A_5	A_6	A_7	A_8
	x_6	8	0	1	0	0	1	0	—1
5	x_1	14	1	2	0	0	0	—1	3
—6	x_5	4	0	1	—1	1		—1	2
	Δ	46	0	1	0	0	0	1	$M+3$

Подставляя (7) в (1) — (3), приходим к эквивалентной задаче

$$\max (5x_1 + 3x_2 + 6x_4 - 6x_5) \tag{8}$$

при условиях

$$\begin{aligned} x_1 + 2x_2 + x_4 - x_5 &\leq 18; \\ 2x_1 - x_2 + 3x_4 - 3x_5 &\leq 16; \\ x_1 + x_2 + x_4 - x_5 &= 10; \end{aligned} \tag{9}$$

$$x_1,\ x_2,\ x_4,\ x_5 \geqslant 0. \tag{10}$$

Будем решать задачу (8) — (10) симплекс-методом, при этом вводим свободные переменные x_6 и x_7 в первое и второе ограничения и искусственную переменную x_8 в третье ограничение (10) и целевую функцию.

Составим начальную симплекс-таблицу 2.14.

Результаты последовательных итераций приводятся соответственно в табл. 2.15, 2.16 и 2.17.

Поскольку в индексной строке табл. 2.17 нет отрицательных членов, то найдено оптимальное решение $x_1^* = 14,\ x_5^* = 4,\ x_6^* = 8,\ x_2^* = 0 = x_4^* = x_7^* = 0$.

С учетом, что $x_3 = x_4 - x_5$, получим окончательно $x_1^* = 14,\ x_2^* = 0,\ x_3^* = -4$. $L_{\max} = 46$.

Используя соотношения (4.31), найдем по индексной строке табл. 2.17 оптимальное решение двойственной задачи. При этом учтем, что $m = n = 3$, а столбцы A_4 и A_5 нужно рассматривать как один столбец.

Тогда $y_{1\text{опт}} = \Delta_{1+3} = \Delta_6^{\text{пр}} = 0;\ y_{2\text{опт}} = \Delta_7 = 1;\ y_{3\text{opt}} = 3$.

Заметим, что при этом все ограничения (5) и (6) выполняются, а $\min (18y_1 + 16y_2 + 10y_3) = 46 = L_{\max}$.

§ 5. МЕТОД ОБРАТНОЙ МАТРИЦЫ

Критерии оптимальности и разрешающие множители

1. Использование идей теории двойственности позволяет по-новому сформулировать критерий оптимальности плана ЛП-задачи. Этот признак использует так называемые разрешающие множители Л. В. Канторовича [57].

Пусть задана произвольная ЛП-задача со смешанными условиями вида:

найти максимум

$$L(\mathrm{x}) = \sum_{j=1}^{n} c_j x_j \tag{5.1}$$

при условии

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i; \qquad i = 1, 2, \ldots, m_1 \leqslant m; \tag{5.2}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i; \qquad i = m_1 + 1, m_1 + 2, \ldots, m; \tag{5.3}$$

$$x_j \geqslant 0; \qquad j = 1, 2, \ldots, n_1 \leqslant n. \tag{5.4}$$

Величины $\lambda_1, \lambda_2, .., \lambda_m$ называют *разрешающими множителями*, если

$$\text{а) } \sum_{i=1}^{m} a_{ij} \lambda_i \geqslant c_j; \qquad j = 1, 2, \ldots, n_1; \tag{5.5}$$

б) $\sum_{i=1}^{m} a_{ij}\lambda_i = c_j; \qquad j = n_1+1,\ n_1+2,\ \ldots,\ n;$ **(5.6)**

в) $\lambda_i \geqslant 0; \qquad i = 1,\ 2,\ \ldots,\ m_1;$ **(5.7)**

г) для некоторого плана $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ задачи выполняются условия:

$$\sum_{i=1}^{n} a_{ij}\lambda_i = c_j \text{ при } x_j > 0 \qquad (1 \leqslant j \leqslant n_1), \tag{5.8}$$

$$\lambda_i = 0 \text{ при } \sum_j a_{ij}x_j < b_i \qquad (1 \leqslant i \leqslant m_1). \tag{5.9}$$

Вектор $\Lambda = (\lambda_1, \ldots, \lambda_m)$ — называют разрешающим вектором задачи.

Покажем, что отыскание разрешающего вектора эквивалентно решению задачи, двойственной к задаче (5.1) — (5.3), для чего рассмотрим следующую теорему.

Теорема 2.10. *Совокупность разрешающих векторов задачи* (5.1) — (5.4) *совпадает с множеством оптимальных планов задачи* (4.35) — (4.37), двойственной к ней.

Доказательство. Пусть $\Lambda = (\lambda_1, \ldots, \lambda_m)$ — произвольный разрешающий вектор задачи (5.1) — (5.3), связанный с планом $\mathbf{x}$ условиями (5.8), (5.9).

В силу условий (5.5) — (5.7) вектор Λ — план двойственной задачи. Остается доказать, что он является оптимальным.

Обозначим через E совокупность индексов j ($j = 1, 2, \ldots, n_1$), для которых $x_j > 0$. Тогда

$$\sum_{j=1}^{n} c_jx_j = \sum_{j\in E} c_jx_j + \sum_{j=n_1+1}^{n} c_jx_j. \tag{5.10}$$

Учитывая равенство (5.8), а также условия (5.6) при $j = \overline{n_1+1, n}$, получим

$$\sum_{j=1}^{n} c_jx_j = \sum_{j\in E}\left(\sum_{i=1}^{m} a_{ij}\lambda_i\right)x_j + \sum_{j=n_1+1}^{n}\left(\sum_i a_{ij}\lambda_i\right)x_j.$$

Но, по предположению, $x_j = 0$, если $j \notin E$ и $1 \leqslant j \leqslant n$, и поэтому

$$\sum_{j=1}^{n} c_jx_j = \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}\lambda_i\right)x_j = \sum_{i=1}^{m}\lambda_i\sum_{j=1}^{n} a_{ij}x_j. \tag{5.11}$$

Принимая во внимание равенство (5.3) при $i = \overline{m_1+1, m}$ и условия (5.9) при $i = \overline{1, m_1}$, получим

$$\sum_{i=1}^{m}\lambda_i\sum_{j=1}^{n} a_{ij}x_j = \sum_{i=1}^{m}\lambda_ib_i. \tag{5.12}$$

Сравнив (5.11) с (5.12), приходим окончательно к соотношению

$$\sum_{j=1}^{n} c_jx_j = \sum_{i=1}^{m}\lambda_ib_i. \tag{5.13}$$

Но согласно теореме 2.5 двойственности соотношение (5.13) указывает на оптимальность планов x и Λ. Теорема доказана.

2. Используя понятие разрешающих множителей, можно легко сформулировать признак оптимальности плана задачи линейного программирования.

Теорема 2.11. *Для оптимальности плана* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *задачи* (5.1) — (5.3) *необходимо и достаточно существование разрешающего вектора* $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, *связанного с этим планом условиями* (5.8), (5.9).

Доказательство. *Необходимость.* Предположим, что $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ — решение задачи (5.1) — (5.4). Тогда двойственная задача (4.35) — (4.37) разрешима. Пусть $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ — один из ее оптимальных планов. В таком случае согласно теореме (2.10.) вектор Λ является разрешающим вектором задачи (5.1) — (5.4), причем, как показано при доказательстве второй части теоремы, он связан соотношениями (5.8) и (5.9) с любым решением задачи (5.1) — (5.4), а следовательно, и с рассматриваемым.

Достаточность. Допустим, что существует разрешающий вектор $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, связанный с данным планом x задачи (5.1) — (5.4) условиями (5.8), (5.9).

В процессе доказательства теоремы 2.10 было получено равенство (5.13). Из этого равенства (если учесть, что x и Λ являются планами задач (5.1) и (4.35) — (4.37) соответственно) и вытекает оптимальность плана x (см. теорему 2.5).

Установленный критерий позволяет легко проверить, является ли данный план оптимальным решением ЛП-задачи или нет. С этой целью составляют систему из уравнений (5.6, 5.8, 5.9), из которой определяют вектор $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$. Затем непосредственной подстановкой проверяют, удовлетворяет ли этот вектор условиям (5.5) и (5.7) или нет. Если да, то вектор x — оптимальный план, в противном случае — неоптимальный план.

Следует заметить, что определение разрешающих множителей эквивалентно решению двойственной задачи, что в общем случае оно отнюдь не проще отыскания решения прямой задачи. Основная роль разрешающих множителей состоит в том, что их использование позволяет *формулировать новый признак оптимальности и построить иной вычислительный метод линейного программирования.*

Рассмотрим теперь задачу ЛП в канонической форме (5.14) — (5.16), что является, очевидно, частным случаем задачи (5.1) — (5.4) при $m_1 = 0, n_1 = n$.

Найти

$$\max \sum_{j=1}^{n} c_j x_j; \tag{5.14}$$

при условии

$$\sum_{j=1}^{n} a_{ij} x_j = b_i; \qquad i = \overline{1, m}, \tag{5.15}$$

$$x_j \geqslant 0; \qquad j = \overline{1, n}. \tag{5.16}$$

Для задачи (5.14) — (5.16) вектор $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ называется разрешающим вектором, если

$$\text{а) } \sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j; \qquad j = 1, 2, \ldots, n; \tag{5.17}$$

б) для некоторого плана $\mathbf{x}$ задачи (5.14) — (5.16) выполняется условие

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j, \text{ если } x_j > 0. \tag{5.18}$$

Критерий оптимальности в этом случае формулируется так. *Для оптимальности плана* $\mathbf{x}$ *задачи* (5.14) — (5.16) *необходимо и достаточно существование разрешающего вектора* Λ, *связанного с* $\mathbf{x}$ *условием* (5.17).

Итак, допустим, что ЛП-задача задана в каноническом виде, и пусть $\{\mathbf{A}_{S_1}, \mathbf{A}_{S_2}, \ldots, \mathbf{A}_{S_m}\} = \{\mathbf{A}_{S_i}\}$, $i = \overline{1, m}$ образуют базис некоторого опорного плана $\mathbf{x}^*$. Обозначим через I_x множество индексов базисных векторов. Определим вектор $\Lambda^* = (\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*)$ из системы уравнений:

$$\sum_{i=1}^{m} a_{ij}\lambda_i^* = c_j, \qquad j \in I_x. \tag{5.19}$$

Тогда опорный план $\mathbf{x}^*$ — оптимален, если

$$\sum_{i=1}^{m} a_{ij}\lambda_i^* \geqslant c_j, \qquad j = \overline{1, n}. \tag{5.20}$$

При рассмотрении симплекс-метода мы получили достаточный признак оптимальности в виде

$$\Delta_j = \sum_{i \in I_x} c_i x_{ij} - c_j \geqslant 0. \tag{5.21}$$

Покажем эквивалентность форм (5.20) и (5.21) для всех $j = \overline{1, n}$. Справедлива следующая цепочка равенств

$$z_j = \sum_{i \in I_x} c_i x_{ij} \overset{(1)}{=} \sum_{i \in I_x} \left(\sum_{\mu=1}^{m} a_{\mu i}\lambda_\mu \right) x_{ij} \overset{(2)}{=} \sum_{\mu=1}^{m} \lambda_\mu \sum_{i \in I_x} a_{\mu i} x_{ij} \overset{(3)}{=}$$

$$= \sum_{\mu=1}^{m} a_{\mu j}\lambda_\mu. \tag{5.22}$$

Равенство (1) вытекает из (5.19), равенство (2) получено переменой порядка суммирования, а равенство (3) следует из разложения небазисного вектора $\mathbf{A}_j$ через базисные:

$$\mathbf{A}_j = \sum_{i \in I_x} \mathbf{A}_i x_{ij}.$$

Таким образом, из (5.22) следует, что условие $\sum_{\mu=1}^{m} a_{\mu j}\lambda_\mu \geqslant c_j$ для всех $j = 1, 2, \ldots, n$ эквивалентно условию $\Delta_j \geqslant 0$ для $j = 1, 2, \ldots, n$.

Метод обратной матрицы (Λ-метод)

Использование признака оптимальности во второй форме (5.20) позволяет сконструировать второй алгоритм симплекс-метода, или *метод обратной матрицы* [57]. Этот алгоритм был впервые применен Л. В. Канторовичем для решения одной из частных ЛП-задач, а затем для общей задачи линейного программирования в 1951 году.

Пусть ЛП-задача задана в канонической форме и пусть $\mathbf{x}$ — опорный план задачи с базисом $\{\mathbf{A}_i\}_i\ i \in I_x$, где I_x — множество индексов базисных векторов. Составим из векторов базиса квадратную матрицу. Очевидно, определитель матрицы $\mathbf{A}_x$ отличен от 0 и существует обратная матрица $\mathbf{A}_x^{-1}$. Покажем, как с помощью матрицы $\mathbf{A}_x^{-1}$ и условий задачи можно построить удобную компактную схему для вычисления параметров, необходимых для реализации симплекс-метода.

Действительно, базисные компоненты текущего плана $\mathbf{x}$ определяются из условия $\mathbf{x} = \mathbf{A}_x^{-1}\,\mathbf{b}$ или $x_{i0} = \sum\limits_{\mu=1}^{m} a_{i\mu}^{-1} b_\mu$.

Оценки векторов условий Δ_j определяются по формулам

$$\Delta_j = z_j - c_j = \sum_{i \in I_x} \lambda_i a_{ij} - c_j, \qquad j = \overline{1,\ n}, \tag{5.23}$$

где параметры λ_i удовлетворяют уравнениям

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j \quad \text{при } j \in I_x,$$

или в векторном виде $\Lambda\mathbf{A}_x = \mathbf{c}_x$, (5.24)

где $\mathbf{c}_x$ — вектор-строка коэффициентов целевой функции, отвечающих базисным переменным. Следовательно,

$$\Lambda = \mathbf{c}_x\mathbf{A}_x^{-1}. \tag{5.25}$$

Формулы (5.23) — (5.25) позволяют, если известны элементы обратной матрицы $\mathbf{A}_x^{-1}$ и условия задачи, вычислить оценки Δ_j векторов условий. Коэффициенты x_{ik} разложения вектора $\mathbf{A}_k$ по векторам текущего базиса вычисляются как элементы произведения $(\mathbf{A}_x^{-1}\mathbf{A}_k)$ матрицы $\mathbf{A}_x^{-1}$ на вектор условий $\mathbf{A}_k$. Коэффициенты x_{ik} вместе с базисными составляющими опорного плана определяют вектор, подлежащий выводу из базиса.

Итак, для реализации метода достаточно уметь вычислять на каждом шаге матрицу $\mathbf{A}_x^{-1}$. Элементы столбцов матрицы $\mathbf{A}_x^{-1}$ удобно рассматривать, как коэффициенты $\mathbf{e}_{ij}$ разложения единичных векторов $\mathbf{e}_j$ $(j = \overline{1,\ m})$ по векторам базиса, где

$$\mathbf{e}_j^T = \underbrace{[\underbrace{0,\ \ldots,\ 0,\ 1}_{j},\ 0,\ \ldots\ 0]}_{m}.$$

При этом пересчет элементов $\{\mathbf{e}_{ij}\}$ матрицы $\mathbf{A}_x^{-1}$ при вводе очередного вектора $\mathbf{A}_k$ производится по следующим рекуррентным формулам метода исключения Жордана — Гаусса:

$$e'_{ij} = \begin{cases} e_{ij} - \dfrac{e_{rj}}{x_{rk}} x_{ik}, & \begin{array}{l} \text{при } i \neq r \\ i = 1, 2, \ldots, m+1 \\ j = 0, 1, \ldots, m \end{array} \\ \dfrac{e_{rj}}{x_{rk}}, & \begin{array}{l} \text{при } i = r \\ j = 0, 1, 2, \ldots, m, \end{array} \end{cases} \tag{5.26}$$

где а) x_{rk} — направляющий элемент преобразования; б) $e_{i0} = x_{i0}$ — базисные компоненты опорного плана $x_0 = \| x_{i0} \|$ $i = \overline{1, m}$; в) $e_{m+1,j} = \lambda_j$ — разрешающие множители (относительные оценки условий задачи), $j = \overline{1, m}$; г) $e_{m+1,0} = L(x_0)$ — значение линейной формы при плане x_0, A_k — вектор, вводимый в базис, A_r — выводимый из базиса.

Для реализации метода обратной матрицы на каждой итерации используется основная таблица и вспомогательная.

Основная таблица состоит из $(m+2)$-х столбцов и $(m+1)$-й строк.

В столбце e_0 записываются базисные компоненты текущего плана, столбцы $e_1, e_2, \ldots, e_m$ суть столбцы обратной матрицы A_x^{-1} текущего базиса; в столбце A_k записаны коэффициенты x_{ik} разложения небазисного вектора A_k по векторам базиса; столбец θ — вспомогательный, содержит величины $\theta_i = \dfrac{e_{i0}}{x_{ik}} \Big| \, x_{ik} > 0$. Столбцы $e_0, e_1, \ldots, e_m$ составляют главную часть таблицы, в $(m+1)$-й строке основной таблицы записывают величины $\{\lambda_j\}$ $j \in I_x$.

Вспомогательная таблица содержит векторы $A_1, A_2, \ldots, A_n$ исходной задачи, а также вектор c. В строке 0 записываются величины оценок Δ_j, $j = \overline{1, n}$, отвечающие начальному опорному плану.

В процессе каждой очередной итерации во вспомогательной таблице дописывается очередная строка (l), содержащая величины $\Delta_j^{(l)}$.

Опишем произвольную $(l+1)$-ю итерацию алгоритма обратной матрицы.

Пусть уже проведено l итераций, в ходе которых вычислены матрица $A_x^{-1}(l)$ и оценки $\Delta_j^{(l)}$, $j = \overline{1, n}$ и еще не найдено оптимальное решение.

1. Определяем вектор A_k, вводимый в базис из условия

$$\Delta_k^{(l)} = \min_j \left(\Delta_j^{(l)} \mid \Delta_j^{(l)} < 0\right). \tag{5.27}$$

2. Из вспомогательной таблицы выбираем компоненты вектора A_k. Определяем его коэффициенты разложения x_{ik} через базисные, используя обратную матрицу A_x^{-1}:

$$x_{ik}^{(l)} = \sum_j e_{ij}^{(l)} a_{ik}, \tag{5.28}$$

и записываем величины x_{ik} в столбец A_k. В $(m+1)$-ю позицию столбца A_k записываем оценку $\Delta_k^{(l)}$ этого вектора. Просматриваем столбец

Основная таблица

N	c_x	$\mathbf{B}_x$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	...	$\mathbf{e}_m$	$\mathbf{A}_k$	Θ
1	c_1	A_1	$\mathbf{e}_{10}^{(l)}$	$\mathbf{e}_{11}^{(l)}$	$\mathbf{e}_{22}^{(l)}$	...	$\mathbf{e}_{1m}^{(l)}$	x_{1k}	
2	c_2	A_2	$\mathbf{e}_{20}^{(l)}$	$\mathbf{e}_{21}^{(l)}$	$\mathbf{e}_{22}^{(l)}$	...	$\mathbf{e}_{2m}^{(l)}$	x_{2k}	
...	...	...	...	...	...	...	...	...	...
r	c_r	A_r	$\mathbf{e}_{r_0}^{(l)}$	$\mathbf{e}_{r1}^{(l)}$	$\mathbf{e}_{r2}^{(l)}$	...	$\mathbf{e}_{rm}^{(l)}$	x_{rk}	Θ_r
...	...	...	...	...	...	...	...	...	...
m	c_m	A_m	$\mathbf{e}_{m_0}^{(l)}$	$\mathbf{e}_{m1}^{(l)}$	$\mathbf{e}_{m2}^{(l)}$	...	$\mathbf{e}_{mm}^{(l)}$	x_{mk}	Θ_m
$m+1$			L	λ_1	λ_2	...	λ_m	Δ_k	

Вспомогательная таблица.

N	$\mathbf{b}=\mathbf{A}_0$	$\mathbf{A}_1$	$\mathbf{A}_2$	...	$\mathbf{A}_k$	...	$\mathbf{A}_{n-1}$	$\mathbf{A}_n$
1				...		...		
2				...		...		
...	...	...	...	...	...	...	...	...
m				...		...		
$m+1$	c	c_1	c_2	...	c_k	...	c_{n-1}	c_n
0	Δ	$\Delta_1^{(0)}$	$\Delta_2^{(0)}$	...	$\Delta_k^{(0)}$	...	$\Delta_{n-1}^{(0)}$	Δ_n^0
1	$\Delta^{(1)}$	$\Delta_1^{(1)}$	$\Delta_2^{(1)}$	...	$\Delta_k^{(1)}$	...	$\Delta_{n-1}^{(1)}$	$\Delta_n^{(1)}$
2	$\Delta^{(2)}$	$\Delta_1^{(2)}$	$\Delta_2^{(2)}$	...	$\Delta_k^{(2)}$	...	$\Delta_{n-1}^{(2)}$	$\Delta_n^{(2)}$
...	...	...	...	...	...	...	...	...
l	$\Delta^{(l)}$	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	...	$\Delta_k^{(l)}$	...	$\Delta_{n-1}^{(l)}$	$\Delta_n^{(l)}$

A_k. Если все $x_{ik}^{(l)} \leqslant 0$, то задача неразрешима. Допустим, что имеется хотя бы один $x_{ik}^{(l)} > 0$. Переходим к п. 3.

3. Вычисляем отношения $\frac{e_{i0}}{x_{ik}}$ и определяем направляющую строку r из условия:

$$\theta_r = \min_i \left\{ \frac{e_{i0}}{x_{ik}} \middle| x_{ik} > 0 \right\}.$$

4. Выполняем одну итерацию симплекс-метода с главной частью таблицы и направляющим элементом $x_{rk}^{(l)}$.

5. Получим новую обратную матрицу $A_x^{-1}(l+1)$, а также новые оценки $\lambda_i^{(l+1)}$; $i \in I_x$. Заполним элементами $A_x^{-1}(l+1)$ главную часть таблицы $(l+1)$-й итерации, при этом заменим индекс r вектора, выведенного из базиса на k в столбце B_x, одновременно заменим c_r на c_k в столбце c_x. Заметим, что для контроля величины $\lambda_j\ (l+1)$ можно вычислить также и по формулам:

$$\lambda_j^{(l+1)} = \sum_{i \in I_x} c_i e_{ij}^{(l+1)}. \tag{5.29}$$

6. Для всех небазисных векторов вычисляем оценки согласно соотношениям

$$\Delta_j^{(l+1)} = \sum_{i \in I_x} a_{ij} \lambda_i^{(l+1)} - c_j, \qquad j = \overline{1, n}. \tag{5.30}$$

Проверяем условие $\Delta_j^{l+1} \geqslant 0$ для всех $j = \overline{1, n}$. Если это выполняется, то конец, иначе переходим к $(l+2)$-й итерации.

Все таблицы алгоритма заполняются по одним и тем же правилам. Некоторые особенности возникают лишь при составлении начальной основной таблицы. Здесь в столбец e_0 записывают базисные компоненты исходного плана. Первые m позиций столбцов $e_1, e_2, \ldots, e_m$ получаются путем обращения матрицы векторов базиса исходного опорного плана. Последняя строка главной части таблицы 0 заполняется произведениями столбца c_x и столбцов e_j этой таблицы, т. е.

$$\lambda_j^{(0)} = \sum_{i \in I_x} c_i e_{ij}, \qquad j = 1, 0, \ldots, m. \tag{5.31}$$

Для контроля вычислений можно использовать две возможности для определения величин $\{\lambda_j^{(l+1)}\}$, $j = \overline{1, m}$.

В любой итерации эти параметры могут быть вычислены как по рекуррентным формулам (5.26) при переходе от таблицы l-й итерации к $(l+1)$-й, так и непосредственно по формуле (5.29).

Мультипликативная форма алгоритма обратной матрицы

В рассмотренной выше вычислительной схеме метода обратной матрицы необходимо запоминать обратную матрицу $A_x^{-1} = \| e_{ij} \|$.

В так называемой *мультипликативной* форме алгоритма требуется регистрировать значительно меньший объем данных. Рассмотрим ее идею. Пусть **X** и **X′** — два последовательных опорных плана задачи.

Обозначим соответствующие матрицы векторов базиса через A_x и $A_{x'}$.

$$A_x = [A_{S_1}, A_{S_2}, \dots, A_{S_r}, \dots, A_{S_m}];$$

$$A_x' = [A_{S_1}, A_{S_2}, \dots, A_k, A_{S_{r+1}}, A_{S_m}].$$

Легко проверить, что обратные матрицы A_x^{-1} и $A_{x'}^{-1}$ связаны соотношением:

$$A_{x'}^{-1} = E_r A_x^{-1}, \tag{5.32}$$

$$E_r = \begin{vmatrix} 1 & 0 & \dots & y_{1k} & \dots & 0 \\ 0 & 1 & \dots & y_{2k} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & y_{rk} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \multicolumn{4}{c}{\underbrace{0 \quad 0 \ \dots \ y_{mk}}_{r}} & \dots & 1 \end{vmatrix},$$

где

$$y_{ik} = -\frac{x_{ik}}{x_{rk}}; \quad i = 1, 2, \dots, m; \quad i \neq r; \quad y_{rk} = \frac{1}{x_{rk}}.$$

Соотношение (5.32) эквивалентно применению рекуррентных формул (5.26) при $i = 1, 2, \dots, m$ и $j = 1, 2, \dots, m$.

Обычно решение задачи ЛП начинается с единичного базиса. Ему соответствует единичная матрица E. Поэтому после первой итерации, когда вместо вектора A_{S_r} в базис вводится A_k, обратная матрица базиса полученного опорного плана X' может быть вычислена по формуле

$$A_{x'}^{-1} = E_{r_1} E,$$

а после l-й итерации:

$$A_{x'}^{-1} = E_{r_l} E_{r_{l-1}} E_{r_1} E. \tag{5.33}$$

Матрица E_r определяется $(m + 1)$-м числом $(r, y_{1k}, y_{2k}, \dots, y_{mk})$. Следовательно, при $l < m$ запись матрицы A_x^{-1} в форме (5.32) меньше загружает память, чем обычная запись, где на каждой итерации необходимо помнить m^2 чисел e_{ij}.

Опишем кратко схему вычислений в отдельной итерации мультипликативной формы алгоритма обратной матрицы.

Пусть уже проведено l итераций, в результате которых установлено, что $x^{(l)}$ — неоптимальный план. Определение вектора, подлежащего вводу в базис и выводу из него, производится по общим правилам симплекс-метода, а компоненты очередного опорного плана вычисляются по рекуррентным формулам (5.32). Вектор относительных оценок условий задачи Λ на l-й итерации вычисляется по соотношению:

$$\Lambda_{l+1} = c_x E_{r_l} E_{r_{l-1}} \dots E_{r_1} E.$$

Вектор оценок $\Delta^{(l)}$ определяется обычным способом

$$\Delta = \Lambda A_x - c_x.$$

Наконец, коэффициенты x_{ik} разложения вектора $\mathbf{A}_k$, подлежащего включению в базис по векторам базиса, определяются из соотношения

$$\mathbf{A}_k^{(l+1)} = \mathbf{E}_{r_l}\mathbf{E}_{r_{l-1}} \ldots \mathbf{E}_{r_1}\mathbf{A}_k. \qquad (5.34)$$

Как видим, мультипликативная форма алгоритма позволяет не только экономить память ЭВМ, но и сокращает объем вычислений на начальных итерациях алгоритма обратной матрицы.

Пример 2.9. Решить методом обратной матрицы следующую задачу

$$\max (4x_1 + 2x_2), \qquad (1)$$

при условии

$$\begin{cases} x_1 + 2x_2 \leqslant 6; \\ x_1 + x_2 \leqslant 9; \\ 3x_1 - x_2 \leqslant 15; \end{cases}$$
$$x_1,\ x_2 \geqslant 0. \qquad (2)$$

Приводим ее к расширенной форме и получаем $\max (4x_1 + 2x_2)$

$$x_1 + 2x_2 + 1x_3 = 6;$$
$$x_1 + x_2 + 1x_4 = 9;$$
$$3x_1 - x_2 + 1x_5 = 15.$$

В качестве начального базиса выберем единичный базис

$$[\mathbf{A}_3,\ \mathbf{A}_4,\ \mathbf{A}_5] = \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Заполняем вспомогательную табл. 2.18 (начальную часть) и основную таблицу первой итерации — (табл. 2.19).

Первая итерация. Заполнив главную часть (табл. 2.19), вычисляем ее индексную строку согласно соотношениям $\lambda_j^{(0)} = \Sigma c_i\ \mathrm{e}_{ij}^{(0)}$. Так как $c_i = 0$ при $i = 3,\ 4,\ 5$, то $\lambda_1^{(0)} = \lambda_2^{(0)} = \lambda_3^{(0)} = 0$. Используя строку $\Delta_j^{(0)}$ табл. 2.18, выбираем вектор A_1, так как $\Delta_1^{(0)} = \min\limits_j \{\Delta_j \mid \Delta_j < 0\} = -4$.

Записываем вектор A_1 в столбец A_k справа от главной части табл. 2.19 и его оценку в индексную строку, Столбец A_k направляющий. Находим направляющий элемент согласно общим правилам симплекс-метода и, выполнив итерацию, заполняем главную часть табл. 2.20.

Итак, найден вектор относительных оценок $\Lambda^{(1)} = \left[0\ \ 0\ \ \frac{4}{3}\right]$. Найдем оценки $\Delta_j^{(1)}$ для всех небазисных векторов, используя соотношение (5.30)

$$\Delta_j^{(1)} = \sum_{i \in Ix} a_{ij}\lambda_i^{(1)} - c_j,$$

и результаты записываем в строку $\Delta_j^{(1)}$ табл. 2.18. Например,

$$\Delta_2 = 2\lambda_1^{(1)} + 1\lambda_2^{(1)} - 1\lambda_3^{(1)} - c_2 = -1 \cdot 4/3 - 2 = -\frac{10}{3}.$$

Вторая итерация. Из индексной строки $\Delta_j^{(1)}$ выбираем вектор A_2 с наибольшей отрицательной оценкой $\Delta_2^{(1)} = -\frac{10}{3} < 0$. Умножив матрицу $[\mathrm{e}_1,\ \mathrm{e}_2,\ \mathrm{e}_3]$ табл. 2.20

Таблица 2.18

N	B	A_1	A_2	A_3	A_4	A_5
1	6	-1	2	1		
2	9	1	1		1	
3	15	3	-1			1
Итерации	c_j	4	2	0	0	0
0	$\Delta_j^{(0)}$	-4	-2	0	0	0
1	$\Delta_j^{(1)}$	0	$-10/3$	0	0	4/3
2	$\Delta_j^{(2)}$	0	0	0	$2\frac{1}{2}$	1/2

Таблица 2.19

N	c_x	B_x	e_0	e_1	e_2	e_3	A_k
1	0	A_3	6	1			-1
2	0	A_4	9		1		1
3	0	A_5	15			1	$\boxed{3}$
			0	0	0	0	-4

Таблица 2.20

N	c_x	B_x	e_0	e_1	e_2	e_3	A_k
1	0	A_3	11	1	1	$\frac{1}{3}$	$2\frac{2}{3}$
2	0	A_4	4		1	$-\frac{1}{3}$	$\boxed{1\frac{1}{3}}$
3	4	A_1	5			$\frac{1}{3}$	$-\frac{1}{3}$
—	—		20	0	0	4/3	$-\frac{10}{3}$

на A_2, получим A_2^1, который записываем в столбец A_k табл. 2.20. Например, $a'_{12} = \sum_{j=1}^{3} e_{1j}A_{j2} = 1 \cdot 2 + 1 \cdot 1 + 1/3 \cdot (-1) = 2^2/_3$. Определив направляющий элемент в столбце A_k $(a_{2k}=1^1/_3)$ и выполнив очередную итерацию симплекс-метода, придем к табл. 2.21. Находим $\Lambda = \left[0\; 2\frac{1}{2}\; \frac{1}{2}\right]$, и, вычислив оценки $\Delta_j^{(2)}$, заносим их в табл. 2.18. Так как все $\Delta_j^{(2)} \geqslant 0$, то текущее базисное решение оптимально.

Таблица 2.21

N	c_x	B_x	e_0	e_1	e_2	e_3
1	0	A_3	3	1	-1	1
2	2	A_2	3	0	3/4	$-1/4$
3	4	A_1	6		1/4	1/4
			30	0	$2\frac{1}{2}$	1/2

Итак, $x_{1\text{опт}} = 6$, $x_{2\text{опт}} = 3$, $x_{3\text{опт}} = 3$.

Оптимальные значения двойственных переменных равны $y_{1\text{опт}} = \lambda_1 = 0$; $y_{2\text{опт}} = \lambda_2 = 2\frac{1}{2}$; $y_{3\text{опт}} = \lambda_3 = 1/2$.

§ 6. ДВОЙСТВЕННЫЙ СИМПЛЕКС-МЕТОД

Использование идей двойственности в сочетании с общей идеей симплекс-метода позволило разработать еще один метод решения ЛП-задач — так называемый *двойственный симплекс-метод*, или *метод последовательного уточнения оценок* [18, 24]. Впервые этот метод был предложен Лемке в 1954 году.

Решение задачи ЛП методом последовательного уточнения оценок сводится к отысканию оптимального плана двойственной задачи посредством перехода от одного ее опорного плана к другому.

Задача ЛП в канонической форме имеет вид:

$$\max L(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j \tag{6.1}$$

при условии

$$\sum_{j=1}^{n} a_{\mu j} x_j = b_\mu \quad (\mu = 1, 2, \ldots, m); \quad \sum_{j=1}^{n} \mathbf{A}_j x_j = \mathbf{b} \quad (n \geqslant m). \tag{6.2}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n$$

Предположим, что ранг матрицы $\mathbf{A}$ равен m. Двойственная задача записывается так:

$$\min \tilde{L}(\mathbf{y}) = \sum_{\mu=1}^{m} b_\mu y_\mu \tag{6.3}$$

при условии

$$\sum_{\mu=1}^{m} a_{\mu j} y_\mu \geqslant c_j; \quad \mathbf{A}_j^T \mathbf{y} \geqslant c_j; \qquad (j = 1, 2, \ldots, n). \tag{6.4}$$

Назовем *сопряженным базисом,* или базисом двойственной задачи, систему из m линейно-независимых векторов матрицы ограничений прямой задачи $\{\mathbf{A}_i\}_{i \in I_б}$, базисное решение которой $\mathbf{y}$, определяемое из условия

$$\mathbf{A}_i^T \mathbf{y} = c_i, \tag{6.5}$$

удовлетворяет всем ограничениям (6.4).

Разложим вектор $\mathbf{b}$ по сопряженному базису:

$$\sum_{i \in I_б} \mathbf{A}_i x_i = \mathbf{b} = \mathbf{A}_0. \tag{6.6}$$

Решив (6.6), получим некоторое базисное решение $\{x_{i0}\}_{i \in I_б}$, которое называется псевдопланом прямой задачи, так как здесь может быть нарушено условие неотрицательности.

Таким образом, *псевдоплан* прямой задачи есть *базисное решение относительно сопряженного базиса.*

Введем обозначение

$$\Delta_j = \sum_{i \in I_б} c_i x_{ij} - c_j. \tag{6.7}$$

Псевдоплан можно найти и независимо от двойственной задачи. Пусть $\{\mathbf{A}_i\}$ $i \in I_б$ — произвольная система линейно-независимых векторов прямой задачи.

Выразим все небазисные векторы $\{\mathbf{A}_j\}$ через базисные

$$\mathbf{A}_j = \sum_{i \in I_б} x_{ij} \mathbf{A}_i, \tag{6.8}$$

$$\mathbf{A}_0 = \mathbf{b} = \sum_{i \in I_б} \mathbf{A}_i x_i. \tag{6.9}$$

Обозначим x_{i0} — решение уравнения (6.9).

Тогда можно дать дополнительное определение псевдоплана: n-мерный вектор $\mathbf{x}$, для которого $x_i = x_{i0}$ при $i \in I_б$ и $x_j = 0$ при $j \notin I_б$, является псевдопланом тогда и только тогда, когда все $\Delta_j \geqslant 0$ $(j = 1, 2, \ldots, n)$.

Доказательство. Векторы $\{\mathbf{A}_i\}$ $i \in I_б$ линейно независимы. Поэтому можно вычислить такой $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$, для которого

$$\mathbf{A}_i^T \mathbf{y} = \sum_{\mu=1}^{m} a_{\mu i} y_\mu = c_i, \qquad i \in I_б. \tag{6.10}$$

Тогда

$$\Delta_j = \sum_{i\in I_б} c_i x_{ij} - c_j = \sum_{i\in I_б}^{m} \left(\sum_{\mu=1}^{m} a_{\mu i} y_\mu\right) x_{ij} - c_j = \sum_{\mu=1}^{m} \left(\sum_{i\in I_б} a_{\mu i} x_{ij}\right) y_\mu - c_j =$$

$$= \sum_{\mu=1}^{m} a_{\mu j} y_\mu - c_j.$$

С учетом (6.4) получим

$$\Delta_j = \sum_{\mu=1}^{m} a_{\mu j} y_\mu - c_j \geqslant 0, \qquad \textbf{(6.11)}$$

что и требовалось доказать.

Рассмотрим некоторый сопряженный базис и соответствующий ему псевдоплан $\{x_{i0}\}$.

Справедлив следующий признак оптимальности: *если среди базисных компонентов псевдоплана* x *нет отрицательных, то псевдоплан* x *оказывается оптимальным решением прямой задачи, а опорный план* y — *оптимальным решением двойственной задачи.*

Доказательство. Имеет место цепочка равенств

$$\tilde{L}_{дв}(\mathrm{y}) = \sum_{\mu=1}^{m} b_\mu y_\mu \overset{(1)}{=} \sum_{\mu=1}^{m} \left(\sum_{i\in I_б} a_{\mu i} x_i\right) y_\mu \overset{(2)}{=} \sum_{i\in I_б} x_i \sum_{\mu=1}^{m} a_{\mu i} y_\mu \overset{(3)}{=} \sum_{i\in I_б}^{m} c_i x_i.$$

Равенство (1) следует из (6.2), равенство (2) получено переменой порядка суммирования, равенство (3) следует из (6.10). Так как

$$x_j = 0 \quad \text{при } j \neq i \in I_б, \qquad \textbf{(6.12)}$$

то

$$\sum_{i\in I_б} c_i x_i = \sum_{j=1}^{n} c_j x_j = L(\mathrm{x}). \qquad \textbf{(6.13)}$$

Таким образом, $L_{дв}(\mathrm{y}) = L(\mathrm{x})$, что и является признаком оптимальности планов x и y при условии, что $\mathrm{x} \geqslant 0$ (см. теорему 2.5).

Этот признак является необходимым и достаточным в случае невырожденности опорного плана y двойственной задачи [18].

Пусть известен некоторый сопряженный базис $\{A_i\}$ $i \in I_б$, которому соответствует псевдоплан x.

Очевидно, $\mathrm{A}_j = \sum_{i\in I_б} \mathrm{A}_i x_{ij}$; $\mathrm{A}_0 = \sum_{i\in I_б} \mathrm{A}_i x_i$. При этом в зависимости от знаков $\{x_i\}$ и $\{x_{ij}\}$ может иметь место один из трех случаев:

1) базисные компоненты $x_i = x_{i0} \geqslant 0$ для всех $i \in I_б$;

2) среди x_i имеются отрицательные и для некоторого i $x_{i0} < 0$, а все $x_{ij} \geqslant 0$ $(j = 1, 2, \ldots, n)$;

3) псевдоплан содержит отрицательные компоненты $x_i < 0$, но для каждой из них среди элементов $\{x_{ij}\}$ $j = 1, 2, \ldots, n$ имеются отрицательные.

В первом случае, как следует из достаточного признака оптимальности, псевдоплан x — оптимальный опорный план. Во втором случае задача неразрешима. В третьем случае можно перейти к некоторому новому сопряженному базису и, следовательно, к новому псевдоплану x_{l+1} с меньшим значением L.

Итак, последовательные переходы от одного сопряженного базиса к другому производят до тех пор, пока не получат решение задачи или не установят ее неразрешимость. Каждый переход от одного псевдоплана к другому составляет одну итерацию (один шаг) двойственного симплекс-метода.

Каждая итерация содержит два этапа. На первом этапе выясняют, не является ли псевдоплан планом прямой задачи, и если нет, то разрешима ли задача. Для этого необходимо вычислить $\{x_i\}$, $i \in I_б$ и установить их знаки.

Второй этап состоит в выборе и осуществлении элементарного преобразования, приводящего к новому опорному плану сопряженной задачи с меньшим значением целевой функции. На этом этапе определяют *направляющую строку* r *с отрицательной базисной компонентой* $x_{r_0} < 0$ и вектор $\mathbf{A}$, который должен заменить в сопряженном базисе исключаемый вектор $\mathbf{A}_r$.

Описание алгоритма. Задачу линейного программирования предполагают заданной в канонической форме (6.1), (6.2). Отыскивают сопряженный базис двойственной задачи и обозначают $\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m\}$. Разлагая $\mathbf{A}_0$ по векторам $\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m\}$ в соответствии с (6.9), находим псевдоплан $\{x_{i0}\}$ $i = \overline{1, m}$ прямой задачи.

Исследуем знаки $\{x_{i0}\}$. Если имеет место случай $x_{i0} \geqslant 0$, $\forall i \in I_б$, то начальный псевдоплан $\mathbf{x}_0$ является оптимальным планом прямой задачи. При наличии отрицательных компонент $\{x_{i0}\}$ вычисляем величины $\{x_{ij}\}$ — коэффициенты разложения векторов $\mathbf{A}_j$ по сопряженному базису в соответствии с (6.8).

Если для некоторого x_r, такого, что $x_{r0} < 0$, все $x_{rj} \geqslant 0$, $j = \overline{1, n}$, то задача не разрешима (второй случай), и на этом процесс вычислений заканчивается.

Если имеет место третий случай (для каждого r, такого, что $x_{r0} < 0$ по крайней мере одна из компонент $x_{rj} < 0$), то переходим ко второму этапу. С этой целью составляют таблицу k-й итерации (аналогичную симплекс-таблице), которая состоит из $(m + 2)$ строк и $(n + 1)$-го столбца (табл. 2.22).

Столбец B_x таблицы, как обычно, содержит векторы $\{\mathbf{A}_i\}$ базиса псевдоплана $\mathbf{x}_k$, а столбец $\mathbf{A}_0$ — базисные компоненты псевдоплана $\{x_{i0}^{(k)}\}$.

Строку $(m + 1)$ заполняют параметрами $\Delta_j^{(k)}$, являющимися оценками векторов $\mathbf{A}_j$:

$$\Delta_j = a_{0j} = \sum_{i \in I_б} c_i x_{ij} - c_j.$$

Величина Δ_0 — значение целевой функции при данном псевдоплане:

$$\Delta_0 = \sum_{i \in I_б} c_i x_{i0}^{(k)}.$$

Итерацию k завершают заполнением главной части таблицы (от первой до $(m + 1)$-й строк).

Таблица 2.22

	c_x	B_x	A_0	A_1	A_2	...	A_k	...	A_n
1	c_1	A_1	$x_{10}^{(k)}$	$x_{11}^{(k)}$	$x_{12}^{(k)}$	...	$x_{1k}^{(k)}$	...	$x_{1n}^{(k)}$
2	c_2	A_2	$x_{20}^{(k)}$	$x_{21}^{(k)}$	$x_{22}^{(k)}$	...	$x_{2k}^{(k)}$	...	$x_{2n}^{(k)}$
r	c_r	A_r	x_{r0}	$x_{r1}^{(k)}$	$x_{r2}^{(k)}$	...	$x_{rk}^{(k)}$	...	$x_{rn}^{(k)}$
...	...	...	...	...	...	...	...	...	...
m	c_m	A_m	$x_{m0}^{(k)}$	$x_{m1}^{(k)}$	$x_{m2}^{(k)}$	...	$x_{mk}^{(k)}$	...,	$x_{mn}^{(k)}$
$m+1$		Δ	$\Delta_0^{(k)}$	$\Delta_1^{(k)}$	$\Delta_2^{(k)}$		$\Delta_m^{(k)}$	...	$\Delta_n^{(k)}$
$m+2$		θ							

На первом этапе $(k+1)$-й итерации выясняют, имеет ли место первый, второй или третий случай.

В третьем случае переходим ко второму этапу.

Сначала определяют вектор A_r, который необходимо вывести из базиса. Его индекс r определяют из условия:

$$x_{r0} = \min_i \{x_{i0}/x_{i0} < 0\}, \tag{6.14}$$

т. е. по максимальной по модулю отрицательной компоненте базисного решения.

Затем заполняют элементы $\{\theta_j^{(k)}\}$ $(m+2)$-й строки, которые вычисляют по формуле:

$$\theta_j^{(k)} = -\frac{\Delta_j}{x_{rj}}\bigg|_{x_{rj}<0}. \tag{6.15}$$

В строке θ заполняют лишь те позиции, для которых $x_{rj} < 0$.

Вектор A_l, который должен быть введен в базис, находят из условия:

$$\theta_l = \min_j \{\theta_j\} = \min_j \left\{-\frac{\Delta_j}{x_{rj}}\bigg|_{x_{rj}<0}\right\}.$$

Определив направляющую строку r и столбец l, вычисляют элементы главной части таблицы $(k+1)$-й итерации по рекуррентным соотношениям:

$$x_{ij}^{(k+1)} = \begin{cases} x_{ij}^{(k)} - \dfrac{x_{rj}^{(k)}}{x_{rl}^{(k)}} x_{il}^{(k)} & \text{при } i \neq r \\ \dfrac{x_{rj}^{(k)}}{x_{rl}^{(k)}} & \text{при } i = r, \end{cases} \qquad (6.16)$$

где x_{rl} — направляющий элемент преобразования.

Вычислительная схема алгоритма двойственного симплекс-метода похожа на вычислительную схему симплекс-метода. Аналогичны и формы таблиц.

Различие между методами заключается в том, что при симплекс-методе производят последовательный переход от одного опорного плана задачи к другому, а при двойственном симплекс-методе — переход от одного псевдоплана к другому.

Формальное различие между вычислительными схемами этих методов проявляется только в правилах перехода от одного базиса к следующему и в признаках оптимальности плана и неразрешимости задачи. В симплекс-методе сначала определяют вектор, вводимый в базис, а затем вектор, исключаемый из базиса, а в двойственном симплекс-методе этот порядок оказывается обратным.

Отметим некоторые важные свойства двойственного симплекс-метода.

1. В отличие от прямого симплекс-метода для своего применения он не требует нахождения начального д. б. р. (опорного плана), а поиск начального псевдоплана часто может оказаться легче, чем поиск д. б. р. Рассмотрим, например, типичную задачу минимизации

$$\min \sum_{j=1}^{n} c_j x_j \qquad (6.17)$$

при

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \qquad i = \overline{1, m}; \qquad (6.18)$$

$$x_j \geqslant 0, \qquad j = \overline{1, n}; \qquad (6.19)$$

и пусть

$$c_j \geqslant 0, \qquad j = \overline{1, n}. \qquad (6.20)$$

Для задачи такого вида найти сразу начальный опорный план нельзя, и необходимо применить метод искусственных переменных и проделать значительный объем вычислений. В то же время псевдоплан находится почти автоматически.

Действительно, перейдем от (6.17) — (6.19) к эквивалентной задаче в расширенной форме, введя свободные переменные $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$:

$$\max \sum_{j=1}^{n} -c_j x_j; \qquad (6.21)$$

$$\sum_{j=1}^{n} a_{ij} x_j - 1 x_{n+i} = b_i, \qquad i = \overline{1, m}. \qquad (6.22)$$

Запишем ограничения двойственной задачи:

$$\begin{cases} \sum_{i=1}^{m} a_{ij}y_i \geqslant -c_j, & j=\overline{1,n}\ ; \quad (6.23) \\ -y_i \geqslant 0, & i = 1,\ m. \quad (6.24) \end{cases}$$

Из (6.23), (6.24) видим, что, поскольку решение $y_i = 0$, $i = \overline{1, m}$ удовлетворяет всем ограничениям (6.23), сопряженный базис образуют вектора $\mathbf{A}_{n+1}$, $\mathbf{A}_{n+2}$, ..., $\mathbf{A}_{n+m}$ при свободных переменных. При этом начальный псевдоплан будет равен $x_{n+i} = -b_i$, $i = \overline{1,\ m}$.

Итак, для задачи вида (6.17) — (6.19) при условии (6.20) применение двойственного симплекс-метода оказывается предпочтительнее в сравнении с прямым.

2. Двойственный симплекс-метод позволяет в процессе итерации добавлять новые дополнительные ограничения к уже найденному некоторому промежуточному решению. Это важное его свойство будет использовано при решении задач целочисленного программирования.

Пример 2.10. Решение задачи линейного программирования двойственным симплекс-методом.

Найти max $(x_1 + x_2)$
при ограничениях

$$\begin{cases} 2x_1 + 11x_2 \leqslant 38; \\ x_1 + x_2 \leqslant 7; \\ 4x_1 - 5x_2 \leqslant 5; \end{cases}$$

$$x_1,\ x_2 \geqslant 0$$

или в расширенной форме

$$\begin{cases} 2x_1 + 11x_2 + 1x_3 + 0x_4 + 0x_5 = 38; \\ 1x_1 + 1x_2 + 0x_3 + 1x_4 + 0x_5 = 7; \\ 4x_1 - 5x_2 + 0x_3 + 0x_4 + 1x_5 = 5, \quad x_1,\ x_2,\ \ldots,\ x_5 \geqslant 0. \end{cases}$$

Двойственная задача записывается следующим образом:

$$\min\,(38y_1 + 7y_2 + 5y_3)$$

при ограничениях

$$\begin{cases} 2y_1 + 1y_2 + 4y_3 \geqslant 1 & (A_1); \\ 11y_1 + 1y_2 - 5y_3 \geqslant 1 & (A_2); \\ y_1 \geqslant 0 & (A_3); \\ y_2 \geqslant 0 & (A_4); \\ y_3 \geqslant 0 & (A_5). \end{cases}$$

Выбираем в качестве базиса векторы $\{\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_5\}$. Тогда решением системы линейных уравнений

$$\begin{cases} 2y_1 + 1y_2 + 4y_3 = 1; \\ y_1 = 0; \\ y_3 = 0 \end{cases}$$

является $y_1 = 0$; $y_2 = 1$; $y_3 = 0$.

Подставив это решение в ограничения $(\mathbf{A}_2)$ и $(\mathbf{A}_4)$, замечаем, что они также удовлетворяются, а потому $\{\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_5\}$ — сопряженный базис двойственной задачи.

Находим псевдоплан $\mathbf{X}_0$ прямой задачи. Для этого решим систему уравнений $\mathbf{A}_0 = \mathbf{A}_1x_{10} + \mathbf{A}_3x_{30} + \mathbf{A}_5x_{50}$. Отсюда $x_{10} = 7$; $x_{30} = 24$; $x_{50} = -23$.

Таблица 2.23

c_i			1	1	0	0	0
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
1	A_1	7	1	1	0	1	0
0	A_3	24	0	9	1	—2	0
0	A_5	—23	0	$\boxed{-9}$	0	—4	1
		7	0	0	0	1	0
	0			0		$\frac{1}{4}$	

Вычисляем коэффициенты разложения $\{x_{ij}\}$:

$$A_2 = A_1x_{12} + A_3x_{32} + A_5x_{52},$$

$$\begin{bmatrix} 11 \\ 1 \\ -5 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} x_{12} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_{32} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_{52}$$

и находим $x_{12} = 1$, $x_{32} = 9$, $x_{52} = -9$.

Аналогично $$A_4 = A_1x_{14} + A_3x_{34} + A_5x_{54},$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} x_{14} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_{34} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_{54};$$

$x_{14} = 1$, $x_{34} = -2$, $x_{54} = -4$.

Таблица 2.24

c_i			1	1	0	0	0
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
1	A_1	$^{40}/_9$	1	0	0	$^5/_9$	$^1/_9$
0	A_3	1	0	0	1	—6	1
4	A_2	$^{23}/_9$	0	1	0	$^4/_9$	$-^1/_9$
	Δ	7	0	0	0	1	0

Теперь можно составить симплекс-таблицу следующего вида (табл. 2.23).

Первый шаг. Определяем направляющую строку. Это строка A_4. Находим направляющий столбец, для чего заполняем строку θ. Направляющий столбец A_2, так как $\theta_2 = 0 < \theta_4 = \frac{1}{4}$.

Следовательно, направляющий элемент $x_{42} = -9$. Выполнив первый шаг симплекс-преобразования, составляем следующую табл. 2.24.

Так как все элементы столбца A_0 $x_{i0} \geqslant 0$, то найден оптимальный план, причем $x_{1\text{опт}} = \frac{40}{9}$; $x_{2\text{опт}} = \frac{23}{9}$.

Целевая функция $L_{\max} = 7$.

§ 7. ДЕКОМПОЗИЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ БОЛЬШОЙ РАЗМЕРНОСТИ

Отличительной особенностью многих практических задач ЛП является их большая размерность. В частности, при решении задач оптимального планирования на уровне страны матрицы ограничений достигают размерности $m = 10^4$, $n = 10^6$. При такой размерности классические методы ЛП, такие как симплекс-метод, двойственный симплекс-метод и др., оказываются малоэффективными. Это обусловило необходимость разработки специальных методов как точных, так и приближенных, предназначенных для решения задач высокой размерности. Большинство из этих методов использует идею декомпозиции, которая заключается в расчленении большой задачи на ряд задач меньшей размерности, нахождении независимых оптимумов для каждой из них и последующей увязке этих частных решений в общее решение исходной задачи. Впервые эта идея применительно к задачам ЛП была реализована Данцигом и Вульфом, а позднее развита в работах Д. Б. Юдина и Е. Г. Гольштейна [18], М. Месаровича, Р. Крона, Л. Лэсдона [33] и др.

Метод декомпозиции Данцига-Вульфа. Метод декомпозиции Данцига-Вульфа был разработан в 1960 г. применительно к решению ЛП-задач высокой размерности со специальной структурой матрицы ограничений [33].

Этот метод оказался наиболее эффективным для решения задач, матрица ограничений которых имеет блочно-диагональный вид с небольшим числом связывающих переменных. Однако, как показали дальнейшие исследования, метод применим также и для задач ЛП с матрицей общего вида. (Соответствующий метод носит название *блочное программирование* и предложен Д. Б. Юдиным и Е. Г. Гольштейном в [18]). Отличительной особенностью метода декомпозиции является использование так называемой «координирующей задачи», которая имеет по сравнению с исходной небольшое число строк и большое число столбцов. Существенным обстоятельством является то, что для решения координирующей задачи не требуется задания всех столбцов в явном виде. Они генерируются в процессе использования симплекс-метода. Такой подход называют методом *генерации столбцов*.

Сущность его состоит в следующем. Пусть имеется ЛП-задача вида

$$\max z = \sum_{j=1}^{n} c_j x_j \tag{7.1}$$

при ограничениях

$$\sum_{j=1}^{n} \mathbf{A}_j x_j = \mathbf{b}, \tag{7.2}$$

$$x_j \geqslant 0, \qquad j = \overline{1, n},$$

где $\mathbf{A}_j$ и $\mathbf{b}$—m-мерные векторы ($m < n$).

Допустим, что известен некоторый опорный план $\mathbf{x}_B$, и соответствующая ему матрица из базисных векторов $\mathbf{A}_x$.

Допустим, что план $\mathbf{x}_B$ был определен методом обратной матрицы. Тогда одновременно был найден и вектор относительных оценок $\Lambda = \{\lambda_i\} = \mathbf{c}_x \mathbf{A}_x^{-1}$, где $\mathbf{c}_x$ — вектор коэффициентов целевой функции для текущего базиса.

Чтобы определить возможность улучшения опорного плана $\mathbf{x}_B$ для каждого небазисного вектора $\mathbf{A}_j$, вычисляется величина оценки

$$\Delta_j = \Lambda \mathbf{A}_j - c_j = \sum_{i \in I_x} \lambda_i a_{ij} - c_j. \tag{7.3}$$

Если $\min \Delta_j = \Delta_s < 0$, то начальное решение $\mathbf{x}_B$ может быть улучшено путем введения в базис переменной x_s. Однако если имеется большое число небазисных столбцов ($n \gg 10^3$), то нахождение $\min \Delta_j$ путем вычисления Δ_j для всех небазисных векторов $j = \overline{1, n}$ и последующего их сравнения практически невозможно. И что очень важно, оказывается, что это и не требуется.

Будем предполагать, что все столбцы $\mathbf{A}_j$ выбираются из некоторого выпуклого множества S, определяемого системой неравенств и равенств.

Тогда вектор-столбец, подлежащий включению в базис, может быть определен в результате решения вспомогательной задачи вида

$$\min \{\Lambda \mathbf{A}_j - c(\mathbf{A}_j)\}, \tag{7.4}$$

где $c(\mathbf{A}_j) = c_j$ — некоторая заданная функция вектора $\mathbf{A}_j$. В зависимости от структуры множества S и вида функции $c(\mathbf{A})$ выбирается наиболее эффективный метод решения указанной задачи. Такой прием и называют методом *генерации столбцов,* так как при решении задачи (7.4) фактически используется лишь небольшое число столбцов, генерируемых по мере надобности. При этом резко снижается требуемый объем памяти для хранения текущих результатов, что является очень существенным преимуществом при решении задач большой размерности.

Принцип декомпозиции

Рассмотрим ЛП-задачу, матрица ограничений которой имеет блочно-диагональную структуру вида

$$A = \begin{bmatrix} A_1 & A_2 & \dots & A_p \\ B_1 & 0 & & 0 \\ 0 & B_2 & & 0 \\ 0 & 0 & & B_p \end{bmatrix}. \qquad (7.5)$$

Строка $[A_1, A_2 \dots A_p]$ носит название связывающей, так как она как бы связывает вместе все переменные задачи.

Соответствующая ЛП-задача записывается в виде:
найти

$$\max_{\{x_i\}} \sum_{i=1}^{p} c_i x_i \qquad (7.6)$$

при условии

$$\sum_{i=1}^{p} A_i x_i = b_0; \qquad (7.7)$$

$$B_i x_i = b_i \qquad i = \overline{1, p}; \qquad (7.8)$$

$$x_i \geqslant 0.$$

Заметим, что к виду (7.5) может быть приведена матрица любой ЛП-задачи при $p = 1$ в результате соответствующего разбиения ограничений на два подмножества.

Действительно, произвольную задачу можно записать в виде:
найти

$$\max f = cx \qquad (7.9)$$

при

$$A_1 x = b_1 \qquad (m_1 \text{ ограничений}), \qquad (7.10)$$

$$A_2 x = b_2 \qquad (m_2 \text{ ограничений}), \qquad (7.11)$$

$$x \geqslant 0. \qquad (7.12)$$

Предположим, что выпуклое многогранное множество S_2, определяемое условием (7.11), является ограниченным (это условие не является ограничительным, как показано в [33]). Тогда справедлива следующая лемма, доказанная в приложении 1.

Пусть $X = \{x : Ax = b;\ x \geqslant 0\}$ — непустое замкнутое ограниченное множество и x_i $(i = \overline{1, r})$ — его крайние точки. Тогда любая точка $x \in X$ может быть представлена в виде выпуклой комбинации крайних точек множества X, т. е.

$$x = \sum_{i=1}^{r} \delta_i x_i, \quad \delta_i \geqslant 0, \quad i = \overline{1, r}, \quad \sum_{i=1}^{r} \delta_i = 1. \qquad (7.13)$$

Обобщение этой леммы на общий случай, когда множество X является неограниченным, формулируется так.

Пусть $X = \{x : Ax = b;\ x \geqslant 0\}$ — непусто. Тогда точка x принадлежит множеству X тогда и только тогда, когда она может быть представлена как сумма выпуклой комбинации крайних точек и линейной комбинации с неотрицательными коэффициентами направляющих векторов неограниченных ребер множества, т. е.

$$\mathbf{x} = \sum_i \delta_i \mathbf{x}_i, \tag{7.14}$$

где $\sum_i \delta_i \varepsilon_i = 1, \quad \delta_i \geqslant 0; \quad \varepsilon_i = \begin{cases} 1, & \text{если } \mathbf{x}_i \text{ — крайняя точка } X, \\ 0, & \text{если } \mathbf{x}_i \text{ — неограниченное ребро множества } X. \end{cases}$

В соответствии с леммой любой элемент S_2 может быть представлен в виде

$$\mathbf{x} = \sum_j \delta_j \mathbf{x}_j,$$

где $\delta_j \geqslant 0$; $\sum_j \delta_j = 1$; $\mathbf{x}_j$ — крайние точки многогранника S_2.

Исходная задача (7.9) — (7.12) может быть сформулирована таким образом. Из всех решений (7.11), (7.12) необходимо выбрать такое, которое удовлетворяет (7.10) и обращает в максимум функцию (7.9). Подставляя (7.14) в (7.9), получим новое выражение для целевой функции

$$f = \sum_j (\mathbf{c}\mathbf{x}_j)\, \delta_j, \tag{7.15}$$

а подставив (7.14) в (7.10), получим

$$\sum_j (\mathbf{A}_1 \mathbf{x}_j)\, \delta_j = \mathbf{b}_1. \tag{7.16}$$

Обозначим $\mathbf{A}_1 \mathbf{x}_j = \mathbf{P}_j$,

$$\mathbf{c}\mathbf{x}_j = z_j. \tag{7.17}$$

С учетом (7.15) — (7.17) мы приходим к следующей задаче. Найти

$$\max_{\{\delta_j\}} \sum_j z_j \delta_j \tag{7.18}$$

при ограничениях

$$\sum_j \mathbf{P}_j \delta_j = \mathbf{b}_1; \tag{7.19}$$

$$\sum_j \delta_j = 1; \tag{7.20}$$

$$\delta_j \geqslant 0. \tag{7.21}$$

Эта задача, которая эквивалентна исходной (7.9) — (7.12), называется *координирующей задачей*. Она имеет только $(m_1 + 1)$-строк ограничений, по сравнению с $(m_1 + m_2)$ строками исходной задачи, и очень большое число столбцов N, равное числу крайних точек множества S_2. Чтобы не хранить эти все столбцы в памяти ЭВМ, мы будем получать их по мере необходимости, пользуясь методом генерации столбцов. С этой целью для каждого небазисного вектора j вычислим величину Δ_j:

$$\Delta_j = \Lambda \begin{bmatrix} \mathbf{P}_j \\ 1 \end{bmatrix} - z_j. \tag{7.22}$$

Представим вектор Λ в виде $\Lambda = (\lambda_1, \lambda_0)$, где вектор λ_1 соответствует ограничениям (19), а λ_0 — единственному ограничению (7.20). Используя формулы (16, 17) для определения $\mathbf{P}_j$ и z_j, получим

$$\Delta_j = \lambda_1 \mathbf{P}_j + \lambda_0 - z_j = (\lambda_1 \mathbf{A}_1 - \mathbf{c})\, \mathbf{x}_j + \lambda_0. \tag{7.23}$$

В соответствии с обычными правилами симплекс-метода для определения переменной δ_s, вводимой в базис, необходимо найти

$$\min_j \Delta_j = \Delta_s = (\lambda_1 \mathbf{A}_1 - \mathbf{c}_j)\, \mathbf{x}_s + \lambda_0. \tag{7.24}$$

Но так как оптимальное решение ЛП-задачи (при условии, что допустимое множество S_2 ограничено) достигается в крайней точке этого множества, то выполнение операции (7.24) эквивалентно решению *подзадачи* вида

$$\min_{\mathbf{x} \geqslant 0} (\lambda_1 \mathbf{A}_1 - \mathbf{c})\, \mathbf{x} \tag{7.25}$$

при ограничениях
$$\mathbf{A}_2 x = \mathbf{b}_2, \quad \mathbf{x} \geqslant 0. \tag{7.26}$$

Найдя ее решение $\mathbf{x}_s$, проверяем условие $\Delta_s = (\lambda_1 \mathbf{A}_1 - \mathbf{c})\, \mathbf{x}_s + \lambda_0) < 0$, и если оно выполняется, то вектор $\mathbf{x}_s$ выгодно ввести в базис. Далее определяем компоненты вектора $\mathbf{P}_s$, который следует ввести в базис координирующей задачи

$$\mathbf{P}_s = \begin{bmatrix} \mathbf{A}_1 \mathbf{x}_s \\ 1 \end{bmatrix}, \tag{7.27}$$

а также соответствующий коэффициент в целевой функции

$$z_s = \mathbf{c}\mathbf{x}_s. \tag{7.28}$$

Этот подход оказывается особенно эффективным, если $p > 1$, т. е. исходная задача записывается в виде:
найти

$$\max f = \mathbf{c}_1 \mathbf{x}_1 + \mathbf{c}_2 \mathbf{x}_2 + \cdots + \mathbf{c}_p \mathbf{x}_p \tag{7.29}$$

при ограничениях

$$\begin{aligned} \mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2 + \cdots + \mathbf{A}_p \mathbf{x}_p &= \mathbf{b}_0; \\ \mathbf{B}_1 \mathbf{x}_1 \qquad\qquad\qquad &= \mathbf{b}_1; \\ \mathbf{B}_2 \mathbf{x}_2 \qquad\qquad &= \mathbf{b}_2; \end{aligned} \tag{7.30}$$

$$\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\mathbf{B}_p \mathbf{x}_p = \mathbf{b}_p; \tag{7.31}$$

$$\mathbf{x}_i \geqslant 0, \qquad i = \overline{1, p}. \tag{7.32}$$

Для такой задачи подзадача (7.25) — (7.26) будет иметь вид:
найти

$$\min \sum_{i=1}^{p} (\lambda_1 \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i \tag{7.33}$$

при ограничениях

$$\mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i, \tag{7.34}$$

$$\mathbf{x}_i \geqslant 0, \qquad i = \overline{1, p}. \tag{7.35}$$

В силу аддитивности целевой функции (7.33) и независимости ограничений (7.34) задача (7.33), (7.34) распадается на p независимых задач вида:

$$\min (\lambda_1 \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i \qquad (7.36)$$

при ограничениях

$$\mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i \qquad (7.37)$$

$$\mathbf{x}_i \geqslant 0.$$

Обозначим решение (7.36) через $\mathbf{x}_i^0$, а

$$\min_{\mathbf{x}_i} (\lambda_1 \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i = f_i^0.$$

Если $\sum\limits_{i=1} f_i^0 + \lambda_0 < 0$, то вектор $\mathbf{X}_0 = \{\mathbf{x}_i^0(\lambda_1)\}_{i=\overline{1,p}}$ может быть введен в базис координирующей задачи. Если же $\sum\limits_{i} f_i^0 + \lambda_0 \geqslant 0$, то текущее решение оптимально.

Описание алгоритма декомпозиции

Дадим формальное описание алгоритма декомпозиции Данцига-Вульфа для решения задачи (7.29) — (7.32).

Пусть уже имеется начальное допустимое базисное решение задачи (7.18) — (7.21), которому соответствует вектор разрешающих множителей

$$\Lambda = (\lambda_1, \lambda_0).$$

Каждая итерация алгоритма состоит из двух этапов.

Первый этап: 1) Используя вектор оценок λ_1 предыдущей итерации, сформируем и решим подзадачи (7.36), (7.37) и найдем оптимальные значения целевой функции f_i^0, а также соответствующие решения

$$\mathbf{x}_i^0(\lambda_1), \qquad i = \overline{1, p}.$$

2) Вычислим минимальную оценку

$$\min_j \Delta_j = \sum_{i=1}^{p} f_i^0 + \lambda_0 \qquad (7.38)$$

Если $\sum\limits_i f_i^0 + \lambda_0 \geqslant 0$, то вычисления заканчиваются и определяем искомое решение задачи (7.29) — (7.32):

$$\mathbf{X}_{\text{опт}} = \sum_j \mathbf{x}_j \delta_j, \qquad (7.39)$$

где $\{\delta_j\}$ — предыдущее базисное решение координирующей задачи, а $\mathbf{x}_j$ — крайняя точка S_2, соответствующая базисной переменной δ_j.

Если $\sum\limits_i f_i^0 + \lambda_0 < 0$, то переходим ко второму этапу.

Второй этап. Формируем столбец $\mathbf{P}$, который необходимо ввести в базис задачи (7.18) — (7.21)

$$\mathbf{P} = \sum_{i=1}^{p} \mathbf{A}_i x_i^0 (\lambda_1). \tag{7.40}$$

Вводим **P** в базис, выполняем 1 шаг симплекс-метода, после чего находим новое д. б.р.: $\{\delta_1^{(н)}, \delta_2^{(н)}, \ldots, \delta_m^{(н)}$, и $\delta_0^{(н)}\}$, а также новый вектор оценок $\Lambda^{(н)} = \{\lambda_1^{(н)}, \lambda_0^{(н)}\}$.

На этом итерация заканчивается и переходим к первому этапу очередной итерации.

Как видно из описания, алгоритм декомпозиции Данцига—Вульфа представляет собой двухуровневый алгоритм, где на первом уровне решаются подзадачи (7.36) — (7.37), а на втором уровне — координирующая задача (7.18) — (7.21). Если координирующая задача является невырожденной, то на каждой итерации значение целевой функции возрастает, и поскольку число базисов ее конечно и ни один из них не используется дважды, оптимальное решение находится за конечное число шагов.

Ограниченная координирующая задача. Как следует из приведенного описания алгоритма декомпозиции, оптимизационная задача в нем решается лишь на первом уровне, тогда как на втором фактически выполняется лишь одна итерация симплекс-метода. Однако возможна модификация этого метода, заключающаяся в решении оптимизационных задач на двух уровнях. В этом случае решается так называемая ограниченная координирующая задача, которая получается из координирующей задачи (7.18) — (7.21) в результате отбрасывания всех столбцов, за исключением базисных и претендующих на включение в базис на текущей итерации.

Такую ограниченную координирующую задачу можно представить в виде

$$\max_{\{\delta_i\}} \sum_{i=1}^{m} z_i \delta_i + z_0 \delta_0 \tag{7.41}$$

при ограничениях

$$\mathbf{P}_1\delta_1 + \mathbf{P}_2\delta_2 + \cdots + \mathbf{P}_m\delta_m + \mathbf{P}\delta_0 = \mathbf{b}_0, \tag{7.42}$$

$$\sum_{i=1}^{m} \delta_i + \delta_0 = 1, \quad \delta_i \geqslant 0, \qquad i = \overline{1, m}; \quad \delta \geqslant 0,$$

где δ_i $(i = \overline{1, m})$-переменные текущего базиса, а δ_0 — переменная, вводимая в базис (для нее $\Delta_0 < 0$). Если текущий базис является невырожденным, то из него будет выведена переменная, для которой $\Delta_l > 0$, и полученное новое решение будет оптимальным.

Заметим, что возможны случаи, когда использование ограниченной координирующей задачи по сравнению с обычной координирующей задачей дает эффект.

Варианты декомпозиции прямой задачи

Нетрудно увидеть, что существует много разных способов декомпозиции (разложения) прямой задачи, каждый из которых приводит к своей форме координирующей и ограниченной координирующей задач.

Рассмотрим, например, вариант декомпозиции для матрицы блочно-диагонального вида.

Обозначим решение системы

$$\mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i, \quad \mathbf{x}_i \geqslant 0, \qquad \mathbf{x}_i = \sum_j \delta_{ij} \mathbf{x}_i^j, \tag{7.43}$$

где $\mathbf{x}_i^j$ — крайние точки многогранника

$$S_i = \{\mathbf{x}\colon \mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i, \ \mathbf{x}_i \geqslant 0\} \qquad \delta_{ij} \geqslant 0.$$

Тогда координирующая задача будет иметь вид:
найти

$$\max \sum_i \sum_j \delta_{ij} z_{ij} \tag{7.44}$$

при ограничениях

$$\sum_i \sum_j \mathbf{P}_{ij} \delta_{ij} = \mathbf{b}_0, \tag{7.45}$$

$$\delta_{ij} \geqslant 0; \quad \sum_j \delta_{ij} = 1, \qquad i = \overline{1, p}, \tag{7.46}$$

где $z_{ij} = \mathbf{c}_i \mathbf{x}_i^j$, $\mathbf{P}_{ij} = \mathbf{A}_i \mathbf{x}_i^j$.

Координирующая задача (7.44) — (7.46) отличается от координирующей задачи (7.18) — (7.21) тем, что, во-первых, она имеет не одно, а «p» ограничений (7.46), а, во-вторых, решение для каждой подсистемы $\mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i$; $\mathbf{x}_i \geqslant 0$ самостоятельно выражается через переменные δ_{ij}, тогда как в задаче (7.18) — (7.21) эти решения рассматривались совместно.

Применим метод обратной матрицы с использованием процедуры генерации столбцов для решения (7.44) — (7.46).

Пусть $\mathbf{B}$ — базисная матрица размерности $(m_1 + p) \times (m_1 + p)$, а $\Lambda = \{\lambda, \lambda_{01}, \lambda_{02}, \ldots, \lambda_{0p}\}$—вектор относительных оценок для этого решения (λ — оценка ограничений (7.45), а λ_{0i} — оценка для i-го ограничения (7.46)).

Вычисляем теперь оценки Δ_{ij} для векторов, соответствующих переменным δ_{ij}:

$$\Delta_{ij} = (\lambda \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i^j + \lambda_{0i}). \tag{7.47}$$

Для определения $\min \Delta_{ij}$ при фиксированном i решаем подзадачу вида
найти

$$\min (\lambda \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i \tag{7.48}$$

при ограничениях

$$\mathbf{B}_i \mathbf{x}_i = \mathbf{b}_i, \quad \mathbf{x}_i \geqslant 0. \tag{7.49}$$

Если $\min\limits_i \min\limits_j (\lambda \mathbf{A}_i - \mathbf{c}_i)\, \mathbf{x}_i^j + \lambda_{0i} = \min\limits_i (f_i^0 + \lambda_{0i}) \geqslant 0$, то текущее решение оптимально, в противном случае в базис задачи (7.44) — (7.46) вводится переменная, для которой

$$\min_i (f_i^0 + \lambda_{0i}) < 0. \tag{7.50}$$

Если этот минимум достигается при $i = s$, а $\mathbf{x}_s(\lambda)$ — решение подзадачи с индексом s, то в базис координирующей задачи вводится столбец

$$\begin{bmatrix} \mathbf{A}_s\mathbf{x}_s(\lambda) \\ \mathbf{u}_s \end{bmatrix},$$

где $\mathbf{u}_s$ — p-мерный вектор, все компоненты которого равны 0, за исключением s-й компоненты, равной 1.

Пример 2.11. Для иллюстрации метода декомпозиции рассмотрим следующую задачу.

Найти

$$\max z = x_1 + x_2 + 2x_3 + x_4, \qquad (1)$$

при ограничениях

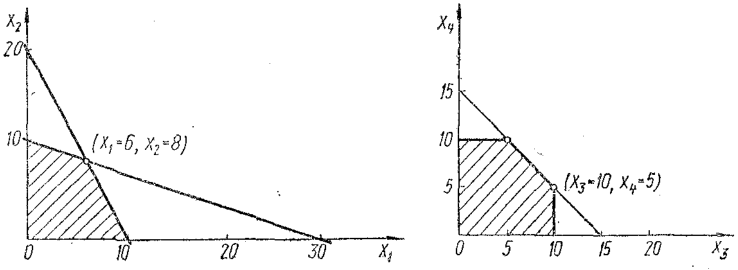
$$x_1 + 2x_2 + 2x_3 + x_4 \leqslant 40, \qquad (2)$$

$$\left.\begin{array}{l} x_1 + 3x_2 \leqslant 30, \\ 2x_1 + x_2 \leqslant 20, \end{array}\right. (S_1) \qquad (3)$$

$$\left.\begin{array}{l} x_3 \leqslant 10, \\ x_4 \leqslant 10, \\ x_3 + x_4 \leqslant 15, \end{array}\right. (S_2) \qquad (4)$$

$x_i \geqslant 0, \ \ldots, \ i = \overline{1, 4}$.

Эта задача имеет общее одно связывающее ограничение и два независимых блока. Области допустимых решений неравенств обеих подсистем представлены на рис. 2.5.



Допустимое множество S_1 Допустимое множество S_2

Рис. 2.5.

Разбиение условий задачи (2) — (4) может быть осуществлено разными способами, и каждому из них будет соответствовать своя координирующая задача с определенным числом ограничений вида $\sum_i \delta_{ij} = 1$. Воспользуемся видом ограничений и построим ограниченную координирующую задачу, которая будет иметь 2 ограничения. Пусть $\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}$ и $\mathbf{y} = \begin{Bmatrix} x_3 \\ x_4 \end{Bmatrix}$ — допустимые решения неравенств блоков 1 и 2 соответственно. Тогда справедливо соотношение

$$\mathbf{x} = \sum_i \alpha_i \mathbf{x}_i; \quad \mathbf{y} = \sum_i \beta_i \mathbf{y}_i,$$

где $\mathbf{x}_i$, $\mathbf{y}_i$ — крайние точки множеств S_1 и S_2, определяемых ограничениями (3) и (4).

Целевая функция (1) в векторном виде запишется

$$z = \mathbf{c}_1^T\mathbf{x} + \mathbf{c}_2^T\mathbf{y},$$

а связывающее ограничение

$$a_1x + a_2y + s = 40,$$

где

$$c_1^T = (1,\ 1);\quad c_2^T\,(2,\ 1);\quad a_1 = (1,\ 2);\quad a_2 = (2,\ 1),\quad s \geqslant 0.$$

Запишем теперь координирующую задачу.

Найти

$$\max z = \sum_i (c_1x_i)\,\alpha_i + \sum_i (c_2y_i)\,\beta_i, \tag{5}$$

при условии

$$\sum_i (a_1x_i)\,\alpha_i + \sum_i (a_2y_i)\,\beta_i + s = 40;$$

$$\sum_i \alpha_i = 1;$$

$$\sum_i \beta_i = 1;$$

$$\alpha_i \geqslant 0;$$

$$\beta_i \geqslant 0.$$

Выберем в качестве начального д. б. р. решение $x_1 = (0, 0)$

$$y_1 = (0,\ 0),\quad \alpha_1 = \beta_1 = 1,\quad s = 40.$$

Будем решать задачу методом обратной матрицы.

Начальная симплекс-таблица будет иметь вид

Таблица 2.25

c	B_x	A_0	A_1	A_2	A_3
0	s_1	40	1		
0	α_1	1		1	
0	β_1	1			1
	Λ	0	0	0	0

В последней строке таблицы находятся переменные $\{\lambda_i\}$. Они равны:

$$\lambda_1 = 0;\quad \lambda_{01} = 0;\quad \lambda_{02} = 0.$$

Первая итерация. Первый этап. Составим и решим подзадачи, отвечающие начальному допустимому решению:

1) Найти $\min z_1 = (\lambda a_1 - c_1)\,x = -c_1x = -x_1 - x_2$ при ограничениях $x \in S_1$.

2) Найти $\min z_2 = (\lambda a_2 - c_2)\,y = -c_2y = -2x_3 - x_4$ при ограничениях $y \in S_2$.

Оптимальные решения этих задач легко находятся графически и равны:

$$x_2 = (6,\ 8);\quad z_1^0 = -14;\quad y_2 = (10,\ 5),\quad z_2^0 = -25.$$

Определим минимальные оценки столбцов

$$\min \bar{f}_x = z_1^0 + \lambda_{01} = -14;\quad \min \bar{f}_y = z_2^0 + \lambda_{02} = -25.$$

Оптимальные решения подзадач x_2 и y_2 образуют новые столбцы координирующей задачи, которые имеют вид

$$\begin{bmatrix} a_1x_2 \\ 1 \\ 0 \\ c_1x_2 \end{bmatrix} = \begin{bmatrix} 22 \\ 1 \\ 0 \\ 14 \end{bmatrix}; \quad \begin{bmatrix} a_2y_2 \\ 0 \\ 1 \\ c_2y_2 \end{bmatrix} = \begin{bmatrix} 25 \\ 0 \\ 1 \\ 25 \end{bmatrix}.$$

Второй этап. Запишем теперь ограниченную координирующую задачу, используя ее текущий базис и эти столбцы.

Найти

$$\max (14\alpha_2 + 25\beta_2). \tag{6}$$

При условии

$$\begin{gathered} s + 22\alpha_2 + 25\beta_2 = 40, \\ \alpha_1 + \alpha_2 = 1, \\ \beta_1 + \beta_2 = 1 \end{gathered} \tag{7}$$

и условии неотрицательности.

Для решения этой задачи используем метод обратной матрицы, как наиболее подходящий для реализации процедуры генерации новых столбцов, и вводим их в исходные ограничения.

Решая ее, получим серию основных табл. 2.26—2.28 и вспомогательную табл. 2.29, где приводятся оценки Δ_j для векторов задачи.

Решение, полученное в табл. 2.28 (цикл 2), оказывается оптимальным, так как для него все оценки Δ_j в табл. 2.29 будут положительными.

В результате получено новое оптимальное решение задачи:

$$\alpha_1 = {}^7/_{22}, \quad \alpha_2 = {}^{15}/_{22} \text{ и } \beta_2 = 1;$$

$$x = \alpha_1x_1 + \alpha_2x_2 = \frac{7}{22}[0 \quad 0] + \frac{15}{22}[6 \quad 8] = \left[\frac{90}{22} \quad \frac{120}{22}\right];$$

$y = \beta_2x_2 = [10,\ 5]$. При этом значение целевой функции равно $36\frac{6}{22}$.

Вторая итерация. Первый этап. Из последней строки табл. 2.28 находим вектор относительных оценок $\Lambda = \left[\lambda\ \lambda_{01}\ \lambda_{02}\right] = \left[\frac{14}{22} \quad 0 \quad \frac{200}{22}\right]$.

Используем λ для формирования целевых функций подзадач.

Подзадача 1.

Найти

$$\min (\lambda a_1 - c_1)\, x = \min \left\{\frac{14}{22}[1 \quad 2] - [1 \quad 1]\right\} x = -\frac{8}{22}x_1 + \frac{6}{22}x_2,$$

при условии

$$x \in S_1.$$

Подзадача 2.

Найти

$$\min (\lambda a_2 - c_2)\, y = \min \left\{\frac{14}{22}\ [2 \quad 1] + [-2 \quad -1]\right\} y = -\frac{16}{22}x_3 + \frac{6}{22}x_4,$$

при условии

$$(x_3,\ x_4) \in S_2.$$

Решив эти подзадачи, получим оптимальные решения:

$$x_3 = [10 \quad 0], \quad z_1^0 = 80/22,$$

$$y_3 = [10 \quad 5], \quad z_2^0 = -200/22,$$

Таблица 2.26

(вводится β_2 и выводится β_1)

c_i	B_x	e_0	e_1	e_2	e_3	β_2	Цикл
0	s	40	1			25	0
0	α_1	1		1		0	
0	β_1	1			1	$\boxed{1}$	
	Λ	0	0	0	0	−25	

Таблица 2.27

(вводится α_2 и выводится s)

c_i	B_x	e_0	e_1	e_2	e_3	α_2	Цикл
0	s	15	1		−25	$\boxed{22}$	1
0	α_1	1		1		1	
25	β_2	1			1	0	
	Λ	25			25	−14	

Таблица 2.28

c_0	B_x	e_0	e_1	e_2	e_3	Цикл
14	α_1	15/22	1/22		−25/22	2
	α_2	7/22	1/22	1	25/22	
25	β_2	1	0		1	
	Λ	$36\frac{6}{22}$	14/22	0	200/22	

Таблица 2.29

(Оценки векторов)

Цикл	s	α_1	β_1	α_2	β_2
0	0	0	0	−14	−25
1	0	0	25	−14	0
2	14/22	0	200/22	0	0

которые отвечают следующим минимальным оценкам:

$$\min \bar{f}_x = z_1^0 + \lambda_{01} = -\frac{80}{22} + 0 = -\frac{80}{22}$$

$$\min \bar{f}_y = z_2^0 + \lambda_{02} = 0,$$

и столбцы

$$\begin{bmatrix} a_1x_3 \\ 1 \\ 0 \\ c_1x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \\ 0 \\ 10 \end{bmatrix} \quad \begin{bmatrix} a_2y_3 \\ 0 \\ 1 \\ c_2y_3 \end{bmatrix} = \begin{bmatrix} 25 \\ 0 \\ 1 \\ 25 \end{bmatrix}.$$

Так как $\min \bar{f}_x < 0$, то необходимо перейти ко второму этапу.

Второй этап. Составим новую ограниченную координирующую задачу, которая включает базисные столбцы оптимального решения предыдущей координирующей задачи и новые столбцы.

Найти

$$\max 14\alpha_2 + 25\beta_2 + 10\alpha_3 + 25\beta_3$$

при ограничениях

$$22\alpha_2 + 25\beta_2 + 10\alpha_3 + 25\beta_3 = 40;$$

$$\alpha_1 + \alpha_2 + \alpha_3;$$

$$\beta_2 + \beta_3 = 1.$$

Решаем эту задачу методом обратной матрицы, использовав в качестве начального базиса оптимальный базис предыдущей координирующей задачи (табл. 2.28). Результаты решения приводятся в табл. (2.30) — (2.32).

Итак, в результате решения координирующей задачи мы нашли новое решение $\alpha_2 = 5/12$; $\alpha_3 = 7/12$; $\beta_2 = 1$.

Ему соответствует следующее решение исходной задачи:

$x = \alpha_2 x_2 + \alpha_3 x_3 = 5/12 \cdot [6\ \ 8] - 7/12\ [10\ \ 0] = \left[\frac{25}{3};\ \ \frac{10}{3}\right]$; $y = \beta_2 y_2 = [10\ \ 5]$ и значение целевой функции $z = \frac{110}{3}$.

Таблица 2.30

(α_3 вводится, α_1 выводится из базиса)

c_i	B_x	e_0	e_1	e_2	e_3	α_3	θ
14	α_2	$\frac{15}{22}$	$\frac{1}{22}$		$-\frac{25}{22}$	$\frac{10}{22}$	$\frac{15}{10}$
	α_1	$\frac{7}{22}$	$\frac{1}{22}$	1	$\frac{25}{22}$	$\boxed{\frac{12}{22}}$	7/12
25	β_2	1	0	0	1	0	
	Λ	$36\frac{6}{11}$	$\frac{14}{22}$	0	$\frac{200}{22}$	$\frac{-80}{22}$	

Таблица 2.31

(проверка оптимальности)

c_i	B_x	e_0	e_1	e_2	e_3
14	α_2	$\frac{5}{12}$	$\frac{1}{12}$	$-\frac{10}{12}$	$-\frac{25}{12}$
10	α_3	$\frac{7}{12}$	$-\frac{1}{12}$	$\frac{22}{12}$	$\frac{25}{12}$
25	β_2	1	0	0	1
	Λ	$\frac{110}{3}$	$\frac{4}{12}$	$\frac{80}{12}$	$\frac{200}{12}$

Вспомогательная таблица 2.32

(приводятся только оценки векторов)

Цикл	α_1	α_2	α_3	β_2	β_3
0	0	0	$-\frac{80}{22}$	0	0
1	$\frac{80}{12}$	0	0	0	0

Третья итерация. Первый этап. Вектор оценок Λ имеет вид (табл. 2.31) $\Lambda = \left[\lambda \ \ \lambda_{01} \ \ \lambda_{02} = \ \right]\left[\frac{4}{12} \ \ \frac{80}{12} \ \ \frac{200}{12}\right]$.

Составим подзадачи

$$1)\ \min z_1 = \min(\mathbf{a}_1\lambda - \mathbf{c}_1)\,\mathbf{x} = \left\{\frac{4}{12}\ [1\ 2] - [1\ 1]\right\}\mathbf{x} = -\frac{2}{3}x_1 - \frac{1}{3}x_2$$

$$2)\ \min z_2 = \min(\mathbf{a}_2\lambda - \mathbf{c}_2)\,\mathbf{y} = \left\{\frac{4}{12}\ [2\ 1] - [2\ 1]\right\}\mathbf{y} = -\frac{4}{3}x_3 - \frac{2}{3}x_4,$$

решив которые, находим оптимальные решения

$$\mathbf{x}_4 = [6\ 8]; \quad \mathbf{y}_4 = [10\ 5]; \quad z_1^0 = -\frac{20}{3}; \quad z_2^0 = -\frac{50}{3}$$

и значения минимальных оценок векторов

$$\min \bar{f}_x = z_1^0 + \lambda_{01} = -\frac{20}{3} + \frac{80}{12} = 0;$$

$$\min \bar{f}_y = z_2^0 + \lambda_{02} = -\frac{50}{3} + \frac{200}{12} = 0;$$

$$\min \bar{c}_s = \lambda_1 = \frac{4}{12} > 0.$$

Так как все они неотрицательны, то полученное решение на второй итерации $\mathbf{x} = \left[\frac{25}{3} \ \ \frac{10}{3}\right]$; $\mathbf{y} = [10\ 5]$ является оптимальным.

Вопросы для самопроверки

1. Что называется *допустимыми базисными решениями*? Каким точкам допустимого множества решений они соответствуют?

2. Сформулируйте основную теорему линейного программирования. Какой практический вывод о возможном методе отыскания оптимальных решений следует из нее?

3. Как выбирается направляющий столбец при решении задачи линейного программирования (ЛП) на максимум и минимум *табличным симплекс-методом*?

4. Сформулируйте признак оптимальности д.б.р. при решении симплекс-методом задачи ЛП на максимум и минимум.

5. Каков смысл двойственных переменных и целевой функции двойственной задачи, если прямая задача максимизации?

6. При каких условиях выгоднее решать двойственную ЛП задачу, чем прямую?

7. Как найти оптимальное решение двойственной задачи, решая соответствующую прямую задачу, и наоборот?

8. Как связаны между собой оптимальные значения целевой функции прямой и двойственной задач?

9. Как связаны между собой компоненты оптимального решения прямой задачи и ограничения двойственной задачи и наоборот?

10. Дайте экономическую интерпретацию теорем 2.6 и 2.7 двойственности.

11. Как провести исследование моделей задач ЛП на чувствительность? Каков смысл оптимальных значений двойственных переменных?

12. Используя прием, описанный в книге, докажите справедливость теорем 2.6 и 2.7 (принцип рентабельности) для пары двойственных задач при произвольных ограничениях.

13. Как определить количество искусственных переменных, которые следует вводить в ограничения задачи ЛП? Поясните смысл коэффициентов $+M$ ($-M$), с которыми вводятся искусственные переменные в целевую функцию.

14. Как связаны между собой разрешающий вектор **A** для прямой задачи и оптимальные планы (решения) двойственной.

15. Сформулируйте признак оптимальности текущего плана при использовании разрешающих множителей.

16. Дайте определение *псевдоплана*. Каким свойством обладают оценки векторов любого псевдоплана? Чем отличается псевдоплан от произвольного допустимого базисного решения?

17. Каким условиям должен удовлетворять произвольный сопряженный базис и как его найти?

18. Сформулируйте признак неразрешимости ЛП-задачи при ее решении двойственным симплекс-методом и дайте его обоснование.

20. Укажите сходства и различия обычного симплекс-метода и двойственного симплекс-метода.

21. Поясните основную идею метода декомпозиции Данцига—Вульфа. На каком свойстве множества решений ЛП-задач базируется этот метод?

22. Поясните смысл координирующей задачи в методе декомпозиции.

23. Почему метод декомпозиции использует метод обратной матрицы (модифицированный симплекс-метод)?

24. Сформулируйте признак оптимальности при решении ЛП-задач методом декомпозиции и докажите его справедливость.

Глава 3. ТРАНСПОРТНАЯ ЗАДАЧА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

§ 1. ПОСТАНОВКА И ОСНОВНЫЕ СВОЙСТВА ТРАНСПОРТНОЙ ЗАДАЧИ

Транспортная задача (Т-задача) является одной из самых распространенных специальных задач ЛП. Частные постановки задачи рассмотрены рядом специалистов по транспорту, например, А. Н. Толстым [58].

Первая строгая постановка Т-задачи принадлежит Хичкоку, и поэтому в зарубежной литературе иногда ее называют проблемой Хичкока.

Первый точный метод решения Т-задачи разработан советскими учеными Л. В. Канторовичем и М. К. Гавуриным.

Постановка Т-задачи. Пусть в пунктах $A_1, A_2, \ldots, A_m$ производят некоторый однородный продукт, причем объем производства в пункте A_i составляет a_i единиц $(i = 1, 2, \ldots, m)$. Допустим, что данный продукт потребляют в пунктах $B_1, \ldots, B_n$, а объем потребления в пункте B_j составляет b_j единиц $(j = 1, 2, \ldots, n)$.

Предположим, что из каждого пункта производства возможна транспортировка продукта в любой пункт потребления. Транспортные издержки по перевозке из пункта A_i в пункт B_j единицы продукции равны c_{ij} $(i = 1, 2, .., m; j = 1, 2, \ldots, n)$.

Задача состоит в определении такого плана перевозок, при котором запросы всех потребителей полностью удовлетворены, весь продукт из пунктов производства вывезен и суммарные транспортные издержки минимальны.

Условия Т-задачи удобно представить в виде табл. 3.1.

Пусть x_{ij} — количество продукта, перевозимого из пункта A_i в пункт B_j. Требуется определить множество переменных $x_{ij} \geqslant 0$ $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$, удовлетворяющих условиям

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1, 2, \ \ldots \ m), \tag{1.1}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1, 2, \ \ldots, \ n), \tag{1.2}$$

и таких, что целевая функция

$$L(x_{ij}) = \sum_{i-1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1.3}$$

достигает минимума.

Условие (1.1) гарантирует полный вывоз продукта из всех пунктов производства, а условие (1.2) означает полное удовлетворение спроса во всех пунктах потребления.

Т-задача представляет собой задачу ЛП с $(m \times n)$ числом переменных x_{ij} и $(m + n)$ числом ограничений-равенств.

Переменные x_{ij} нумеруют с помощью двух индексов и потому набор $\{x_{ij}\}$, удовлетворяющий условиям (1.1) и (1.2), записывают в виде матрицы:

Таблица 3.1

Пункты потребления Пункты производства	B_1	B_2	B_3	. . .	B_n	B_j a_i
A_1	c_{11}	c_{12}	c_{13}	. . .	c_{1n}	a_1
A_2	c_{21}	c_{22}	c_{23}	. . .	c_{2n}	a_2
. . .	. . .	. . .	. . .	. . .	. . .	. . .
A_m	c_{m1}	c_{m2}	c_{m3}	. . .	c_{mn}	a_m
A_i b_j	b_1	b_2	b_3	. . .	b_n	Объем производства Объем потребления

$$\mathbf{X} = \|x_{ij}\|_{m,n} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix}. \qquad (1.4)$$

Матрицу $\mathbf{X}$ называют планом перевозок Т-задачи, а переменные x_{ij} — перевозками. План $\mathbf{X}_{\text{опт}}$, при котором целевая функция минимальна, называется оптимальным планом. Матрица $\mathbf{C} = \| c_{ij} \|$ называется матрицей транспортных издержек.

Вектор $\mathbf{P}_{ij}$, компоненты которого состоят из коэффициентов при переменных x_{ij} в ограничениях (1.1) и (1.2), называют вектором коммуникаций:

$$\mathbf{P}_{ij}^T = [\underbrace{\underbrace{\underbrace{0,\ 0,\ \ldots,\ 0,\ 1}_{i},\ 0,\ \ldots,\ 0,\ 1}_{m+j},\ 0,\ \ldots,\ 0}_{m+n}].$$

Вводят также вектор производства-потребления $\mathbf{P}_0$, где $\mathbf{P}_0^T = (a_1, a_2, \ldots, a_m, b_1, \ldots, b_n)$.

Тогда ограничения (1.1) и (1.2) можно записать в векторной форме так:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}\mathbf{P}_{ij} = \mathbf{P}_0. \tag{1.5}$$

Графический способ задания условий Т-задачи показан на рис. 3.1.

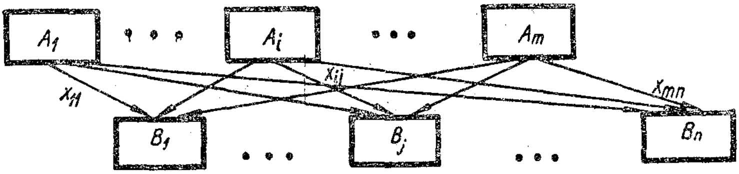


Рис. 3.1.

Отрезок $\overline{\mathrm{A}_i\mathrm{B}_j}$ называют коммуникацией. На всех коммуникациях ставят величины перевозок x_{ij}.

Свойства транспортной задачи

1. *Для разрешимости Т-задачи необходимо и достаточно, чтобы выполнялось условие баланса:*

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j, \tag{1.6}$$

другими словами, объем производства должен быть равен объему потребления.

Доказательство. Пусть переменные x_{ij} ($i = 1, 2, \ldots, m$; $j = 1, \ldots, n$) удовлетворяют условиям (1.1) и (1.2). Суммируя (1.1) по i, а (1.2) по j, получим

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}; \quad \sum_{j=1}^{n} b_j = \sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij}.$$

Отсюда $\sum\limits_{i=1}^{m} a_i = \sum\limits_{j=1}^{n} b_j$, что и доказывает необходимость условия баланса Т-задачи.

Пусть справедливо условие (1.6). Обозначим $\bar{x}_{ij} = \frac{a_i b_j}{d}$, где $d = \sum\limits_{i=1}^{m} a_i = \sum\limits_{j=1}^{n} b_j$. Нетрудно показать, что $\bar{x}_{ij}$ составляет план задачи. Действительно

$$\sum_{j=1}^{n} \bar{x}_{ij} = \frac{a_i}{d} \sum_{j=1}^{n} b_j = a_i \qquad (i = 1, 2, \ldots, m);$$

$$\sum_{i=1}^{m} \bar{x}_{ij} = \frac{b_j}{d} \sum_{i=1}^{m} a_i = b_j \qquad (j = 1, 2, \ldots, n).$$

Таким образом доказана достаточность условия баланса для Т-задачи.

2. Ранг системы ограничений (1.1), (1.2) равен $r = m + n - 1$. Доказательство. Так как количество уравнений (1.1), (1.2) равно $m + n$, то ранг этой системы $r \leqslant m + n$.

Теперь пусть набор $\{x_{ij}\}_{j=n}^{i=m}$ удовлетворяет всем уравнениям, кроме первого. Покажем, что он удовлетворяет также и первому уравнению.

Очевидно,

$$\sum_{j=1}^{n} x'_{1j} = \sum_{i=1}^{m}\sum_{j=1}^{n} x'_{ij} - \sum_{i=2}^{m}\sum_{j=1}^{n} x'_{ij}.$$

Так как

$$\sum_{j=1}^{n} x'_{ij} = a_i \quad (i = 2, \ldots, m) \text{ и } \sum_{i=1}^{m} x'_{ij} = b_j \quad (j = 1, 2, \ldots, n),$$

то

$$\sum_{i=1}^{m}\sum_{i=1}^{n} x'_{ij} = \sum_{j=1}^{n} b_j, \quad \sum_{i=2}^{m}\sum_{j=1}^{n} x'_{ij} = \sum_{i=2}^{m} a_i.$$

Откуда

$$\sum_{j=1}^{n} x'_{ij} = \sum_{j=1}^{n} b_j - \sum_{i=2}^{m} a_i.$$

Учитывая условие баланса (1.6), получим

$$\sum_{j=1}^{n} x'_{1j} = \sum_{i=1}^{m} a_i - \sum_{i=2}^{m} a_i = a_1,$$

т. е. первое уравнение системы (1.1) тоже удовлетворяется. Таким образом, ранг системы уравнений $r \leqslant m + n - 1$.

Теперь докажем, что ранг системы уравнений (1.1), (1.2) равен точно $m + n - 1$. Для этого составим матрицу из первых $(m + n - 1)$ компонент векторов $\mathbf{P}_{1n}$, $\mathbf{P}_{2n}$, ..., $\mathbf{P}_{mn}$, $\mathbf{P}_{11}$, ..., $\mathbf{P}_{1,n-1}$:

$$\begin{array}{c|cccccccc|}
 & \mathbf{P}_{1n} & \mathbf{P}_{2n} & \ldots & \mathbf{P}_{mn} & \mathbf{P}_{11} & \mathbf{P}_{12} & \ldots & \mathbf{P}_{1,n-1} \\
 & 1 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 \\
\hline
\uparrow & 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
m & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\downarrow & 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 \\
\hline
\uparrow & 0 & 0 & & 0 & 1 & 0 & & 0 \\
n-1 & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
\downarrow & 0 & 0 & & 0 & 0 & 0 & & 1 \\
\hline
 & \multicolumn{4}{c}{\underbrace{\qquad\qquad\qquad\qquad}_{m}} & \multicolumn{4}{c}{\underbrace{\qquad\qquad\qquad\qquad}_{n-1}}
\end{array}$$

Очевидно, что эта матрица не вырождена. Поэтому векторы $\{\mathbf{P}_{1n}, \mathbf{P}_{2n}, \ldots, \mathbf{P}_{mn}, \mathbf{P}_{11}, \mathbf{P}_{12}, \ldots, \mathbf{P}_{1,n-1}\}$ образуют базис. Так как базис системы состоит из $(m+n-1)$ векторов, то ранг системы $r=m+$ $+n-1$.

Двойственная транспортная задача ($\tilde{T}$-задача). Для Т-задачи, как и для любой задачи ЛП, существует двойственная к ней $\tilde{T}$-задача.

Переменные $\tilde{T}$-задачи обозначим $v_1, v_2, \ldots, v_n$ и $(-u_1), (-u_2), \ldots$ $\ldots, (-u_m)$.

Теорема 3.1. $\tilde{T}$-задача всегда имеет решение и если $X_{\text{опт}} =$ $= \| x_{ij} \|$ и $W_{\text{опт}} = \{v_{1\text{опт}}, v_{2\text{опт}}, \ldots, v_{n\text{опт}}, -u_{1\text{опт}}, -u_{2\text{опт}}, \ldots$ $\ldots, -u_{m\text{опт}}\}$ — оптимальные решения Т- и $\tilde{T}$-задач соответственно, то

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij\text{опт}} = \sum_{j=1}^{n} b_j v_{j\text{опт}} - \sum_{i=1}^{m} a_i u_{i\text{опт}}. \tag{1.7}$$

Смысл этой теоремы состоит в том, что суммарные транспортные расходы при оптимальном плане перевозок равны приращению суммарной оценки стоимости продукции при полном удовлетворении спроса.

Переменные v_j и u_i называют потенциалами пунктов A_i и B_j для Т-задачи.

Таким образом, теорема 3.1 утверждает, что при оптимальных решениях значения целевой функции прямой и двойственной Т-задач равны между собой.

Справедливость теоремы следует из основной теоремы двойственности ЛП (теорема 2.5).

Сформулируем теперь необходимые и достаточные условия оптимальности плана Т-задачи.

Теорема 3.2. *Для оптимальности плана* X_0 *Т-задачи необходимо и достаточно существование* $v_1, v_2, \ldots, v_n$ и $(-u_1), (-u_2), \ldots$ $\ldots, \ldots, (-u_m)$ таких, что

$$v_j - u_i \leqslant c_{ij} \qquad (i=1, 2, \ldots, m;\ j=1, 2, \ldots, n) \tag{1.8}$$

при этом, если

$$x_{ij}^0 > 0, \text{ то } v_j - u_i = c_{ij}.$$

Эта теорема дает достаточные условия оптимальности плана X.

Справедливость этой теоремы вытекает из общих идей теории двойственности линейного программирования (в частности, теорем 2.5 и 2.7). Дадим экономическую интерпретацию условий теоремы 3.2. Разность между потенциалами пунктов B_j и A_i, т. е. величину $v_j - u_i$, можно рассматривать как приращение ценности единицы продукции при перевозке из пункта A_i в пункт B_j.

Поэтому если $v_j - u_i < c_{ij}$, то перевозка по коммуникации $\overrightarrow{A_iB_j}$ нерентабельна и $x_{ij}^0 = 0$.

Если $v_j - u_i = c_{ij}$, то такая перевозка рентабельна и $x_{ij}^{(0)} > 0$ (сравните с теоремой 2.7, § 4, гл. 2).

Транспортная задача с ограниченными пропускными способностями (T_d-задача). Важной в практическом отношении является Т-задача, в которой существуют ограничения на пропускные способности коммуникаций.

Пусть d_{ij} — пропускная способность коммуникации $\overrightarrow{A_iB_j}$. Тогда

$$x_{ij} \leqslant d_{ij}. \tag{1.9}$$

T_d-задача состоит в минимизации целевой функции (1.3) при условиях (1.1), (1.2) и (1.9).

Даже в случае разрешимости Т-задачи T_d-задача может оказаться неразрешимой, поскольку величины пропускных способностей будут недостаточны для полного вывоза продукта из пункта A_i ($i = 1, 2, \ldots, m$) и его полного ввоза в пункт B_j ($j = 1, 2, \ldots, n$). Поэтому для T_d-задачи вводят еще два условия:

$$\sum_{j=1}^{n} d_{ij} \geqslant a_i \qquad (i = 1, 2, \ldots, m) \tag{1.10}$$

и

$$\sum_{i=1}^{m} d_{ij} \geqslant b_j \qquad (j = 1, 2, \ldots, n). \tag{1.11}$$

Но и при добавочных условиях (1.10), (1.11) T_d-задача не всегда разрешима. Для установления совместимости всех условий строят любой план T_d-задачи. Если план построить удается, то система уравнений (1.1), (1.2), (1.9), (1.10) и (1.11) совместна. В противном случае Т-задача неразрешима.

Теорема 3.3. *Для оптимальности плана* X_0 *T_d-задачи необходимо и достаточно существование таких* $v_1, v_2, \ldots, v_n$ *и* $(-u_1), (-u_2), \ldots \ldots (-u_m)$, *при которых*

$$v_j - u_i \leqslant c_{ij}, \text{ если } x_{ij}^{(0)} = 0; \tag{1.12}$$

$$v_j - u_i = c_{ij}, \text{ если } 0 < x_{ij}^{(0)} < d_{ij}; \tag{1.13}$$

$$v_j - u_i \geqslant c_{ij}, \text{ если } x_{ij}^{(0)} = d_{ij}. \tag{1.14}$$

Смысл условий оптимальности (1.12) (1.13), (1.14) состоит в следующем. Если приращение стоимости продукта $v_j - u_i$ при перевозке по коммуникации $\overrightarrow{A_iB_j}$ меньше транспортных расходов c_{ij} (1.12), то такая перевозка убыточна, а потому $x_{ij}^{(0)} = 0$.

Если же приращение стоимости продукта $v_j - u_i$ больше транспортных расходов c_{ij} (1.14), то эта перевозка прибыльна и ее величина в оптимальном плане должна быть максимальной, т. е.

$$x_{ij}^{(0)} = d_{ij}.$$

Таким образом, теорема 3.3. по существу выражает *принцип рентабельности для T_d-задачи.*

Открытые транспортные модели. Существует ряд практических задач, в которых условие баланса $\sum\limits_{i=1}^{m} a_i = \sum\limits_{j=1}^{n} b_j$ не выполняется. Такие модели называются открытыми Т-моделями. Возможны два случая: 1) $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$; 2) $\sum_{i} a_i > \sum_{j} b_j$.

В первом случае полное удовлетворение спроса невозможно. Такую задачу можно привести к обычной Т-задаче следующим образом.

Обозначим через r_j величину штрафа из-за неудовлетворения запросов на единицу продукта в пункте B_j.

Тогда требуется минимизировать суммарные затраты

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} r_j y_j \tag{1.15}$$

при условиях

$$\sum_{j=1}^{n} x_{ij} = a_i \text{ для всех } i; \quad \sum_{i=1}^{m} x_{ij} \leqslant b_j, \qquad j = 1, \ldots, n,$$

где $y_j = b_j - \sum_{i=1}^{m} x_{ij}$ — неудовлетворенный спрос.

Задачу (1.15) приводят к обычной Т-задаче введением фиктивного пункта производства A_{m+1} с объемом производства $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$ и транспортными издержками $c_{m+1,j} = r_j$, $j = 1, \ldots, n$.

В таком случае Т-задача будет иметь вид:
найти

$$\min_{\{x_{ij}\}} \sum_{i=1}^{m+1} \sum_{j=1}^{n} c_{ij} x_{ij},$$

$$\sum_{j=1}^{n} x_{ij} = a_i; \quad \sum_{i=1}^{m+1} x_{ij} = b_j.$$

В найденном решении $X_{\text{опт}}$ полагаем все перевозки из фиктивного пункта A_{m+1} равными нулю, т. е. $x_{m+1,j}^{\text{опт}} = 0$, $j = \overline{1, n}$.

Примеры задач, приводимых к транспортной модели

Задача наилучшего распределения боевых средств (задачи целераспределения). Пусть имеется система обслуживания, состоящая из n типов различных приборов. Обозначим N_j $(j = 1, \ldots, n)$ — количество приборов j-го типа, тогда общее количество приборов $\sum_{j=1}^{n} N_j = N$.

В систему обслуживания в определенные моменты времени прибывают заявки разных типов. Обозначим $d_i^{(t)}$ $(i = 1, \ldots, m)$ — число заявок i-го типа, прибывших в систему в момент времени t.

Общее число заявок равно

$$D = \sum_{i=1}^{m} d_i.$$

Различные приборы имеют разную эффективность (производительность) при обслуживании соответствующих типов заявок. Обозначим матрицу эффективностей $M = \| \mu_{ij} \| / \begin{matrix} i = \overline{1, m}. \\ j = \overline{1, n} \end{matrix}$

Если предположить, что каждый прибор в любой момент времени может обслуживать не более одной заявки, то в данной задаче требуется распределить свободные приборы по заявкам так, чтобы интегральная эффективность системы была наибольшей.

Обозначим x_{ij} $(0 \leqslant x_{ij} \leqslant N_j)$ — число приборов j-го типа, отведенных для обслуживания заявок i-го типа.

Тогда сформулированную выше задачу можно записать таким образом:

найти

$$\max_{\{x_{ij}\}} \sum_{i=1}^{m} \sum_{j=1}^{n} \mu_{ij} x_{ij}, \tag{1.16}$$

при ограничениях $\sum\limits_{i=1}^{m} x_{ij} \leqslant N_j$; $\sum\limits_{j=1}^{n} x_{ij} \leqslant d_i^{(t)}$; $i = 1, \ldots, m$; $j = 1, \ldots, n$.

Полученная модель является моделью открытой Т-задачи.

Задача перевозки неоднородного продукта на разнотипном транспорте. Выше была рассмотрена такая постановка Т-задачи, когда требовалось перевозить однородный продукт на однородном транспорте. В данном случае при отказе от этих упрощающих предположений задача значительно усложняется, однако иногда ее удается привести к классической Т-задаче искусственным путем.

Пусть для обеспечения перевозок используют S автохозяйств, в каждом из которых имеется r типов машин. Допустим, что разнотипные машины, обладая различными эксплуатационными характеристиками и разной скоростью, могут доставить любой из m видов грузов каждому из n потребителей.

Вводим следующие величины, которые предполагаем известными:

a_{lk} — количество машин k-го типа в автохозяйстве l $(l = 1, \ldots, s$; $k = 1, \ldots, r)$;

c_{ij} — количество i-го груза, которое подлежит перевозке j-му потребителю $(i = 1, \ldots, m$; $j = 1, \ldots, n)$;

d_{ij} — количество i-го груза, которое перевозят j-му потребителю (определяют по известной грузоподъемности машин);

t_{ijlk} — время занятости одной машины k-го типа автохозяйства l по перевозке i-го груза j-му потребителю.

Определить, сколько машин каждого типа из каждого автохозяйства следует направить для полного удовлетворения спроса при минимальных общих затратах автомобиле-часов.

Количество машин k-го типа из автохозяйства l, занятых перевозкой i-го груза j-му потребителю, обозначим x_{ijlk}.

Найти

$$\min_{\{x_{ijlk}\}} \sum_{i}^{m} \sum_{j}^{n} \sum_{l=1}^{S} \sum_{k=1}^{r} t_{ijlk} x_{ijlk} \tag{1.17}$$

при условиях

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ijlk} \leqslant a_{lk}; \quad l = 1, \ldots, S; \quad k = 1, \ldots, r$$

и

$$\sum_{l} \sum_{k} d_{ij} x_{ijlk} = c_{ij}; \quad 0 \leqslant x_{ijlk}.$$

Заменим пары индексов (i, j) и (l, k) соответственно индексами λ и μ по следующим формулам:

$$\lambda = i + m(j-1); \quad \mu = l + S(k-1).$$

Если индекс i принимает все значения от 1 до m, а индекс j — от 1 до n, то индекс λ принимает все целые значения от 1 до mn.

Аналогично при изменении значений индекса l от 1 до S, а индекса k — от 1 до r, индекс μ принимает все значения от 1 до Sr.

Введем для переменных следующие обозначения: $x_{ijlk} = z_{\lambda\mu}$, $t_{ijlk} = \tau_{\lambda\mu}$, $a_{lk} = b_{\mu}$, а для отношения $\frac{c_{ij}}{d_{ij}} = g_{\lambda}$.

В новых обозначениях задача (1.17) имеет такой вид: найти

$$\min_{\{z_{\lambda\mu}\}} \sum_{\lambda=1}^{mn} \sum_{\mu=1}^{Sr} \tau_{\lambda\mu} z_{\lambda\mu}, \tag{1.18}$$

при условиях

$$\sum_{\lambda=1}^{mn} z_{\lambda\mu} \leqslant b_{\mu}; \quad \mu = 1, \ldots, Sr,$$

$$\sum_{\mu=1}^{Sr} z_{\lambda\mu} = g_{\lambda}; \quad \lambda = 1, \ldots, mn; \quad z_{\lambda\mu} \geqslant 0.$$

Итак, это обычная Т-задача с размерностью $mn \times Sr$.

По компонентам оптимального плана $z_{\lambda\mu\,\text{опт}}$ вычисляют составляющие решения исходной задачи $x_{ijlk\,\text{опт}}$. При этом индексы i и j вычисляют следующим образом:

$$i = \begin{cases} m, \text{ если } \lambda \text{ кратно } m; \\ \text{остатку от деления } \lambda \text{ на } m, \text{ если } \lambda \text{ не} \\ \text{кратно } m; \end{cases}$$

$$j = \begin{cases} \dfrac{\lambda}{m}, \text{ если } \lambda \text{ кратно } m, \\ \text{целой части} \left(\dfrac{\lambda}{m} + 1\right), \text{ если } \lambda \text{ не кратно } m. \end{cases}$$

Аналогично вычисляются индексы l и k по значениям μ.

Опорные планы транспортной задачи

Опорным планом Т-задачи называют любое ее допустимое базисное решение. Понятие опорного плана имеет наглядную геометрическую интерпретацию.

Последовательность коммуникаций

$$\overrightarrow{A_{i_1}B_{j_1}},\ \overrightarrow{A_{i_2}B_{j_1}},\ \ldots,\ \overrightarrow{A_{i_s}B_{j_s}} \tag{1.19}$$

называют *маршрутом,* соединяющим пункты A_{i_1} и B_{j_s} (рис. 3.2).

Используя маршрут, составленный из коммуникаций, можно осуществить перевозку из пункта A_{i_1} в пункт B_{j_s}, проходя через пункты B_{j_1}, A_{i_2}, ..., A_{i_s}. В процессе движения коммуникации, стоящие на четных местах в (1.19), будут пройдены в противоположном направлении.

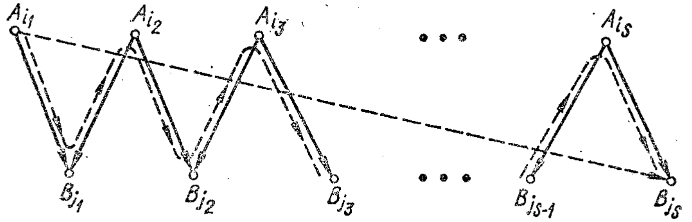


Рис. 3.2.

Маршрут (1.19), к которому добавлена коммуникация $\overrightarrow{A_{i_1}B_{j_s}}$, называется *замкнутым маршрутом,* или *циклом.*

Способ проверки произвольного плана T-задачи на опорность основан на следующих теоремах (прямой и обратной).

Теорема 3.4. *Система, составленная из векторов* $\mathbf{P}_{ij}$ *T-задачи, является линейно независимой тогда и только тогда, когда из коммуникаций, соответствующих этим векторам, нельзя составить замкнутый маршрут.*

Доказательство. Докажем сначала необходимость. Пусть векторы $\{\mathbf{P}_{i_1j_1}, \mathbf{P}_{i_2j_1}, \ldots, \mathbf{P}_{i_sj_s}\}$ линейно независимы. Если бы существовал замкнутый маршрут из коммуникаций $\overrightarrow{A_{i_1}B_{j_1}}$, $\overrightarrow{A_{i_2}B_{j_1}}$, ..., $\overrightarrow{A_{i_s}B_{j_s}}$ и $\overrightarrow{A_{i_1}B_{j_s}}$, то, очевидно, начиная движение из пункта A_{i_1} и пройдя все пункты $B_{j_1} \to A_{i_2} \to B_{j_2} \ldots \to B_{i_s}$ по последней коммуникации $A_{i_1}B_{j_s}$, мы вернемся в начальный пункт A_{i_1}. Тогда справедливо равенство

$$\mathbf{P}_{i_1j_1} - \mathbf{P}_{i_2j_1} + \mathbf{P}_{i_2j_2} + \cdots + \mathbf{P}_{i_sj_s} - \mathbf{P}_{i_1j_s} = 0, \tag{1.20}$$

которое указывает на линейную зависимость векторов $\{\mathbf{P}_{i_1j_1}, \mathbf{P}_{i_2j_1}, \ldots, \mathbf{P}_{i_1j_s}\}$.

Полученное противоречие доказывает необходимость условий теоремы 3.4. Перейдем к доказательству достаточности.

Допустим, что из коммуникаций, отвечающих векторам $\mathbf{P}_{ij}$ системы R, нельзя составить замкнутый маршрут. Докажем, что R — линейно-независимая система.

Если предположить противное, т. е. линейную зависимость векторов системы R, то существуют такие числа a_{ij}, $(i, j) \in E$, среди которых не все нули, для которых выполняется условие:

$$\sum_{(i,j)\in E} a_{ij}\mathbf{P}_{ij} = 0. \tag{1.21}$$

Пусть, например, $a_{i_1j_1} \neq 0$. Перенесем тогда соответствующий вектор $\mathbf{P}_{i_1j_1}$ вправо и получим

$$\sum_{(i,j)\in E_1} a_{ij}\mathbf{P}_{ij} = -a_{i_1j_1}\mathbf{P}_{i_1j_1}, \tag{1.22}$$

где E_1 образуется вычеркиванием в E пары индексов (i_1, j_1). Компонента с номером i_1 в правой части векторного равенства не равна 0. Следовательно, это же относится и к левой части этого равенства, т. е. среди векторов $\mathbf{P}_{ij}$, $(i, j) \in E_1$, найдется хотя бы один вектор вида $\mathbf{P}_{i_1j_2}$ с коэффициентом $a_{i_1j_2} \neq 0$. Перенеся его в правую часть равенства (1.22), получим:

$$\sum_{(i,j)\in E_2} a_{ij}\mathbf{P}_{ij} = -a_{i_1j_1}\mathbf{P}_{i_1j_1} - a_{i_1j_2}\mathbf{P}_{i_1j_2}, \tag{1.23}$$

где $E_2 = E_1 - (i_1, j_2)$. Но поскольку $j_2 \neq j_1$, компонента с номером $m + j_2$ правой части (1.23) отлична от нуля. Поэтому среди векторов левой части (1.23) найдется хотя бы один вектор вида $\mathbf{P}_{i_2j_2}$, для которого $a_{i_2j_2} \neq 0$. Перенося его в правую часть (1.23), получим:

$$\sum_{(i,j)\in E_3} a_{ij}\mathbf{P}_{ij} = -a_{i_1j_1}\mathbf{P}_{i_1j_1} - a_{i_1j_2}\mathbf{P}_{i_1j_2} - a_{i_2j_2}\mathbf{P}_{i_2j_2}, \tag{1.24}$$

где $E_3 = E_2 - (i_2, j_2)$.

Этот процесс переноса векторов в правую часть можно продолжить аналогичным образом и дальше. Допустим, что уже проведено $2k - 1$ шагов. Тогда имеет место соотношение

$$\sum_{(i,j)\in E_{2k-1}} a_{ij}\mathbf{P}_{ij} = -\sum_{\mu=1}^{k} a_{i_\mu j_\mu}\mathbf{P}_{i_\mu j_\mu} - \sum_{\mu=1}^{k-1} a_{i_\mu j_{\mu+1}}, \tag{1.25}$$

где

$$E_{2k-1} = E_{2k-2} - (i_k, j_k).$$

Возможны 2 случая:

1. $i_k = i_s$, при некотором $1 \leqslant s \leqslant k - 1$,
2. $i_k \neq i_s$.

В первом случае процесс переноса заканчивается, причем из векторов в правой части (1.25) можно образовать замкнутый маршрут. Таким маршрутом является

$$\overrightarrow{A_{i_s}B}_{j_{s+1}}, \ \overrightarrow{A_{i_{s+1}}B}_{j_{s+1}}, \ \ldots, \ \overrightarrow{A_{i_{k-1}}B}_{j_k}, \ \overrightarrow{A_{i_s}B}_{j_k}.$$

Во втором случае процесс переноса продолжается, и поскольку $i_k \neq i_s$ $(1 \leqslant s \leqslant k - 1)$, среди векторов $\mathbf{P}_{ij}$, где $(i, j) \in E_{2k-1}$, обязательно найдется вектор $\mathbf{P}_{i_kj_{k+1}}$ с коэффициентом $a_{i_kj_{k+1}} \neq 0$. Описанный процесс не может длиться бесконечно, так как все вектора, переносимые вправо, различны. Поэтому через конечное число шагов мы обязательно столкнемся со случаем 1, который, как показано, ведет к образованию замкнутого маршрута. Итак, допустив, что система векторов $\mathbf{P}_{ij}$ линейно зависима, мы пришли к противоречию с условием теоремы, согласно которому замкнутый маршрут из коммуникаций $\mathbf{P}_{ij}$ системы R составить нельзя. Остается принять, что система R состоит из линейно-независимых векторов. Достаточность условий теоремы доказана.

Согласно определению план $\mathrm{X} = \| x_{ij} \|$ называется опорным, если система векторов P_{ij}, отвечающих ненулевым перевозкам x_{ij}, линейно независима.

Назовем коммуникацию $\overrightarrow{\mathrm{A}_i\mathrm{B}_j}$ Т-задачи *основной коммуникацией плана* X, *если* $x_{ij} > 0$.

Тогда, используя теорему 3.4, можно сформулировать следующий признак проверки произвольного плана на опорность.

План $\mathrm{X} = \| x_{ij} \|$ *Т-задачи является опорным, если из его основных коммуникаций нельзя составить замкнутый маршрут.*

Теорема 3.5. *Вектор* P_{kl} *является линейной комбинацией векторов системы* R *тогда и только тогда, когда из векторов этой системы можно составить маршрут, соединяющий пункты* A_k *и* B_l. *Если этот маршрут имеет вид*

$$\mathrm{A}_k\mathrm{B}_{j_1},\ \mathrm{A}_{i_1}\mathrm{B}_{j_1},\ \ldots,\ \mathrm{A}_{i_s}\mathrm{B}_{j_s},\ \mathrm{A}_{i_s}\mathrm{B}_l,$$

то

$$\mathrm{P}_{kl} = \mathrm{P}_{kj_1} - \mathrm{P}_{i_1j_1} + \mathrm{P}_{i_1j_2} - \cdots - \mathrm{P}_{i_sj_s} + \mathrm{P}_{i_sl}. \qquad (1.26)$$

Доказательство этой теоремы основано на теореме 3.4. Пусть P_{kl} выражен в виде линейной комбинации векторов системы R. Добавив к системе R вектор P_{kl}, получим систему линейно зависимых векторов. Тогда в силу теоремы 3.4 появляется замкнутый маршрут γ. Этот замкнутый маршрут должен содержать коммуникацию $\mathrm{A}_k\mathrm{B}_l$ и, следовательно, все остальные коммуникации должны соединить A_k и B_l.

Тогда

$$\mathrm{P}_{kj_1} - \mathrm{P}_{i_1j_1} + \mathrm{P}_{i_1j_2} - \cdots - \mathrm{P}_{i_sj_s} + \mathrm{P}_{i_sl} - \mathrm{P}_{kl} = 0.$$

Перенеся P_{kl} в правую часть, получим выражение (1.26), что и требовалось доказать.

Рассмотрим произвольную матрицу $\mathrm{X}^{m \times n}$. Между позициями матрицы X и векторами P_{ij} можно установить следующее соответствие. Вектор P_{ij} соответствует элементу матрицы X на пересечении i-й строки и j-го столбца. Тогда можно задать произвольную систему из векторов $\{\mathrm{P}_{ij}\}$, выделив единицами соответствующие элементы матрицы X. Рассмотрим матрицу на рис. 3.3.

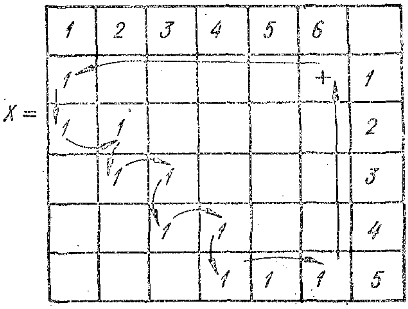


Рис. 3.3.

Здесь единицами отмечена система векторов R:

$$\{\mathrm{P}_{11}, \mathrm{P}_{21}, \mathrm{P}_{22}, \mathrm{P}_{32}, \mathrm{P}_{33}, \mathrm{P}_{43}, \mathrm{P}_{44}, \mathrm{P}_{54}, \mathrm{P}_{55}, \mathrm{P}_{56}\}.$$

При использовании матрицы X *критерий проверки линейной независимости формулируется так: для линейной независимости системы* векторов $\{\mathbf{P}_{ij}\}$ необходимо и достаточно, чтобы из ненулевых элементов матрицы $\mathbf{X}$, отвечающих этим векторам, невозможно было составить замкнутый маршрут (цикл).

Так как из выделенных единицами позиций нельзя составить замкнутый маршрут, то данная система линейно независима и образует базис (рис. 3.3).

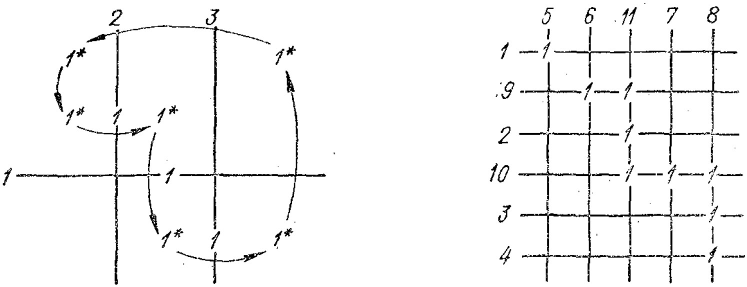


Рис. 3.4.

Введем вектор $\mathbf{P}_{16}$, отметив его знаком «+». Чтобы разложить $\mathbf{P}_{16}$ по векторам системы R, составим цепочку из выделенных элементов, которая замыкается на элементе x_{16}:

$$\mathbf{P}_{16} = \mathbf{P}_{11} - \mathbf{P}_{21} + \mathbf{P}_{22} - \mathbf{P}_{32} + \mathbf{P}_{33} - \mathbf{P}_{43} + \mathbf{P}_{44} - \mathbf{P}_{54} + \mathbf{P}_{56}.$$

При большом размере матрицы $\mathbf{X}$ визуальное отыскание замкнутых цепочек в ней представляет значительные трудности. В таком случае прибегают к формализованному методу вычеркивания.

Метод вычеркивания позволяет выделить в произвольном плане $\mathbf{X}$ Т-задачи замкнутую цепочку, если она существует.

Этот метод состоит в следующем. Выделив в плане $\mathbf{X}$ некоторое множество ненулевых элементов S, выясняют, существуют ли во множестве элементов S циклы. Просматривают одну за другой строки плана $\mathbf{X}$ и вычеркивают строки, не содержащие элементы S, и строки, которые содержат не более одного элемента S. Просмотрев все строки плана $\mathbf{X}$, переходят к столбцам и вычеркивают те, которые содержат не более одного элемента S. При этом элементы, содержащиеся в ранее вычеркнутых строках, в расчет не принимают. Далее повторяют весь процесс, просматривая сначала строки, а потом столбцы оставшейся после вычеркивания строк и столбцов подматрицы.

После конечного числа шагов процесс заканчивается одним из следующих двух исходов: 1) все строки (столбцы) вычеркнуты; 2) получена подматрица, в каждой строке (столбце) которой содержится не менее двух элементов S.

В первом случае из элементов S составить цикл невозможно. Во втором случае множество элементов S содержит циклы из невычеркнутых элементов S.

На рис. 3.4 показаны два плана Т-задачи: неопорный и опорный. Номера линий указывают порядок вычеркивания. Звездочками отмечены элементы, которые вычеркнуть нельзя. Они образуют цикл.

Нахождение начальных опорных планов

Метод северо-западного угла используют для нахождения произвольного опорного плана Т-задачи. Основную идею метода рассмотрим на конкретном примере.

Дана Т-задача с четырьмя пунктами производства $\{a_i\} = 1, 2, 3, 4$ и потребления $\{b_j\} = 5, 1, 2, 2$. Строим матрицу размером 4×4, причем для удобства вычислений снизу от нее выписываем неудовлетворенные потребности, а в столбцах справа — остатки невывезенного продукта (табл. 3.2):

Таблица 3.2

					a_i	$a_i^{(1)}$	$a_i^{(2)}$	$a_i^{(3)}$	$a_i^{(4)}$	$a_i^{(5)}$
	1	0	0	0	1	0	0	0	0	0
	2	0	0	0	2	2	0	0	0	0
$X =$	2	1	0	0	3	3	3	1	0	0
	0	0	2	2	4	4	4	4	4	2
b_j	5	1	2	2						
$b_j^{(1)}$	4	1	2	2						
$b_j^{(2)}$	2	1	2	2						
$b_j^{(3)}$	0	1	2	2						
$b_j^{(4)}$	0	0	2	2						
$b_j^{(5)}$	0	0	0	2						

Заполнение таблицы начинаем с левого верхнего элемента x_{11}, что и обусловило название — метод северо-западного угла. Сравнив $a_1 = 1$ с $b_1 = 5$, выбираем меньшее из них, и получим $x_{11} = 1$. Так как выбор произведен по строке, то остальная часть первой строки должна быть заполнена нулями. Во вспомогательном столбце записываем остатки невывезенного продукта, а в вспомогательной строке — неудовлетворенные потребности после одного шага заполнения.

Переходим ко второй строке и начинаем заполнение с элемента x_{21}. Сравнивая $a_2^{(1)} = 2$ и $b_2^{(1)} = 4$, выбираем меньшее из них и потому

$x_{21} = 2$. Остальную часть второй строки заполним нулями. Процесс заполнения далее продолжаем аналогично.

Полученный план является опорным, так как из его ненулевых перевозок нельзя составить цепочку. Кроме того, он удовлетворяет условиям задачи, так как

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1, 2, 3, 4) \quad \sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1, 2, 3, 4).$$

Формальное описание алгоритма. 1. Определяют верхний левый элемент матрицы $\mathbb{X}$:

$$x_{11} = \min(a_1, b_1).$$

Возможны три случая:

а) если $a_1 < b_1$, то $x_{11} = a_1$, и вся первая строка, начиная со второго элемента, заполняют нулями;

б) если $a_1 > b_1$, то $x_{11} = b_1$, а все оставшиеся элементы первого столбца заполняют нулями;

в) если $a_1 = b_1$, то $x_{11} = a_1 = b_1$, и все оставшиеся элементы первых столбца и строки заполняют нулями.

На этом один шаг метода заканчивается.

2. Пусть уже проделано k шагов. $(k + 1)$-й шаг состоит в следующем.

Определяют верхний левый элемент незаполненной части матрицы $\mathbb{X}$. Пусть это элемент

$$x_{\lambda_\mu} \ (\lambda + \mu = k + 2),$$

причем

$$x_{\lambda_\mu} = \min\{a_\lambda^{(k)}, b_\mu^{(k)}\}, \tag{1.27}$$

где

$$a_\lambda^{(k)} = a_\lambda - \sum_{j=1}^{\mu-1} x_{\lambda j} \quad \text{и} \quad b_\mu^{(k)} = b_\mu - \sum_{i=1}^{\lambda-1} x_{i\mu}. \tag{1.28}$$

Если $a_\lambda^{(k)} < b_\mu^{(k)}$, то заполняем нулями λ-ю строку, начиная с $(\mu + 1)$-го элемента. В противном случае заполняем нулями оставшуюся часть μ-го столбца.

Метод минимального элемента позволяет построить начальный опорный план Т-задачи и является вариантом метода северо-западного угла, учитывающим специфику матрицы $\mathbb{C} = \| c_{ij} \|$. В отличие от метода северо-западного угла данный метод позволяет сразу получить достаточно экономичный план, сокращая общее количество итераций по его оптимизации.

Формальное описание метода. Элементы матрицы $\mathbb{C}$ нумеруют, начиная от минимального в порядке возрастания, а затем в этом же порядке заполняют матрицу $\mathbb{X}_0$.

Пусть элементом с минимальным порядковым номером оказался элемент $x_{ij} = \min\{a_i, b_j\}$.

Возможны три случая: а) если $\min\{a_i, b_j\} = a_i$, то оставшуюся часть i-й строки заполняем нулями; *б*) если $\min\{a_i, b_j\} = b_j$, то оставшуюся часть j-го столбца заполняем нулями;

в) если $a_i = b_j$, то оставшуюся часть строки и столбца заполняем нулями.

Далее этот процесс повторяют с незаполненной частью матрицы.

Пусть элементом с k-м порядковым номером оказался $x_{\lambda\mu}^{(k)}$. Тогда $x_{\lambda\mu} = \min\{a_\lambda^{(k)};\ b_\mu^{(k)}\}$,

где
$$a_\lambda^{(k)} = a_\lambda - \sum_{j=1}^{\mu-1} x_{\lambda j}^{(g)}, \quad g = 1, \ldots, (k-1);$$
$$b_\mu^{(k)} = b_\mu - \sum_{i=1}^{\lambda-1} x_{i\mu}^{(l)}, \quad l = 1, \ldots, (k-1).$$

Возможны три случая: *а*) $a_\lambda^{(k)} < b_\mu^{(k)}$, тогда $x_{\lambda\mu}^{(k)} = a_\lambda^{(k)}$ и оставшуюся часть строки λ заполняют нулями; *б*) $a_\lambda^{(k)} > b_\mu^{(k)}$, тогда $x_{\lambda\mu}^{(k)} = b_\mu^{(k)}$ и остаток столбца μ заполняют нулями, $a_\lambda^{(k)} = b_\mu^{(k)}$ и тогда оставшуюся часть строки λ и столбца μ заполняем нулями.

Пример 3.1. Найти начальный опорный план методом минимального элемента для следующей задачи (табл. 3.3):

Таблица 3.3

B_j / A_i	1	2	3	4	B_j / a_i
1	$7^{(10)}$	$8^{(11)}$	$5^{(7)}$	$3^{(5)}$	11
2	$2^{(3)}$	$4^{(6)}$	$5^{(8)}$	$9^{(12)}$	11
3	$6^{(9)}$	$3^{(4)}$	$1^{(1)}$	$2^{(2)}$	8
A_i / b_j	5	9	9	7	a_i / b_j

Цифры в скобках указывают порядок заполнения элементов в матрице X_0 (табл. 3.4).

Таблица 3.4.

$$\begin{array}{rcccc|cccc}
 & & & & & a_i & & & \\
 & 0 & 3 & 1 & 7 & 11 & 4 & 3 & 0 \\
X_0 = & 5 & 6 & 0 & 0 & 11 & 6 & 0 & \\
 & 0 & 0 & 8 & 0 & 8 & 0 & & \\
\hline
b_j & 5 & 9 & 9 & 7 & & & & \\
 & 0 & 3 & 1 & 0 & & & & \\
 & & 0 & 0 & & & & &
\end{array}$$

Соответствующее значение целевой функции равно

$$L(X_0) = \sum_i \sum_j c_{ji} x_{ji}^{(0)} = 3 \cdot 8 + 1 \cdot 5 + 3 \cdot 7 + 5 \cdot 2 + 6 \cdot 4 + 8 = 92.$$

§ 2. МЕТОД ПОТЕНЦИАЛОВ[1]

Метод потенциалов позволяет, исходя из некоторого опорного плана перевозок, построить за конечное число итераций решение Т-задачи.

Общая схема метода такова. В данном начальном опорном плане каждому пункту ставят в соответствие некоторое число, называемое его *предварительным потенциалом.* Предварительные потенциалы выбирают так, чтобы их разность для любой пары пунктов A_i и B_j, связанных основной коммуникацией, была равна c_{ij}. Если окажется, что разность предварительных потенциалов для всех других коммуникаций не превосходит c_{ij}, то данный план перевозок — оптимальное решение задачи. В противном случае указывают способ улучшения опорного плана Т-задачи.

Описание алгоритма метода потенциалов. Алгоритм складывается из предварительного этапа и конечного числа однотипных итераций.

На предварительном этапе строят начальный опорный план X_0 и составляют матрицу

$$C_1 = \| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \| m, n,$$

где $v_j^{(0)}$, $u_i^{(0)}$ — предварительные потенциалы пунктов A_i $(i = 1, 2, \ldots$ $\ldots, m)$ и B_j $(j = 1, 2, \ldots, n)$.

Предварительный этап. С помощью известного метода (например, метода северо-западного угла) определяют начальный опорный план X_0 и вычисляют предварительные потенциалы $v_j^{(0)}$, $u_i^{(0)}$.

Вычисление предварительных потенциалов производят так. По найденному плану X_0 строят схему Т-задачи из основных коммуникаций плана. Далее образуют следующие множества: J_1— множество индексов всех пунктов B_j, которые связаны с пунктом A_1 основными коммуникациями; I_1 — множество индексов тех пунктов производства A_i, которые связаны с множеством J_1; наконец, J_2 — множество индексов пунктов потребления, которые связаны основными коммуникациями с множеством J_1 и т. д. Образование таких множеств продолжают до тех пор, пока не получат пустое множество.

Поскольку на выполнение условий оптимальности плана оказывают влияние лишь разности $v_j - u_i$, то за начало отсчета (нуль) можно принять потенциал любого из пунктов (см. теорему 3.2).

Полагаем для определенности $u_1^{(0)} = 0$ и вычислим систему потенциалов относительно A_1. Тогда $v_j^{(0)} = c_{1j} + u_1^{(0)} = c_{1j}$, где $j \in J_1$.

По значениям $v_j^{(0)}$ $(j \in J_1)$ определяют потенциалы пунктов A_i $(i \in I_1)$:

$$u_i^{(0)} = v_j^{(0)} - c_{ij} \, (j \in J_1; \; i \in I_1).$$

---

[1] Метод потенциалов, называемый в зарубежной литературе модифицированным распределительным методом, впервые предложен советскими учеными Л. В. Канторовичем и М. К. Гавуриным в 1949 г. [58]. Позже аналогичный метод был предложен Данцигом, который исходил из общих идей ЛП.

Аналогично вычисляют потенциалы $v_j^{(0)}$ $(j \in J_2)$ и т. д. После того как потенциалы всех пунктов найдены, строят матрицу

$$\mathbf{C}_1 = \| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \|.$$

Очевидно, позиции матрицы $\mathbf{C}_1$, отвечающие базисным элементам плана $\mathbf{X}_0$, будут заняты нулями.

Если матрица $\mathbf{C}_1$ не содержит отрицательных элементов, то $\mathbf{X}_0$ — оптимальный план. В противном случае $\mathbf{X}_0$ — неоптимальный план, который может быть улучшен. Тогда переходят к выполнению однотипных итераций.

$(k + 1)$*-я итерация*. Каждая итерация, кроме первой, где отсутствует первый этап, состоит из двух этапов. Предположим, что уже проведено k итераций $(k = 1, 2, \ldots)$, в результате которых получен план $\mathbf{X}_k$ и матрица $\mathbf{C}_k$.

Цель $(k + 1)$-й итерации — построение матрицы $\mathbf{C}_{k+1}$, а также либо установление оптимальности плана $\mathbf{X}_k$, либо нахождение более экономичного плана $\mathbf{X}_{k+1}$.

Первый этап. Вычисляют матрицу $\mathbf{C}_{k+1}$. Преобразование матрицы $\mathbf{C}_k$ в матрицу $\mathbf{C}_{k+1}$ состоит в следующем. Выбирают наибольший по модулю отрицательный элемент матрицы $\mathbf{C}_k$. Пусть это элемент $c^{k}_{\lambda\mu} = \Delta_k$. Выделяют строку, в которой содержится элемент $c^{k}_{\lambda\mu}$, а множество существенных элементов этой строки, не совпадающих с данным элементом, обозначают G_1. При этом $\mathbf{X}_k$*-существенными элементами* называют те элементы матрицы $\mathbf{C}_k$, которые отвечают базисным перевозкам плана $\mathbf{X}_k$.

Затем выделяют столбцы матрицы $\mathbf{C}_k$, которые содержат элементы множества G_1. Множество $\mathbf{X}_k$-существенных элементов, которые находятся в столбцах матрицы $\mathbf{C}_k$ и отличны от G_1 элементов множества, обозначают G_2.

Процесс выделения продолжают до тех пор, пока очередное множество не окажется пустым. Поскольку каждые строка и столбец не могут быть выделены дважды, то весь процесс заканчивается за $l = m + n - 1$ шагов.

Далее строят матрицу $\mathbf{C}_{k+1}$. Для этого величину Δ_k прибавляют ко всем выделенным столбцам и вычитают из всех выделенных строк матрицы $\mathbf{C}_k$. При этом все выделенные $\mathbf{X}_k$-существенные элементы матрицы $\mathbf{C}_k$ остаются равными нулю, а кроме того, в нуль превращается и элемент $c_{\lambda\mu}$.

Если все элементы матрицы $\mathbf{C}_{k+1}$ окажутся неотрицательными, то $\mathbf{X}_k$ — оптимальный план, и на этом процесс заканчивается. В противном случае переходят ко второму этапу.

Второй этап. Производят улучшение плана $\mathbf{X}_k$. Выбирают наибольший по модулю отрицательный элемент матрицы $\mathbf{C}_{k+1}$. Пусть это элемент $c_{i_0 j_0}^{(k+1)} = \Delta_{k+1} < 0$. Затем составляют, применив, например метод вычеркивания, цепочку из положительных элементов плана $\mathbf{X}_k$, которая замыкается на $x_{i_0 j_0}$.

После того как цепочка построена, в ней находят минимальный нечетный по порядку следования элемент: $\theta_{k+1} = \min\limits_{0 \leqslant \mu < S} x^{(k)}_{i_\mu j_{\mu+1}}$.

Прибавляют θ_{k+1} ко всем четным элементам цепочки и к элементу $x_{i_0 j_0}$ и вычитают θ_{k+1} из всех нечетных элементов. Остальные элементы X_k оставляют без изменения.

Новый план X_{k+1} построен. Он является опорным, так как число его ненулевых перевозок не изменилось.

Пусть L_k — величина транспортных издержек, отвечающих плану X_k. Тогда новое значение целевой функции, отвечающее плану X_{k+1}, находят по соотношению

$$L_{k+1} = L_k + \theta_{k+1}\Delta_{k+1}. \qquad (2.1)$$

Так как $\theta_{k+1} > 0$ и $\Delta_{k+1} < 0$, то $L_{k+1} < L_k$. Поэтому X_{k+1} является улучшенным опорным планом.

Затем производят аналогично $(k+2)$-ю итерацию.

Пример 3.2. Решить методом потенциалов Т-задачу, условия которой заданы в табл. 3.5.

Проверим условие баланса $\sum\limits_{i=1}^{3} a_i = 11 + 11 + 8 = 30$, $\sum\limits_{j=1}^{4} b_j = 30$. $\sum\limits_{i} a_i = \sum\limits_{j} b_j$, поэтому задача разрешима.

При решении данной задачи используем следующие обозначения и приемы: X_k — существенные элементы матрицы C_k обозначаем чертой сверху;

при вычислении матрицы C_k помещаем $(-\Delta_k)$ справа от выделенных строк матрицы C_{k-1}, а Δ_k $(\Delta_k < 0)$ — под выделенными столбцами;

минимальный элемент Δ_k матрицы C_k обводим рамкой $\boxed{}$

элементы цепочки в плане X_k отмечаем знаком *;

после каждой итерации выписываем значения целевой функции

$$L_k = L_{k-1} + \theta_k\Delta_k.$$

Предварительный этап. Определяем исходный опорный план X_0 с помощью метода северо-западного угла:

$$\begin{array}{ccccc|ccccc}
 & & & & & a_i & & & & \\
 & 5 & 6 & 0 & 0 & 11 & 6 & 0 & & \\
 & 0 & 3 & 8 & 0 & 11 & 11 & 11 & 8 & 0 \\
 & 0 & 0 & 1 & 7 & 8 & 8 & 8 & 8 & 7 \\
\cline{2-5}
b_j & 5 & 9 & 9 & 7 & & & & & \\
 & 0 & 9 & 9 & 7 & & & & & \\
 & 0 & 3 & 9 & 7 & & & & & \\
 & & 0 & 9 & 7 & & & & & \\
 & & & 1 & 7 & & & & & \\
 & & & 0 & 7 & & & & &
\end{array}$$

$$X_0 = \begin{Vmatrix} 5 & 6 & 0 & 0 \\ 0 & 3 & 8 & 0 \\ 0 & 0 & 1 & 7 \end{Vmatrix}$$

Целевая функция $L_0 = \sum\sum c_{ij}x_{ij}^{(0)} = 150$.

Находим предварительные потенциалы задачи. Для этого строим схему по основным коммуникациям опорного плана X_0 (рис. 3.5). Наносим на основные коммуникации соответствующие величины транспортных издержек.

Таблица 3.5

A_i \ B_j	1	2	3	4	B_j \ a_i
1	7	8	5	3	11
2	2	4	5	9	11
3	6	3	1	2	8
A_i \ b_j	5	9	9	7	b_j \ a_i

Предварительные потенциалы вычисляем в таком порядке. Пусть $u_1^{(0)} = 0$. Рассмотрев коммуникацию A_1B_1, находим $v_1^{(0)} = c_{11} + u_1^{(0)} = 7$. Используя коммуникацию A_1B_2, определим $v_2^{(0)} = c_{12} + u_1^{(0)} = 8 + 0 = 8$. Аналогично: $u_2^{(0)} = v_2^{(0)} - c_{22} = 8 - 4 = 4$; $v_3^{(0)} = c_{23} + u_2^{(0)} = 5 + 4 = 9$; $u_3^{(0)} = v_3^{(0)} - c_{33} = 9 - 1 = 8$; $v_4^{(0)} = c_{34} + u_3^{(0)} = 2 + 8 = 10$.

Теперь вычисляем элементы матрицы $C_1 = \| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \|$, например, $c_{21}^{(1)} = c_{21} - (v_1^{(0)} - u_2^{(0)}) = 2 - (7 - 4) = -1$ и т. д.:

$$C_1 = \left\| \begin{array}{cccc} 0 & 0 & -4 & \boxed{-7} \\ -1 & 0 & 0 & 3 \\ 7 & 3 & 0 & 0 \end{array} \right\|.$$

Так как в C_1 есть отрицательные элементы, *план X_0 — неоптимальный.*

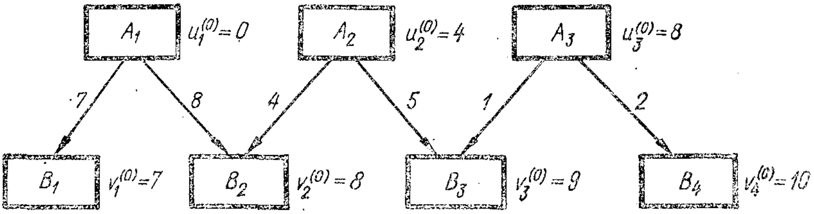


Рис. 3.5.

Первая итерация. Здесь первый этап отсутствует, так как матрица C_1 уже определена.

На втором этапе в матрице C_1 находим наибольший отрицательный элемент $\Delta_1 = c_{14} = -7$. Перейдя к плану X_0, строим цепочку из его базисных элементов, которая замыкается на x_{14}.

Элементы цепочки отмечены звездочками. Нечетные элементы по порядку следования $x_{12}^* = 6$, $x_{23}^* = 8$, $x_{34} = 7$. Определив минимальный элемент $\Theta_1 = \min\{6, 8, 7\} = 6$, прибавим Θ_1 ко всем четным элементам, а также к $x_{14} = 0$ и вычтем из всех нечетных элементов цепочки. X_1 — опорный план, так как число ненулевых перевозок в нем не изменилось и из них нельзя построить замкнутую цепочку. Получим

$$X_0 = \left\| \begin{array}{cccc} 5 & 6^* & 0 & \overset{+}{0} \\ 0 & 3^* & 8^* & 0 \\ 0 & 0 & 1^* & 7^* \end{array} \right\| \xrightarrow{\Theta_1 = 6} X_1 = \left\| \begin{array}{cccc} 5 & 0 & 0 & 6 \\ 0 & 9 & 2 & 0 \\ 0 & 0 & 7 & 1 \end{array} \right\|$$

Целевая функция $L_1 = L_0 + \Delta_1\Theta_1 = 150 - 7 \cdot 6 = 108$.

Вторая итерация. Первый этап заключается в построении матрицы C_2 по матрице C_1. В матрице C_1 чертой сверху отметим X_1-существенные элементы. Вычеркиваем первую строку, содержащую наибольший отрицательный элемент $c_{14} = -7 = \Delta_1$, и находим в ней существенный нуль (т. е. $\overline{0}$), принадлежащий первому столбцу. Поэтому вычеркиваем также первый столбец и просматриваем, нет ли в нем еще существенных нулей. Поскольку таковых нет, процесс вычеркивания заканчивается.

Далее из вычеркнутой строки вычтем $\Delta_1 = -7$ (т. е. прибавим 7), а к выделенному столбцу прибавим $\Delta_1 = -7$ (или вычтем 7). Получим матрицу C_2. Так как матрица C_2 содержит отрицательные элементы, предыдущий план необходимо улучшить.

$$C_1 = \left\| \begin{array}{cccc|c} \overline{0} & 0 & -4 & -7 & +7 \\ -1 & \overline{0} & \overline{0} & 3 & \\ 7 & 3 & \overline{0} & \overline{0} & \\ \hline -7 & & & & \end{array} \right\| \to C_2 = \left\| \begin{array}{cccc} 0 & 7 & 3 & 0 \\ \boxed{-8} & 0 & 0 & 3 \\ 0 & 3 & 0 & 0 \end{array} \right\|$$

На втором этапе находим в плане X_1 элемент x_{21}, соответствующий $c_{21} = \Delta_2 = -8$, и строим цепочку из базисных элементов этого плана, которая замыкается на x_{21}. Нечетные элементы по порядку следования $\overset{*}{x}_{23} = 2$, $\overset{*}{x}_{34} = 1$, $\overset{*}{x}_{11} = 5$. Выбрав $\Theta_2 = \min\{2, 1, 5\} = 1$, прибавим Θ_2 ко всем четным элементам, а также к x_{21} и вычтем из нечетных элементов цепочки. Получим новый улучшенный план X_2.

$$X_1 = \left\| \begin{array}{cccc} 5^* & 0 & 0 & 6^* \\ 0^+ & 9 & 2^* & 0 \\ 0 & 0 & 7^* & 1^* \end{array} \right\| \xrightarrow{\theta_2=1} X_2 = \left\| \begin{array}{cccc} 4 & 0 & 0 & 7 \\ 1 & 9 & 1 & 0 \\ 0 & 0 & 8 & 0 \end{array} \right\|$$

Целевая функция $L_2 = L_1 + \Theta_2\Delta_2 = 108 - 8 \cdot 1 = 100$.

Последующие итерации проводим аналогично.

Третья итерация. Первый этап.

$$C_2 = \left\| \begin{array}{cccc|c} \overline{0} & 7 & 3 & 0 & \\ \boxed{-8} & \overline{0} & \overline{0} & 3 & +8 \\ 0 & 3 & \overline{0} & 0 & +8 \\ \hline & -8 & -8 & & \end{array} \right\| \to C_3 = \left\| \begin{array}{cccc} 0 & -1 & \boxed{-5} & 0 \\ 0 & 0 & 0 & 11 \\ 8 & 3 & 0 & 8 \end{array} \right\|$$

$$\Delta_3 = -5$$

Второй этап.

$$X_2 = \left\| \begin{array}{cccc} 4^* & 0 & 0^+ & 7 \\ 1^* & 9 & 1^* & 0 \\ 0 & 0 & 8 & 0 \end{array} \right\| \xrightarrow[\theta_3=1]{} X_3 = \left\| \begin{array}{cccc} 3 & 0 & 1 & 7 \\ 2 & 9 & 0 & 0 \\ 0 & 0 & 8 & 0 \end{array} \right\|$$

$\theta_3 = \min\{4, 1\} = 1$; $L_3 = 95$.

Четвертая итерация. Первый этап.

$$C_3=\left\|\begin{array}{cccc|c} 0 & -1 & \boxed{-5} & 0 & +5 \\ 0 & 0 & 0 & 11 & +5 \\ 8 & 3 & \bar{0} & 8 & \\ \hline -5 & -5 & & -5 & \end{array}\right\| \to C_4=\left\|\begin{array}{cccc} 0 & -1 & 0 & 0 \\ 0 & 0 & 5 & 11 \\ 3 & \boxed{-2} & 0 & 3 \end{array}\right\|$$

Второй этап.

$$\begin{tikzcd}[row sep=small]
 & 3^* \arrow[d, bend left] & 0 & 1^* \arrow[ll, bend right] & 7 \\
X_3= & 2^* \arrow[r, bend right] & 9^* \arrow[d, bend left] & 0 & 0 \arrow[r, "\theta_4=3"'] & X_4= \\
 & 0 & 0^+ \arrow[r, bend right] & 8^* \arrow[uu, bend right] & 0
\end{tikzcd}$$

$$X_4=\left\|\begin{array}{cccc} 0 & 0 & 4 & 7 \\ 5 & 6 & 0 & 0 \\ 0 & 3 & 5 & 0 \end{array}\right\|$$

$$\theta_4=\min\{8,3,9\}=3;\ L_4=95-3\cdot 2=89.$$

Пятая итерация. Первый этап.

$$C_4=\left\|\begin{array}{cccc|c} 0 & -1 & 0 & 0 & +2 \\ 0 & 0 & 5 & 11 & \\ 3 & -2 & 0 & 3 & +2 \\ \hline & & -2 & -2 & \end{array}\right\| \to C_5=\left\|\begin{array}{cccc} 2 & 1 & 0 & 0 \\ 0 & 0 & 3 & 9 \\ 5 & 0 & 0 & 3 \end{array}\right\|$$

Все элементы $c_{ij}^{(5)} \geqslant 0$. Следовательно, $\mathbf{X}_4$ — искомое решение задачи. Целевая функция $L_4 = 89$.

Связь метода потенциалов с симплекс-методом. Пусть $\mathbf{X}_k = \| x_{ij}^{(k)} \|$ — некоторый опорный план Т-задачи. Множество, состоящее из базисных векторов $\mathbf{P}_{ij}$, которые отвечают основным коммуникациям плана $\mathbf{X}$, обозначим K_x, т. е. K_x состоит из $(m + n - 1)$ векторов.

На первом этапе метода потенциалов вычисляют предварительные потенциалы $v_1^{(0)}, v_2^{(0)}, \ldots, v_n^{(0)}, u_1^{(0)}, u_2^{(0)}, \ldots, u_m^{(0)}$, которые удовлетворяют условиям: $v_j^{(0)} - u_i^{(0)} = c_{ij}$ для всех векторов $\mathbf{P}_{ij} \in K_x$. Величина $c_{ij} - (v_j - u_i)$ имеет смысл симплекс-разности.

Предположим, что $c_{ij} - (v_j - u_i) \geqslant 0$ для всех $\mathbf{P}_{ij} \notin K_x$. Это означает, что при введении любого из векторов $\mathbf{P}_{ij}$ уменьшить целевую функцию $L(\mathbf{X})$ нельзя, а поэтому план $\mathbf{X}$ — оптимальный. Если же для некоторого вектора $\mathbf{P}_{i_0 j_0}$ величина $v_{j_0} - u_{i_0} > c_{i_0 j_0}$, то при введении вектора $\mathbf{P}_{i_0 j_0}$ в базис можно уменьшить значение целевой функции $L(\mathbf{X})$.

Переход к новому базису, который включает вектор $\mathbf{P}_{i_0 j_0}$, связан с построением цепочки, которая замыкается на элементе $x_{i_0 j_0}$. Чтобы определить, какой вектор выводится из базиса, разлагают $\mathbf{P}_{i_0 j_0}$ по векторам старого базиса. Далее отыскивают

$$\theta_k = \min_{(i,j)\in K_x} \left\{ \frac{x_{ij}}{x_{ij,i_0 j_0}} \right\} = \frac{x_{i_k j_k}}{x_{i_k j_k i_0 j_0}}, \tag{2.2}$$

где $x_{i_k j_k}$ — базисное значение переменной при векторе $\mathbf{P}_{i_k j_k}$, выводимом из базиса; $x^{i}{}_{k^{j}{}_{k}{}^{i_0 j_0}}$ — коэффициент при выводимом из базиса векторе $\mathbf{P}_{i_k j_k}$ в разложении для вектора $\mathbf{P}_{i_0 j_0}$.

Для Т-задачи всегда $x_{ij,i_0 j_0} = \pm 1$. При этом $x_{ij,i_0 j_0} = +1\ (-1)$, если вектор базиса $\mathbf{P}_{ij}$ занимает нечетную (четную) позицию в маршруте, который замыкается на элементе $x_{i_0 j_0}$. Новые базисные значения переменных в общем случае определяются по формулам

$$x_{ij}^{(\text{н})} = x_{ij} - \theta_k \cdot x_{ij,i_0 j_0}, \quad (i,\ j) \in \breve{K}_x \tag{2.3}$$

$$x_{i_k j_k}^{(\text{н})} = 0,$$

где i_k, j_k — индексы выводимого из базиса вектора.

Поскольку $x_{ij,i_0 j_0} = \pm 1$, формулы (2.2) и (2.3) применительно к Т-задаче можно представить следующим правилом изменения перевозок: перевозки, запланированные по нечетным (четным) коммуникациям, уменьшаются (увеличиваются) на величину θ, перевозка между A_{i_0} и B_{j_0} полагается равной θ, а остальные перевозки сохраняют свои значения, где θ — минимальная нечетная по порядку следования перевозка маршрута, который соединяет пункты A_{i_0} и B_{j_0}.

Таким образом, метод потенциалов является частным случаем симплекс-метода, учитывающим специфику ограничений Т-задачи.

Решение транспортной задачи при вырожденном опорном плане

Опорный план называется *вырожденным*, если число его ненулевых перевозок k меньше ранга матрицы ограничений.

В процессе построения начального плана или при его улучшении очередной план может оказаться вырожденным.

Рассмотрим два случая:

1. Вырожденный план является начальным (X_0). Тогда выбирают некоторые нулевые элементы матрицы X_0 в качестве базисных элементов так, чтобы при этом не нарушалось условие опорности.

Далее данные элементы матрицы заменяют на $\varepsilon > 0$ (где ε — произвольное, бесконечно малое число) и рассматривают как обычные базисные элементы плана.

Задачу решают как невырожденную, а в оптимальном плане вместо ε пишут нули.

2. Вырожденный план получается при построении X_{k+1}, если цепочка содержит не менее двух минимальных нечетных элементов. В таком случае в матрице X_{k+1} полагают равным нулю только один из этих элементов, остальные заменяют на ε и решают задачу как невырожденную. Если на k-м шаге $\theta_k = \varepsilon$, то при переходе от $X_k \to X_{k+1}$ целевая функция не изменяется, а в базис вводится элемент, для которого перевозка станет равной ε.

Пример 3.3. Решить Т-задачу со следующими условиями (табл.3.6), Проверим условие баланса

$$\sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j = 24.$$

Таблица 3.6

b_j / a_i	4	6	8	6
6	$2^{(5)}$	$2^{(4)}$	$3^{(8)}$	$4^{(11)}$
8	$6^{(12)}$	$4^{(10)}$	$3^{(9)}$	$1^{(3)}$
10	$1^{(1)}$	$2^{(6)}$	$2^{(7)}$	$1^{(2)}$

Предварительный этап. Строим начальный опорный план методом минимального элемента:

$$
\begin{array}{rcccc|ccc}
 & & & & & a_i & & \\
 & \varepsilon\varnothing & 6 & 0 & 0 & 6 & 0 & \\
X_0 = & 0 & \varepsilon\varnothing & 8 & 0 & 8 & 0 & \\
 & 4 & 0 & 0 & 6 & 10 & 6 & 0 \\
\cline{2-5}
 & 4 & 6 & 8 & 6 & & & \\
b_j & 0 & 0 & 0 & 0 & & &
\end{array}
$$

$m+n-1=6,\ k=4.$

План оказался вырожденным.

Два нулевых элемента делаем базисными так, чтобы не нарушить опорность плана. Выберем в качестве базисных элементов $x_{11}^{(0)}$ и $x_{22}^{(0)}$ и положим их равными ε.

Схема перевозок для плана X_0 показана на рис. 3.6.

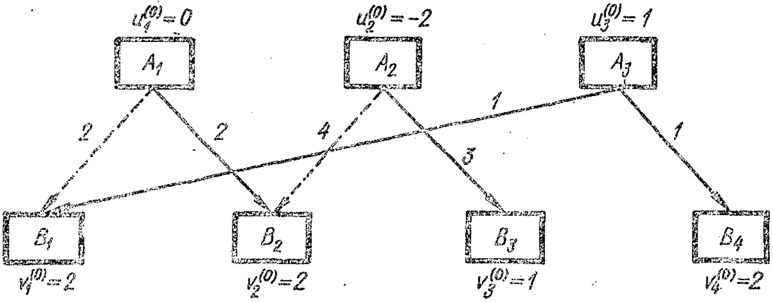


Рис. 3.6.

Для вычисления предварительных потенциалов выберем начальный пункт A_1 и допустим, что $u_1=0$. Потенциалы всех остальных пунктов вычисляем по формулам:

$$u_i^{(0)} = v_j^{(0)} - c_{ij},\ v_j^{(0)} = c_{ij} + u_i^{(0)}.$$

Для проверки оптимальности плана X_0 строим матрицу C_1. Элементы матрицы C_1 вычисляем по соотношению

$$C_1 = \| c_{ij}^{(1)} \| = \| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \| \qquad \begin{array}{l} i=\overline{1,\ m} \\ j=\overline{1,\ n}. \end{array}$$

$$C_1 = \left\| \begin{array}{cccc} 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & \boxed{-3} \\ 0 & 1 & 2 & 0 \end{array} \right\|$$

Так как в матрице C_1 элемент $c_{24} = -3$, то план X_0 неоптимальный и, следовательно, должен быть улучшен.

Первая итерация. Второй этап

$$C_{24} = -3 = \Delta_1$$

$$X_0 = \left\| \begin{matrix} \varepsilon^* & 6^* & 0 & 0 \\ 0 & \varepsilon^* & 8 & 0^+ \\ 4^* & 0 & 0 & 6^* \end{matrix} \right\| \quad \theta_1 = \min\{6, \varepsilon, \varepsilon\} = \varepsilon \longrightarrow$$

$$\longrightarrow X_1 = \left\| \begin{matrix} \varepsilon\cancel{0} & 6 & 0 & 0 \\ 0 & 0 & 8 & \varepsilon \\ 4 & 0 & 0 & 6 \end{matrix} \right\|$$

В результате такого построения плана X_1 минимальный нечетный элемент обращается в нуль и становится небазисным, а вместо него в базис вводим $x_{24} = \varepsilon$. План X_1 является также опорным.

План X_1 является вырожденным (два минимальных нечетных элемента).

Один из нулевых нечетных элементов x_{11} заменяем на ε.

Вторая итерация. Первый этап.

$$C_1 = \left\| \begin{matrix} \bar{0} & \bar{0} & 2 & 2 \\ 2 & 0 & \bar{0} & -3 \\ \bar{0} & 1 & 2 & \bar{0} \\ & & -3 & \end{matrix} \right\| \xrightarrow{+3} C_2 = \left\| \begin{matrix} 0 & 0 & -1 & 2 \\ 5 & 3 & 0 & 0 \\ 0 & 1 & \boxed{-1} & 0 \end{matrix} \right\|$$

$$\Delta_2 = c_{33} = -1.$$

Второй этап.

$$X_1 = \left\| \begin{matrix} \varepsilon & 6 & 0 & 0 \\ 0 & 0 & 8^* & \varepsilon^* \\ 4 & 0 & 0^+ & 6^* \end{matrix} \right\|$$

$$\theta_3 = \min\{8, 6\} = 6$$

$$\longrightarrow X_2 = \left\| \begin{matrix} \varepsilon & 6 & 0 & 0 \\ 0 & 0 & 2 & 6 \\ 4 & 0 & 6 & 0 \end{matrix} \right\|$$

Третья итерация. Первый этап.

$$C_2 = \left\| \begin{array}{cccc|c} \bar{0} & \bar{0} & -1 & 2 & +1 \\ 5 & 3 & \bar{0} & \bar{0} & \\ \bar{0} & 1 & \boxed{-1} & 0 & +1 \\ \hline -1 & -1 & & & \end{array} \right\| \longrightarrow C_3 = \left\| \begin{matrix} 0 & 0 & 0 & 3 \\ 4 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{matrix} \right\|$$

Так как в матрице C_3 нет отрицательных элементов, план X_2 является оптимальным.

§ 3. ВЕНГЕРСКИЙ МЕТОД[1]

Венгерский метод является одним из наиболее распространенных методов решения транспортных задач.

Рассмотрим сначала основные идеи венгерского метода на примере решения задачи выбора или задачи о назначениях, которая является частным случаем Т-задачи, а затем обобщим этот метод для произвольной Т-задачи.

Венгерский метод для задачи выбора

Постановка задачи. Предположим, что имеется n различных работ $A_1, A_2, \ldots, A_n$ и механизмов $B_1, B_2, \ldots, B_n$, каждый из которых может выполнять любую работу. Производительность механизма B_i при выполнении работы A_j обозначим c_{ij} ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n$). Требуется так распределить механизмы по работам, чтобы суммарный эффект от их использования был максимален. Такая задача называется *задачей выбора.*

Формально задача выбора записывается так. Необходимо выбрать такую последовательность элементов $c_{1j_1}, c_{2j_2}, \ldots, c_{nj_n}\}$ из матрицы

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ c_{n1} & c_{n2} & \ldots & c_{nn} \end{bmatrix};$$

чтобы сумма $\sum_{k=1}^{n} c_{kj_k}$ была максимальна и при этом из каждой строки и столбца $\mathbf{C}$ был выбран только один элемент.

Введем следующие понятия:

1. Нулевые элементы $z_1, z_2, \ldots, z_k$ матрицы $\mathbf{C}$ называются *независимыми нулями,* если для любого $1 \leqslant i \leqslant k$ строка и столбец, на пересечении которых расположен элемент z_i, не содержат другие нули z_k (для всех $k \neq i$).

2. Две прямоугольные матрицы $\mathbf{C}$ и $\mathbf{D}$ называются эквивалентными ($\mathbf{C} \sim \mathbf{D}$), если $c_{ij} = d_{ij} + \alpha_i + \beta_j$ для всех i, j. Задачи выбора, определяемые эквивалентными матрицами, являются эквивалентными.

3. Элементы, расположенные в выделенных строках или столбцах, называются выделенными элементами.

Описание алгоритма венгерского метода. Алгоритм состоит из предварительного этапа и не более чем $(n - 2)$ последовательно проводимых итераций. Каждая итерация связана с эквивалентными преобразованиями матрицы, полученной в результате проведения предыдущей итерации, и с выбором максимального числа независимых

---

[1] Этот метод впервые предложен венгерским математиком Эгервари в 1931 г. Длительное время работа оставалась малоизвестной. В 1953 г. математик Г. Кун [58] перевел эту работу на английский язык, заново открыв ее для специалистов, развил идеи Эгервари и усовершенствовал метод, который в честь первого автора и был назван венгерским.

нулей. Окончательным результатом итерации является увеличение числа независимых нулей на единицу.

Как только количество независимых нулей станет равным n, проблема выбора оказывается решенной, а оптимальный вариант определяется позициями независимых нулей в последней матрице.

Предварительный этап. Разыскивают максимальный элемент в j-м столбце и все элементы этого столбца последовательно вычитают из максимального. Эту операцию проделывают над всеми столбцами матрицы $\mathfrak{C}\,(1 \leqslant j \leqslant n)$. В результате образуется матрица с неотрицательными элементами, в каждом столбце которой имеется по крайней мере один нуль.

Рассматриваем i-ю строку полученной матрицы и из каждого ее элемента вычитаем минимальный элемент этой строки. Меняя i от 1 до n, получаем матрицу $\mathfrak{C}_0$ с неотрицательными элементами, в каждом столбце и строке которой имеется по крайней мере один нуль. Отмечаем произвольный нуль в первом столбце звездочкой.

Затем просматриваем второй столбец, и если в нем есть нуль, расположенный в строке, где нет нуля со звездочкой, то отмечаем его звездочкой.

Аналогично просматриваем один за другим все столбцы матрицы $\mathfrak{C}_0$. Очевидно, что нули матрицы $\mathfrak{C}_0$, отмеченные звездочкой, являются по построению независимыми. На этом предварительный этап заканчивается.

$(k+1)$-я *итерация*. Допустим, что k-я итерация уже проведена и в результате получена матрица $\mathfrak{C}_k$. Если в матрице $\mathfrak{C}_k$ имеется ровно n нулей со звездочкой, то процесс решения заканчивается. Если же число нулей со звездочкой меньше n, то переходим к $(k+1)$-й итерации.

Каждая итерация начинается первым и заканчивается вторым этапом. Между ними может несколько раз проводиться пара этапов: третий — первый. Перед началом итерации знаком «+» выделяют столбцы матрицы $\mathfrak{C}_k$, которые содержат нули со звездочкой.

Первый этап. Просматривают невыделенные столбцы матрицы $\mathfrak{C}_k$. Если среди них не окажется нулевых элементов, то переходят к третьему этапу.

Если же невыделенный нуль матрицы $\mathfrak{C}_k$ обнаружен, то возможен один из двух случаев: 1) строка, содержащая невыделенный нуль, содержит также и нуль со звездочкой; 2) эта строка не содержит нуля со звездочкой.

В первом случае невыделенный нуль отмечают штрихом и выделяют строку, в которой он содержится, постановкой справа от нее знака «+». Затем уничтожают знак «+», обводя его кружком над тем столбцом, на пересечении которого с данной выделенной строкой содержится нуль со звездочкой.

Далее просматривают этот столбец, отыскивают в нем невыделенный нуль (нули), не отмеченный звездочкой, отмечают его штрихом и выделяют строку (строки), содержащую такой нуль (нули). Затем просматривают эту строку (строки), отыскивая в них нуль со звездочкой.

Этот процесс за конечное число шагов заканчивается одним из следующих исходов:

все нули матрицы C_k выделены, т. е. находятся в выделенных строках или столбцах. При этом переходят к третьему этапу;

имеется невыделенный нуль в строке, где нет нуля со звездочкой. Тогда переходят ко второму этапу, отметив последний по порядку нуль штрихом.

Во втором случае, отметив невыделенный нуль штрихом, сразу переходят ко второму этапу.

Второй этап. Строят следующую цепочку из элементов матрицы C_k: исходный нуль со штрихом, нуль со звездочкой, расположенный в одном столбце с первым, нуль со штрихом, расположенный в одной строке с предшествующим нулем со звездочкой, и т. д. Итак, цепочка образуется передвижением от $0'$ к 0^* по столбцу, от 0^* к $0'$ по строке и т. д.

Можно доказать, что описанный алгоритм построения цепочки однозначен и конечен. При этом цепочка всегда начинается и заканчивается нулем со штрихом. Далее над элементами цепочки, стоящими на нечетных местах ($0'$), ставим звездочки, уничтожая их над четными элементами (0^*). Затем уничтожаем все штрихи над элементами матрицы C_k и знаки $+$. При этом количество независимых нулей будет увеличено на единицу. $(k+1)$-я итерация закончена.

Третий этап. К этому этапу переходят после первого, если все нули матрицы C_k выделены, т. е. находятся на выделенных строках или столбцах. В таком случае среди невыделенных элементов матрицы C_k выбирают минимальный и обозначают его $h > 0$. Далее вычитают h из всех элементов матрицы C_k, расположенных в невыделенных строках, и прибавляют ко всем элементам, расположенным в выделенных столбцах. Получают новую матрицу $C_k^{(1)}$, эквивалентную C_k.

Поскольку среди невыделенных элементов матрицы $C_k^{(1)}$ появятся новые нули (согласно определению), переходят к первому этапу, а вместо матрицы C_k рассматривают матрицу $C_k^{(1)}$. Завершив первый этап либо переходят ко второму этапу, либо вновь возвращаются к третьему этапу, если все нули матрицы $C_k^{(1)}$ оказываются выделенными.

В первом случае после проведения второго этапа итерация заканчивается, а во втором — после проведения третьего этапа получают матрицу $C_k^{(2)} \sim C_k^{(1)} \sim C_k$. В матрице $C_k^{(2)}$ будут невыделенные нули, и всю последовательность операций, начиная с первого этапа, надо повторить. После конечного числа повторений очередной первый этап обязательно закончится переходом на второй этап и количество независимых нулей увеличится на единицу $(k+1)$-я итерация закончена.

Пример 3.4. Решить задачу выбора, которая определяется матрицей:

$$C = \begin{Vmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{Vmatrix}$$

При решении задачи используем следующие обозначения: знак выделения +, подлежащий уничтожению, обводим кружком. Цепочку во втором этапе указываем стрелками.

Предварительный этап. Максимальный элемент первого столбца матрицы C равен 4. Поэтому для получения первого столбца матрицы C' необходимо из 4 вычесть элементы первого столбца исходной матрицы. Аналогично, для получения второго, третьего, четвертого и пятого столбцов вычитаем элементы этих столбцов из 5, 3, 2 и 3 соответственно. Получим матрицу C'. Так как в каждой строке матрицы C' есть нуль , то $C' = C_0$. Находим независимые нули C_0 и отметим их звездочкой. В результате получим два независимых нуля (в первом и в четвертом столбцах). Отметим их звездочкой.

Первая итерация. Выделяем знаком + первый и четвертый столбцы, которые содержат 0^*. Просматриваем невыделенные нули матрицы C_0. Отмечаем штрихом нуль, расположенный во второй строке и во втором столбце. Поскольку в этой строке имеется 0^*, то строка подлежит выделению (ставим + справа от второй строки) и одновременно обводим кружком знак + над первым столбцом.

Обращаемся к невыделенным нулям первого стобца. Отмечаем нуль этого столбца, лежащий в третьей строке, которая не содержит 0^*. Следовательно, в этом случае необходимо переходить ко второму этапу.

Второй этап. От последнего по порядку $0'$ движемся к 0^* (первый столбец, вторая строка), затем от 0^*, расположенного во второй строке, переходим к $0'$, расположенному в той же строке. Поскольку во втором столбце, где расположен $0'$, нет 0^*, процесс образования цепочки закончен. Искомая цепочка состоит из элементов: $0'_{31}$; 0^*_{21}; $0'_{22}$. Для завершения второго этапа, а вместе с ним всей первой интерации, ставим звездочки над нулями, которые отмечены штрихом, уничтожаем звездочку над единственным четным элементом и все знаки выделения.

Итак, в результате первой итерации число независимых нулей увеличилось на единицу.

Первая итерация

Первый, второй этапы

Вторая итерация

Первый этап

Третий этап

$$C_0=\left\|\begin{array}{ccccc} \overset{\oplus}{1} & 1 & 1 & \overset{+}{0^*} & 2\\ 0^* & 0' & 0 & 1 & 0\\ 0' & 2 & 2 & 1 & 2\\ 1 & 4 & 1 & 0 & 1\\ 3 & 2 & 2 & 0 & 2 \end{array}\right\| \begin{array}{c} \\ + \\ \\ \\ \\ \end{array} \rightarrow C_1=\left\|\begin{array}{ccccc} \overset{+}{1} & \overset{\oplus}{1} & 1 & \overset{+}{0^*} & 2\\ 0 & 0^* & 0' & 1 & 0\\ 0^* & 2 & 2 & 1 & 2\\ 1 & 4 & 1 & 0 & 1\\ 3 & 2 & 2 & 0 & 2 \end{array}\right\| \begin{array}{c} \\ + \\ h=1 \\ \\ \\ \end{array}$$

Вторая итерация

Первый этап

Второй этап

$$C_1'=\left\|\begin{array}{ccccc} \overset{+}{1} & \overset{\oplus}{0} & 0 & \overset{\oplus}{0^*} & 1\\ 1 & 0^* & 0' & 2 & 0\\ 0^* & 1 & 1 & 1 & 1\\ 1 & 3 & 0 & 0 & 0\\ 3 & 1 & 1 & 0' & 1 \end{array}\right\| \begin{array}{c} + \\ + \\ \\ \\ \\ \end{array} \rightarrow C_2=\left\|\begin{array}{ccccc} 1 & 0^* & 0 & 0 & 1\\ 1 & 0 & 0^* & 2 & 0\\ 0^* & 1 & 1 & 1 & 1\\ 1 & 3 & 0 & 0 & 0\\ 3 & 1 & 1 & 0^* & 1 \end{array}\right\|$$

Третья итерация

Первый этап Второй этап

$$C_2=\left\|\begin{array}{ccccc} \overset{+}{1} & \overset{+}{0}{}^* & \overset{+}{0} & \overset{+}{0} & 1\\ 1 & 0 & 0^* & 2 & 0\\ 0^* & 1 & 1 & 1 & 1\\ 1 & 3 & 0 & 0 & 0'\\ 3 & 1 & 1 & 0^* & 1\end{array}\right\| \begin{array}{l} \\ \\ \rightarrow C_3= \\ + \\ \\ \end{array} \left\|\begin{array}{ccccc} 1 & 0^* & 0 & 0 & 1\\ 1 & 0 & 0^* & 2 & 0\\ 0^* & 1 & 1 & 1 & 1\\ 1 & 3 & 0 & 0 & 0^*\\ 3 & 1 & 1 & 0^* & 1\end{array}\right\|$$

Поскольку последующие итерации проводятся аналогично, выше приведены результаты без дополнительных пояснений.

После третьей итерации количество независимых нулей (0*) стало равно размерности матрицы C и потому процесс выбора закончен. Искомые элементы матрицы соответствуют позициям независимых нулей матрицы C_3 (т. е. нулей со звездочкой).

Целевая функция $L=c_{12}+c_{23}+c_{31}+c_{45}+c_{54}=4+3+4+2+2=15$.

Венгерский метод для транспортной задачи

Рассмотренная выше задача выбора представляет частный случай Т-задачи с единичными объемами производства и потребления. Поэтому венгерский метод, применимый для решения транспортной задачи специального вида, можно распространить на общий случай Т-задачи [57, 58].

Пусть требуется решить Т-задачу следующего типа:
найти

$$\min \sum_i \sum_j c_{ij}x_{ij},$$

при

$$\sum_{j=1}^{n} x_{ij}=a_i, \quad i=1, 2, \ldots, m; \quad \sum_{i=1}^{m} x_{ij}=b_j, \quad j=1, 2, \ldots, n; \; x_{ij}\geqslant 0.$$

Алгоритм решения Т-задачи, основанный на венгерском методе, состоит из предварительного этапа и конечного числа итераций (как и для задачи выбора).

В результате предварительного этапа строят матрицу $X_0=\|x_{ij}\|_{m,n}$, элементы которой удовлетворяют следующим условиям:

$$\sum_{j=1}^{n} x_{ij}^{(0)} \leqslant a_i, \qquad i=1, 2, \ldots, m; \tag{3.1}$$

$$\sum_{i=1}^{m} x_{ij}^{(0)} \leqslant b_j, \qquad j=1, 2, \ldots n. \tag{3.2}$$

Если в условиях (3.1) и (3.2) получены строгие равенства, то матрица $\mathbf{X}_0$ является решением Т-задачи. Матрицу, построенную в результате k-й итерации, обозначим $\mathbf{X}_k = \| x_{ij}^{(k)} \|_{m,n}$.
Пусть

$$\sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2 \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^{(k)} = \Delta_k. \tag{3.3}$$

Величина Δ_k называется *суммарной невязкой* для матрицы $\mathbf{X}_k$. Она характеризует близость матрицы $\mathbf{X}_k$ к опорному плану Т-задачи. Итерации проводятся до тех пор, пока величина Δ_k не станет при некотором k равна нулю.

Описание алгоритма венгерского метода. Предварительный этап. В каждом из столбцов матрицы транспортных издержек $\mathbf{C} = \| c_{ij} \|$ отыскивают минимальный элемент, который вычитают из всех элементов этого столбца. Получают матрицу $\mathbf{C}'$. Далее в каждой строке матрицы $\mathbf{C}'$ выбирают минимальный элемент и вычитают его из всех элементов рассматриваемой строки. Приходят к матрице $\mathbf{C}_0$, все элементы которой неотрицательны, причем в каждой строке и столбце имеем по крайней мере один нуль.

Далее строят матрицу $\mathbf{X}_0$ так, чтобы ее ненулевые элементы были расположены в позициях нулей матрицы $\mathbf{C}_0$.

Пусть i_{k_j} — номер строки, в которой расположен k-й нуль j-го столбца матрицы $\mathbf{C}_0$. Тогда элементы первого столбца матрицы $\mathbf{X}_0$ определяют по рекуррентной формуле

$$x_{ij}^{(0)} = \left\{ \begin{array}{ll} 0; \quad i \neq i_{k_1}, & k = 1, 2, \ldots, r \\ \min\left(a_i;\ b_1 - \sum\limits_{\mu=1}^{i-1} x_{\mu 1}^{(0)}\right); & i = i_{k_1} \end{array} \right\}. \tag{3.4}$$

Все элементы первого столбца $\mathbf{X}_0$, которым соответствуют ненулевые элементы в матрице $\mathbf{C}_0$, заполняют нулями, а остальные элементы этого столбца заполняют по методу северо-западного угла.

Допустим, что столбцы матрицы $\mathbf{X}_0$ до $(j-1)$-го включительно уже заполнены. Тогда элементы j-го столбца определяют в соответствии с формулой

$$x_{ij}^{(0)} = \left\{ \begin{array}{ll} 0; \quad i \neq i_{k_j} & k = 1, 2, \ldots, r_j \\ \min\left\{a_i - \sum\limits_{\mu=1}^{j-1} x_{i\mu}^{(0)};\ b_j - \sum\limits_{\mu=1}^{i-1} x_{\mu j}^{(0)}\right\}_{i=i_{k_j}} & \end{array} \right\}. \tag{3.5}$$

Далее вычисляют невязку:

$$\Delta_0 = \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2 \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^{(0)}. \tag{3.6}$$

Если $\Delta_0 = 0$, то $\mathbf{X}_0$ — оптимальное решение Т-задачи.

Если же $\Delta_0 > 0$, то переходим к первой итерации.

$(k+1)$-*я итерация.* Каждая итерация состоит из трех этапов. Итерация начинается первым этапом, а заканчивается вторым.

Между первым и вторым этапами в общем случае могут быть несколько раз проведены первый и третий этапы.

Допустим, что уже проведено k итераций, причем $\Delta_k > 0$. В этом случае необходимо, используя матрицы $\mathbf{C}_k$ и $\mathbf{X}_k$, провести следующую $(k+1)$ итерацию. Перед началом итерации выделяют знаком «+» те столбцы матрицы $\mathbf{C}_k$, для которых невязки по столбцам равны

$$\delta_j^{(k)} = b_j - \sum_{i=1}^{m} x_{ij}^{(k)} = 0.$$

Первый этап. Если все нулевые элементы матрицы $\mathbf{C}_k$ окажутся в выделенных столбцах, то переходят к третьему этапу. В противном случае выбирают произвольный невыделенный нуль $\mathbf{C}_k$, расположенный, например, в i-й строке и j-м столбце, и вычисляют величину невязки i-й строки:

$$\delta_i^{(k)} = a_i - \sum_{\mu=1}^{n} x_{i\mu}^{(k)}.$$

При этом возможен один из двух случаев:

1) $\delta_i^{(k)} = 0$; 2) $\delta_i^{(k)} > 0$.

В первом случае i-ю строку $\mathbf{C}_k$ отмечают знаком + справа от нее, а элемент матрицы $c_{ij}^{(k)}$, равный нулю, отмечают штрихом. Далее просматривают все выделенные столбцы матрицы $\mathbf{C}_k$ по этой строке. Если μ-й столбец матрицы $\mathbf{C}_k$ выделен и элемент $x_{i\mu}^{(k)} > 0$, то знак «+» над столбцом уничтожают, а сам элемент $c_{i\mu}^{(k)}$ [1] (он обязательно равен нулю) отмечают звездочкой.

Затем просматривают μ-й столбец и отыскивают его нулевые элементы $c_{r\mu}$, расположенные в отличных от i-й строках. Если таковые имеются, то их выделяют штрихом.

Далее процесс выделения нулей продолжают аналогично, завершая одним из следующих двух исходов:

1) обнаружен невыделенный нуль матрицы $\mathbf{C}_k$, для которого невязка $\delta_t > 0$. В этом случае переходим ко второму этапу;

2) все нули матрицы $\mathbf{C}_k$ оказались выделенными, причем для каждого из нулей, выделяемых штрихом, невязка $\delta_i = 0$. Тогда переходим к третьему этапу.

Во втором случае, отметив этот нуль штрихом, сразу переходим ко второму этапу.

Второй этап следует за первым, если $\delta_i > 0$, и состоит в построении цепочки из нулей матрицы $\mathbf{C}_k$, отмеченных штрихами и звездочками, и в последующем переходе к новой матрице $\mathbf{X}_{k+1}$.

Пусть для некоторого невыделенного нуля матрицы $\mathbf{C}_k$, расположенного, например, в позиции $\lambda_1\mu_1$, невязка $\delta_{\lambda_1} > 0$. Начиная с этого элемента, строят цепочку из отмеченных нулей матрицы $\mathbf{C}_k$ в соответствии со следующим правилом.

В столбце μ_1 матрицы $\mathbf{C}_k$ выбирают нуль со звездочкой $0^*_{\lambda_2\mu_1}$, в λ_2-й строке выбираем нуль со штрихом $0'_{\lambda_2\mu_2}$. Далее от $0'_{\lambda_2\mu_2}$ движемся по μ_2-му столбцу до нуля со звездочкой и т. д. Последовательный переход

---

[1] Такой элемент $c_{i\mu}$ называется существенным нулем.

от нуля со штрихом к нулю со звездочкой по столбцу и от нуля со звездочкой к нулю со штрихом по строке осуществляют до тех пор, пока это возможно. Можно доказать, что построение цепочки производится всегда однозначно, цепочка не имеет замкнутых циклов и заканчивается нулем со штрихом.

После того как цепочка вида

$$0'_{\lambda_1\mu_1} - 0^*_{\lambda_2\mu_1} - 0'_{\lambda_2\mu_2} - \cdots - 0'_{\lambda_s\mu_s}$$

построена, осуществляют переход к матрице $\mathbf{X}_{k+1}$ от матрицы $\mathbf{X}_k$ по формуле

$$x_{ij}^{k+1} = \begin{cases} x_{ij}^{(k)}, & \text{если } c_{ij}^{(k)} \text{ не входит в цепочку;} \\ x_{ij}^{(k)} + \theta_k, & \text{если } c_{ij}^{(k)} - \text{нечетный элемент цепочки;} \\ x_{ij}^{(k)} - \theta_k, & \text{если } c_{ij}^{(k)} - \text{четный элемент цепочки;} \end{cases} \tag{3.7}$$

где

$$\begin{gathered} \theta_k = \min_{1 \leqslant t \leqslant s} \{x^{(k)}_{\lambda_t \mu_{t-1}};\quad \delta^{(k)}_{\lambda_1};\quad \delta^{(k)}_{\mu_s}\}; \\ \delta^{(k)}_{\lambda} = a_{\lambda_1} - \sum_{j=1}^{n} x^{(k)}_{\lambda_1 j};\quad \delta^{(k)}_{\mu_s} = b_{\mu_s} - \sum_{i=1}^{m} x_{i\mu_s}{}^{(k)}. \end{gathered} \tag{3.8}$$

Таким образом, θ_k — *минимальный элемент среди совокупности четных элементов цепочки, невязки строки, где начинается цепочка, и столбца, где она заканчивается.*

После выполнения второго этапа $(k + 1)$-ю итерацию заканчивают, причем невязка $\Delta_{k+1} = \Delta_k - 2\theta_k$.

Третий этап. Итак, допустим, что все нули выделены. Этот этап заключается в переходе от матрицы $\mathbf{C}_k$ к матрице $\mathbf{C}'_k$, в которой появляется один (или более) невыделенный нуль.

Пусть $h = \min \{c_{ij}^{k}\}$, где минимум выбирают из всех невыделенных элементов матрицы $\mathbf{C}_k$. Тогда из элементов матрицы $\mathbf{C}_k$, принадлежащих к невыделенным строкам, вычитают h, а к элементам выделенных столбцов прибавляют h. Полученную при этом матрицу обозначают $\mathbf{C}_k^{(1)}$.

Далее переходят к первому этапу, а затем либо переходят ко второму, либо снова возвращаются к третьему этапу.

Если после второго этапа суммарная невязка Δ_{k+1} равна нулю, то $\mathbf{X}_{k+1}$ оптимальный план.

В противном случае переходят к следующей $(k + 2)$-й итерации.

Отметим некоторые достоинства венгерского метода.

1. Поскольку данный метод в отличие от метода потенциалов не использует опорных планов, то явление вырожденности для него отсутствует. Это устраняет возможность зацикливания, связанного с вырожденностью планов, и облегчает машинную реализацию метода на ЭВМ.

2. На каждой итерации метод позволяет по величине невязке Δ_k судить о близости $\mathbf{X}_k$ к оптимальному плану, а также оценить верхнюю

границу числа оставшихся итераций $N_{\text{ост}}$:

$$N_{\text{ост}} \leqslant \frac{\Delta_k}{2}. \tag{3.9}$$

Эта формула справедлива для целочисленных значений всех переменных

$$a_i, \ i = \overline{1, m}, \ b_j = \overline{1, n}.$$

Обоснование венгерского метода

Прежде всего докажем справедливость признака оптимальности, т. е. если $\Delta_k = 0$, то план $\mathbf{X}_k$ — оптимален.

Действительно, в силу построения $\mathbf{X}_k$, если $x_{ij}^{(k)} > 0$, то $c_{ij}^{(k)} = 0$ (эти нули $\mathbf{C}_k$ называют существенными). Поэтому план $\mathbf{X}_k$ оказывается оптимальным для задачи с матрицей $\mathbf{C}_k$, так как

$$L_{C_k}(\mathbf{X}_k) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^{(k)} x_{ij}^{(k)} = 0. \tag{3.10}$$

Но матрица $\mathbf{C}_k$ получена эквивалентными преобразованиями из исходной матрицы $\mathbf{C}$. Докажем, что $\mathbf{X}_k$ оптимален и для задачи с матрицей $\mathbf{C}$. Матрицы $\mathbf{C}$ и $\mathbf{C}_k$, как эквивалентные, связаны соотношениями

$$c_{ij}^{(k)} = c_{ij} - \alpha_i - \beta_j \text{ для всех } i, j. \tag{3.11}$$

Тогда значение целевой функции для плана $\mathbf{X}_k$ при матрице $\mathbf{C}$ будет равно

$$L_c(\mathbf{X}_k) = \sum_i \sum_j c_{ij} x_{ij}^{(k)} = \sum_i \sum_j (c_{ij}^{(k)} + \alpha_i + \beta_j) x_{ij}^{(k)} =$$

$$= \sum_i \sum_j c_{ij}^{(k)} x_{ij}^{(k)} + \sum_i \alpha_i \sum_j x_{ij}^{(k)} + \sum_j \beta_j \sum_i x_{ij}^{(k)}. \tag{3.12}$$

Но так как $\Delta_k = 0$, то $\sum_j x_{ij}^{(k)} = a_i$ и $\sum_i x_{ij}^{(k)} = b_j$, $i = \overline{1, m}$, $j = \overline{1, n}$.

(3.13)

Подставляя (3.13) в (3.12), получим с учетом (3.10):

$$L_c(\mathbf{X}_k) = L_{c_k}(\mathbf{X}_k) + \sum_i \alpha_i a_i + \sum_j \beta_j b_j = \sum_i \alpha_i a_i + \sum_j \beta_j b_j. \tag{3.14}$$

Но $\sum_i \alpha_i a_i + \sum_j \beta_j b_j = \text{const}$ и не зависит от плана $\mathbf{X}_k$, поэтому план $\mathbf{X}_k$ оптимален и для исходной задачи с матрицей $\mathbf{C}$.

Перейдем к обоснованию алгоритма венгерского метода.

Предварительный этап. На предварительном этапе строят матрицу $\mathbf{X}_0$, элементы $x_{ij}^{(0)}$ которой удовлетворяют условиям

$$\sum_{j=1}^{n} x_{ij}^{(0)} \leqslant a_i \tag{3.15}$$

$$\sum_{i=1}^{m} x_{ij}^{0} \leqslant b_j \tag{3.16}$$

Если все условия (3.15) и (3.16) выполняются как строгие неравенства, то план X_0 — оптимален, согласно только что доказанному.

Первый этап. Цель первого этапа состоит в отыскании такого невыделенного нуля $c^{(k)}_{\lambda_1\mu_1} = 0$, для которого невязка по строке $\delta^{(k)}_{\lambda_1} > 0$. Предположим, что такой нуль найден и мы перешли ко второму этапу.

Второй этап. Он состоит в построении цепочки из нулей со штрихами и звездочками и переходе к X_{k+1}.

Пусть цепочка имеет вид:

$$0'_{\lambda_1\mu_1} — 0^*_{\lambda_2\mu_1} — 0'_{\lambda_2\mu_2} — \dots — 0^*_{\lambda_s\mu_{s-1}} — 0'_{\lambda_s\mu_s}.$$

Элементы матрицы X_{k+1} вычисляют по рекуррентным формулам

$$x^{(k+1)}_{ij} = \begin{cases} x^{(k)}_{ij} + \theta_k, & \text{если } c^{(k)}_{ij} \text{ — нечетный элемент цепочки,} \\ x^{(k)}_{ij} - \theta_k, & \text{если } c^{(k)}_{ij} \text{ — четный элемент цепочки,} \\ x^{(k)}_{ij}, & \text{в остальных случаях,} \end{cases} \tag{3.17}$$

$$\text{где } \theta_k = \min_{1 \leqslant t \leqslant s} \{\delta_{\lambda_1}, \delta_{\mu_s}, x_{\lambda_t\mu_{t-1}}\}. \tag{3.18}$$

Так как в каждом столбце и строке имеется как элемент $0'$, так и 0^*, либо они оба отсутствуют, за исключением строки λ_1 и столбца μ_s, где имеется лишь $0'$, то

$$\sum_{j=1}^{n} x^{(k+1)}_{ij} = \begin{cases} \sum_j x^{(k)}_{ij}, & \text{если } i \neq \lambda_1, \\ \sum_j x^{(k)}_{ij} + \theta_k, & \text{если } i = \lambda_1, \end{cases} \tag{3.19}$$

$$\sum_i x^{(k+1)}_{ij} = \begin{cases} \sum_i x^{(k)}_{ij}, & \text{если } j \neq \mu_s, \\ \sum_i x^{(k)}_{ij} + \theta_k, & \text{если } j = \mu_s. \end{cases} \tag{3.20}$$

Поэтому, если матрица X_k удовлетворяла ограничениям (3.15) и (3.16), то и X_{k+1} будет им также удовлетворять.

Наконец, на основании соотношений (3.19), получим

$$\Delta_{k+1} = \sum_i a_i + \sum_j b_j - 2\sum_i \sum_j x^{(k+1)}_{ij} = \sum_i a_i + \sum_j b_j -$$
$$-2\left(\sum_i \sum_j x^{(k)}_{ij} + \theta_k\right) = \Delta_k - 2\theta_k.$$

Третий этап. В соответствии с правилами перехода от C_k к C'_k и при выборе элемента h невыделенные элементы C_k уменьшаются на величину h, появляются новые нули и можно снова перейти к первому этапу. При этом по правилу выделения строк и столбцов все существенные нули C_k останутся нулевыми элементами и в матрице C'_k,

Пример 3.5. Найти решение транспортной задачи со следующими условиями:

$$C = \left| \begin{array}{cccc} 7 & 8 & 5 & 3 \\ 2 & 4 & 5 & 9 \\ 6 & 3 & 1 & 2 \end{array} \right| \begin{array}{lll} a_1 = 11, & a_2 = 11, & a_3 = 8; \\ b_1 = 5, & b_2 = 9, & b_3 = 9; \\ b_4 = 7. & & \end{array}$$

Проверим условие баланса Т-задачи: $\sum\limits_i a_i = \sum\limits_j b_j = 30.$

Предварительный этап. Вычитаем из первого столбца 2, из второго — 3, из третьего — 1, из четвертого — 2. Приходим к матрице C_1. Далее, вычитая минимальные элементы в каждой строке, получим матрицу C_0.

$$C_1 = \left\| \begin{array}{cccc} 5 & 5 & 4 & 1 \\ 0 & 1 & 4 & 7 \\ 4 & 0 & 0 & 0 \end{array} \right\| \to C_0 = \left\| \begin{array}{cccc} 4 & 4 & 3 & 0 \\ 0 & 1 & 4 & 7 \\ 4 & 0 & 0 & 0 \end{array} \right\|.$$

Строим начальную матрицу перевозок:

$$\begin{array}{rcc} & & a_i \quad \delta_i \\ X_0 = & \left\| \begin{array}{cccc} 0 & 0 & 0 & 7 \\ 5 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \end{array} \right\| & \begin{array}{cc} 11 & 4 \\ 11 & 6 \\ 8 & 0 \end{array} \\ b_j & \begin{array}{cccc} 5 & 9 & 9 & 7 \end{array} & \\ \delta_j & \begin{array}{cccc} 0 & 1 & 9 & 0 \end{array} & \end{array}$$

Невязки для столбцов $\delta_j = \{0, 1, 9, 0\}$,
для строк $\delta_i = \{4, 6, 0\}$.

Суммарная невязка $\Delta_0 = \sum\limits_i a_i + \sum\limits_j b_i - 2\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} x_{ij} = \sum\limits_{i=1}^{m}\delta_i + \sum\limits_{j=1}^{n}\delta_j = 20 > 0.$

Первая итерация. Первый этап. Отмечаем знаком «+» сверху первый и четвертый столбцы, которым соответствуют нулевые невязки, а знаком «×» слева первую и вторую строки, которым отвечают ненулевые невязки.

$$C_0 = \begin{array}{c} \\ \times \\ \times \\ \\ \end{array} \left\| \begin{array}{cccc} + & & & + \\ 4 & 4 & 3 & \bar{0} \\ \bar{0} & 1 & 4 & 7 \\ 4 & \bar{0}' & 0 & 0 \end{array} \right\| \begin{array}{c} \\ \\ \\ + \end{array}.$$

Выбираем нулевой элемент c_{32} матрицы C, расположенный в невыделенном столбце. Так как $\delta_3 = 0$, то выделяем третью строку знаком «+». Поскольку в матрице C_0 больше не осталось невыделенных нулей и все они расположены либо в выделенных столбцах, либо в выделенных строках, то переходим к третьему этапу.

Третий этап. Среди невыделенных строк и столбцов матрицы C_0 находим минимальный элемент $h = 1$ и прибавляем его ко всем выделенным столбцам и вычитаем из всех невыделенных строк. Получим матрицу C_1.

$$C_0 = \begin{array}{c} \\ \times \\ \times \\ \\ \end{array} \left\| \begin{array}{cccc} + & & & + \\ 4 & 4 & 3 & 0 \\ \bar{0} & 1 & 4 & 6 \\ 4 & \bar{0} & 0 & 0 \end{array} \right\| \begin{array}{c} \\ \\ \\ + \end{array} \xrightarrow{h=1} C_1 = \begin{array}{c} \\ \times \\ \times \\ \\ \end{array} \left\| \begin{array}{cccc} + & & & + \\ 5 & 3 & 2 & 0 \\ 0 & 0' & 3 & 6 \\ 5 & 0 & 0 & 0 \end{array} \right\|$$

Переходим к первому этапу.

Первый этап. Среди невыделенных столбцов находим нулевой элемент c_{22}, который расположен в строке с ненулевой невязкой, а потому переходим ко второму этапу.

Второй этап. Определим $\theta_1 = \min\{1, 6\} = 1$ и прибавим $\theta_1 = 1$ к элементу x_{22}. Получим

$$\mathbf{X}_1 = \begin{Vmatrix} 0 & 0 & 0 & 7 \\ 5 & 1 & 0 & 0 \\ 0 & 8 & 0 & 0 \end{Vmatrix} \begin{matrix} 4 \\ 5 \\ 0. \end{matrix}$$

$$\delta_j = 0 \quad 0 \quad 9 \quad 0$$

Вторая итерация. Первый этап. В матрице $\mathbf{C}_1$ отмечаем знаком «+» первый, второй и четвертый столбцы, которым отвечают нулевые невязки. Находим в третьем столбце элемент c_{33} и отмечаем его штрихом.

Так как невязка в третьей строке равна нулю, выделяем ее знаком «+». Просматриваем эту строку и находим в ней существенный нуль c_{32}, расположенный в выделенном столбце. Отмечаем его звездочкой и уничтожаем знак выделения второго столбца.

Далее просматриваем второй столбец и отыскиваем в нем невыделенный нуль c_{22}. Так как невязка по строке $\delta_2 > 0$, то, отметив этот нуль штрихом, переходим ко второму этапу.

Второй этап. Строим цепочку в матрице $\mathbf{C}_1$ вида $0'_{22} — 0^*_{32} — 0'_{33}$, а затем аналогичную цепочку в матрице $\mathbf{X}_1$. В результате получаем матрицу $\mathbf{X}_2$.

$$C_1 = \begin{Vmatrix} \overset{+}{4} & \overset{\oplus}{3} & 2 & \overset{+}{0} \\ \bar{0} & \bar{0}' & 3 & 7 \\ 5 & \bar{0}^* & 0' & 1 \end{Vmatrix}_{+} \quad \theta_2 = 5 \quad X_2 = \begin{Vmatrix} 0 & 0 & 0 & 7 \\ 5 & 1+5 & 0 & 0 \\ 0 & 8-5 & 0+5 & 0 \end{Vmatrix} = \begin{array}{c|cccc|c} & & & & & \delta_i \\ & 0 & 0 & 0 & 7 & 4 \\ & 5 & 6 & 0 & 0 & 0 \\ & 0 & 3 & 5 & 0 & 0 \\ \hline \delta_j & 0 & 0 & 4 & 0 & \end{array}$$

$$\theta_2 = \min\{5, 8, 9\} = 5.$$

Третья итерация. Первый этап. В матрице $\mathbf{C}_2$ отмечаем знаком «+» первый, второй и четвертый столбцы, которым соответствуют нулевые невязки. Находим нулевой элемент c_{33} в третьем столбце. Так как ему соответствует нулевая невязка в третьей строке, то отмечаем этот нуль штрихом. Далее просматриваем нулевые элементы, отмеченные сверху чертой в третьей строке.
Выделяем звездочкой элемент $c_{32} = 0$ и уничтожаем знак «+» во втором столбце. Просматриваем второй столбец, находим в нем нулевой элемент $c_{22} = 0$ и отмечаем вторую строку знаком «+», а сам элемент — штрихом.

Далее, просмотрев вторую строку, находим нулевой элемент $c_{21} = 0$ в первом столбце, выделяем его звездочкой и потому уничтожаем знак «+» над первым столбцом.

На этом процесс выделения нулей заканчиваем. Так как больше невыделенных нулей не имеется, то переходим к третьему этапу.

Третий этап. Находим минимальный элемент $h = 2$ в невыделенной части матрицы $\mathbf{C}_2$, вычитаем h из невыделенных строк и прибавляем к выделенным столбцам (т. е. прибавляем к третьему столбцу и вычитаем из первой строки). В результате увеличим число невыделенных нулей третьего столбца на единицу (элемент $c_{13} = 0$).

$$h = \min\{4, 3, 2\} = 2$$

$$C_2 = \begin{Vmatrix} \overset{\oplus}{4} & \overset{\oplus}{3} & 2 & \overset{\pm}{0} \\ \bar{0}^* & \bar{0}' & 3 & 7 \\ 5 & \bar{0}^* & 0' & 1 \end{Vmatrix}_{+} \xrightarrow[+]{h=2} C_3 = \begin{Vmatrix} \overset{+}{2} & \overset{+}{1} & 0' & \overset{\pm}{0} \\ \bar{0} & \bar{0} & 3 & 9 \\ 5 & \bar{0} & \bar{0} & 3 \end{Vmatrix}$$

Далее переходим к первому этапу.

Первый этап. Выделяем в матрице C_3 элемент c_{13}, который расположен в невыделенном третьем столбце и первой строке. Так как $\delta_1 > 0$, то переходим ко второму этапу.

Второй этап. Вся цепочка состоит из одного элемента $x_{13}^{(2)}$. Поэтому $\theta_3 = \min\{\delta_1 = 4, \delta_2 = 4\} = 4$. Прибавив 4 к $x_{13}^{(2)}$, приходим к матрице X_3.

$$\begin{array}{r|cccc|c} & & & & & \delta_i\\ X_2= & 0&0&0'&7&4\\ &5&6&0&0&0\\ &0&3&5&0&0\\ \hline \delta_j&0&0&4&0& \end{array} \xrightarrow{\theta_3=4} \begin{array}{r|cccc|c} & & & & & \delta_i\\ X_3= & 0&0&4&7&0\\ &5&6&0&0&0\\ &0&3&5&0&0\\ \hline \delta_j&0&0&0&0& \end{array}$$

Так как все невязки стали равны нулю, то X_3 — оптимальный план перевозок.

Целевая функция $L_{\text{opt}} = \sum\limits_i \sum\limits_j c_{ij}x_{ij}^{(3)} = 89$. (Сравни с результатами решения этой задачи методом потенциалов).

Транспортная задача с ограниченными пропускными способностями

Рассмотрим T_d-задачу, состоящую в минимизации линейной формы:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij}$$

при условиях

$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, \ldots, m;$$

$$\sum_{i=1}^{m} x_{ij} = b_j; \qquad j = 1, \ldots, n; \quad 0 \leqslant x_{ij} \leqslant d_{ij}.$$

Введем следующие определения:

1. *Элемент $c_{ij} = 0$ матрицы C называется X-неполным нулем, если в плане X решаемой T_d-задачи перевозка x_{ij} меньше величины d_{ij}.* Если же $x_{ij} = d_{ij}$, то элемент $c_{ij} = 0$ называется X-полным нулем.

2. *X-существенным нулем матрицы C называется такой элемент $c_{ij} = 0$, для которого $x_{ij} > 0$*, а в противном случае этот элемент называется несущественным нулем.

Описание алгоритма венгерского метода. Алгоритм решения T_d-задачи, основанный на венгерском методе, состоит из предварительного этапа и ряда однотипных итераций.

Предварительный этап. В каждом столбце матрицы C разыскивают минимальный элемент и вычитают его из всех элементов данного столбца. В результате получают матрицу C'. Далее из всех элементов каждой строки матрицы C' вычитают минимальный элемент этой строки и получают матрицу $C_0 = \| c_{ij}^{(0)} \|$, в каждой строке и столбце которой имеется нуль.

После этого формируют матрицу $\mathbf{X}_0 = \| x_{ij}^{(0)} \|$, процесс построения которой ведут по столбцам. Пусть уже заполнены первые $(j-1)$ столбцы матрицы $\mathbf{X}_0$. Перенумеруем нули j-го столбца матрицы $\mathbf{C}_0$ сверху вниз и через i_{k_j} $(k = 1, 2, \ldots, r_j)$ обозначим номер строки, содержащей k-й нуль j-го столбца. Тогда элементы j-го столбца определяются в соответствии с формулой

$$x_{ij}^{(0)} = \begin{cases} 0, \text{ если } i \neq i_{k_j}; \\ \min\left\{a_i - \sum\limits_{\mu=1}^{j-1} x_{i\mu}^{(0)};\ b_j - \sum\limits_{\mu=1}^{i-1} x_{\mu j}^{(0)};\ d_{ij}\right\} \text{ если } i = i_{k_j}, \end{cases}$$
$$k = 1, 2, \ldots, r_j. \qquad (3.21)$$

Если для матрицы $\mathbf{X}_0$ суммарная невязка

$$\Delta_0 = \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^{(0)} = 0,$$

то $\mathbf{X}_0$ — оптимальный план T_d-задачи.

Если же $\Delta_0 > 0$, то переходят к первой итерации.

Каждая итерация алгоритма в общем случае включает три этапа: начинается первым этапом, затем несколько раз могут повторяться первый и третий этапы, а заканчивается итерация вторым этапом, либо установлением неразрешимости данной задачи.

$(k+1)$-*я итерация*. Предположим, что уже осуществлено k итераций алгоритма, в результате которых получены матрицы $\mathbf{X}_k$ и $\mathbf{C}_k$. Пусть $\Delta_k = \sum\limits_{i=1}^{m} a_i + \sum\limits_{j=1}^{n} b_j - 2\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} x_{ij}^{(k)} > 0$ и еще не установлена неразрешимость T_d-задачи.

Целью $(k+1)$-й итерации является построение матрицы $\mathbf{X}_{k+1}$ и проверка ее на оптимальность или на установление неразрешимости T_d-задачи. Перед началом итерации знаком «+» выделяют те столбцы матрицы $\mathbf{C}_k$, для которых невязки

$$\delta_j = b_j - \sum_{i=1}^{m} x_{ij}^{(k)} = 0.$$

Первый этап. Выбирают произвольный невыделенный $\mathbf{X}_k$-неполный нуль матрицы $\mathbf{C}_k$. Если это элемент $c_{ij}^{(k)}$, то вычисляют невязку строки, его содержащей

$$\delta_i^{(k)} = a_i - \sum_{\mu=1}^{n} x_{i\mu}.$$

Тогда возможен один из двух случаев:

$$\delta_i^{(k)} = 0; \quad \delta_i^{(k)} > 0.$$

В первом случае знаком «+» выделяют i-ю строку матрицы $\mathbf{C}_k$, а элемент $c_{ij}^{(k)}$ отмечают штрихом. Если на пересечении μ-го выделенного столбца и i-й строки матрицы $\mathbf{C}_k$ расположен X_k-существенный нуль матрицы $\mathbf{C}_k$, то знак выделения этого столбца уничтожают, а элемент

$c_{i\mu}^{(k)} = 0$ отмечают звездочкой. Далее просматривают столбец μ, отыскивают в нем невыделенный X_k-неполный нуль (нули), который отмечают штрихом. За конечное число шагов процесс выделения X_k-неполных нулей матрицы $\mathbf{C}_k$ заканчивается одним из следующих трех исходов:

1. Найден X_k-неполный нуль в строке i, где $\delta_i^{(k)} > 0$, тогда переходим ко второму этапу, отметив этот нуль штрихом.

2. Все X_k-неполные нули выделены (для каждого из них $\delta_i^{(k)} = 0$), а среди невыделенных элементов матрицы $\mathbf{C}_k$ имеются либо положительные, либо среди дважды выделенных элементов $\mathbf{C}_k$ (т. е. элементы, расположенные на пересечении выделенных строк и столбцов) отрицательные элементы. В этом случае переходим к третьему этапу.

3. Все X_k-неполные нули выделены (для каждого из них $\delta_i^{(k)} = 0$), все невыделенные элементы $\mathbf{C}_k$ — отрицательны, а дважды выделенные — положительны. Это означает неразрешимость T_d-задачи.

Во втором случае, отметив невыделенный неполный нуль $(c_{ij}^{(k)} = 0)$ знаком штрих, сразу переходим ко второму этапу.

Второй этап состоит в построении цепочки из нулей матрицы $\mathbf{C}_k$, отмеченных штрихами и звездочками. С помощью этой цепочки осуществляют переход от $\mathbf{X}_k$ к $\mathbf{X}_{k+1}$. Итак, пусть первый этап завершился таким образом (первый исход), что для некоторого невыделенного X_k-неполного нуля, расположенного, например, на пересечении строки λ_1 и столбца μ_1, невязка $\delta_{\lambda_1}^{(k)} > 0$. Этот элемент принимают за начало цепочки из отмеченных нулей матрицы $\mathbf{C}_k$. Цепочку строят так. В столбце μ_1 матрицы $\mathbf{C}_k$ выбирают $\overset{*}{0}_{\lambda_2\mu_1}$, а в строке λ_2 выбирают $0'_{\lambda_2\mu_2}$ и т. д. Процесс построения цепочки, складывающийся из последовательных переходов от нуля со штрихом к нулю со звездочкой по столбцу и от нуля со звездочкой к нулю со штрихом по строке, всегда начинается и заканчивается на нуле со штрихом.

Пусть в результате образована цепочка вида

$$0'_{\lambda_1\mu_1} - \overset{*}{0}_{\lambda_2\mu_1} - 0'_{\lambda_2\mu_2} - \cdots - \overset{*}{0}_{\lambda_s\mu_{s-1}} - 0'_{\lambda_s\mu_s}. \tag{3.22}$$

Элементы $x_{ij}^{(k+1)}$ матрицы $\mathbf{X}_{k+1}$ вычисляют по рекуррентной формуле

$$x_{ij}^{(k+1)} = \begin{cases} x_{ij}^{(k)}, & \text{если } c_{ij}^{(k)} \text{ не входит в цепочку;} \\ x_{ij}^{(k)} + \theta_k, & \text{если } c_{ij}^{(k)} \text{ — нечетный элемент цепочки;} \\ x_{ij} - \theta_k, & \text{если } c_{ij}^{(k)} \text{ — четный элемент цепочки.} \end{cases} \tag{3.23}$$

Параметр θ_k определяют из соотношения

$$\theta_k = \min\{\theta'_k,\ \theta''_k;\ \delta_{\lambda_1}^{(k)},\ \delta_{\mu_s}^{(k)}\}, \tag{3.24}$$

где $\theta'_k = \min\limits_{1 \leqslant t \leqslant s} \{x_{\lambda_t\mu_{t-1}}^{(k)}\}$ — минимальный четный элемент цепочки; $\theta''_k = \min\limits_{1 \leqslant t \leqslant s} \{d_{\lambda_t\mu_t} - x_{\lambda_t\mu_t}^{(k)}\}$ — минимальный резерв до насыщения для нечетных элементов цепочки; $\delta_{\lambda_1}^{(k)} = a_{\lambda_1} - \sum\limits_{j=1}^{n} x_{\lambda_1 j}^{(k)}$ — невязка строки, откуда начинается цепочка; $\delta_{\mu_s}^{(k)} = b_{\mu_s} - \sum\limits_{i=1}^{m} x_{i\mu_s}^{(k)}$ — невязка столбца, где цепочка заканчивается.

Если суммарная невязка $\Delta_{k+1} = \sum\limits_{i=1}^{m} a_i + \sum\limits_{j=1}^{n} b_j - 2\sum\limits_{i}\sum\limits_{j} x_{ij}^{(k+1)} = 0$, то матрица $\mathbf{X}_{k+1}$ является решением T_d-задачи.

Если же $\Delta_{k+1} > 0$, то переходят к следующей итерации.

Третий этап. На этом этапе производят преобразование матрицы $\mathbf{C}_k$ в эквивалентную матрицу $\mathbf{C}_k'$ ($\mathbf{C}_k \sim \mathbf{C}_k'$).

Пусть первый этап закончился вторым исходом. Обозначим минимальный элемент

$$h = \min\{h', h''\}, \tag{3.25}$$

где h' — минимальный среди невыделенных положительных элементов матрицы $\mathbf{C}_k$; h'' — минимальный среди дважды выделенных отрицательных элементов матрицы $\mathbf{C}_k$, взятых с обратным знаком.

Вычитаем h из элементов матрицы $\mathbf{C}_k$, расположенных в невыделенных строках, и прибавляем его к элементам $\mathbf{C}_k$, расположенным в выделенных столбцах. Получают матрицу $\mathbf{C}_k'$. Если дважды выделенный отрицательный элемент $\mathbf{C}_k$ становится нулем $\mathbf{C}_k'$, то знак выделения над столбцом уничтожают, а сам элемент отмечают звездочкой. Остальные знаки выделения, а также все отметки нулей переносят из матрицы $\mathbf{C}_k$ в матрицу $\mathbf{C}_k'$.

Далее снова переходят к первому этапу, заменив $\mathbf{C}_k$ на $\mathbf{C}_k'$. Если первый этап снова завершится вторым исходом, то опять возвращаются к третьему этапу. Циклы, состоящие из первого и третьего этапов, проводят до тех пор, пока последний из них не закончится первым или третьим исходом. При первом исходе переходят ко второму этапу, которым заканчивают итерацию, а при третьем исходе делают вывод о неразрешимости T_d-задачи из-за несовместимости ее условий.

Пример 3.6. Решить венгерским методом T_d-задачу со следующими условиями:

$$\mathbf{C} = \begin{Vmatrix} 1 & 4 & 2 & 5 \\ 2 & 1 & 4 & 1 \\ 3 & 2 & 1 & 3 \end{Vmatrix}; \quad \begin{array}{lccccccc} & a_1 & a_2 & a_3 & b_1 & b_2 & b_3 & b_4 \\ \mathbf{P}^T = [& 6; & 3; & 3; & 4; & 2; & 4; & 2]; \end{array}$$

$$\mathbf{D} = \begin{Vmatrix} 3 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 1 & 1 & 1 \end{Vmatrix} \text{— матрица пропускных способностей коммуникаций.}$$

Предварительный этап. Составляем матрицу $\mathbf{C}_0$:

$$\mathbf{C} = \begin{Vmatrix} 1 & 4 & 2 & 5 \\ 2 & 1 & 4 & 1 \\ 3 & 2 & 1 & 3 \end{Vmatrix} \to \mathbf{C}' = \begin{Vmatrix} 0 & 3 & 1 & 4 \\ 1 & 0 & 3 & 0 \\ 2 & 1 & 0 & 2 \end{Vmatrix} = \mathbf{C}' = \mathbf{C}_0.$$

Затем строим матрицу $\mathbf{X}_0$ с учетом матрицы $\mathbf{D}$:

$$\begin{array}{c||cccc||cc}
 & & & & & a_i & \delta_i \\
\mathbf{X}_0 = & 3 & 0 & 0 & 0 & 6 & 3 \\
 & 0 & 1 & 0 & 2 & 3 & 0 \\
 & 0 & 0 & 1 & 0 & 3 & 2 \\
b_j & 4 & 2 & 4 & 2 & & \\
\delta_j & 1 & 1 & 3 & 0 & &
\end{array}$$

$$x_{11}^{(0)} = \min\{a_1,\ b_1,\ d_{11}\} = d_{11} = 3;$$
$$x_{22}^{(0)} = \min\{a_2,\ b_2,\ d_{22}\} = d_{22} = 1;$$
$$x_{33}^{(0)} = d_{33} = 1;$$
$$x_{24} = \min\{a_2 - x_{22}^{(0)},\ b_2,\ d_{24}\} = d_{24} = 2.$$

Будем отмечать одной точкой сверху $\mathbf{X}_k$-неполные нули матриц $\mathbf{C}_k$, а двумя точками — $\mathbf{X}_k$-полные нули этих матриц. Строки матрицы $\mathbf{C}_k$, которым отвечают ненулевые невязки, отмечаем знаком «×».

Первая итерация.

$$\mathbf{C}_0 = \begin{array}{c}\times\\ \\ \times\end{array}\left\|\begin{array}{cccc} & & & + \\ \ddot{0} & 3 & 1 & 4 \\ 1 & \ddot{0} & 3 & \ddot{0} \\ 2 & 1 & 0 & 2 \end{array}\right\|_{h=1} \to \mathbf{C}_0^{(1)} = \begin{array}{c}\times\\ \\ \times\end{array}\left\|\begin{array}{cccc} & & & + \\ -1 & 2 & \dot{0} & 4 \\ \dot{0} & -1 & 2 & \ddot{0} \\ 1 & \dot{0}' & -1 & 2 \end{array}\right\| = \mathbf{C}_1$$

$$\mathbf{X}_0 = \left\|\begin{array}{cccc} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0\uparrow & 1 & 0 \end{array}\right\|_{\theta_1 = 1} \to \begin{array}{c||cccc||c} & & & & & \bar{\delta}_i \\ \mathbf{X}_1 = & 3 & 0 & 0 & 0 & 3 \\ & 0 & 1 & 0 & 2 & 0 \\ & 0 & 1 & 1 & 0 & 1 \\ \bar{\delta}_j & 1 & 0 & 3 & 0 & \end{array}$$

$$\Delta_1 = 8.$$

Первый этап. Знаком «+» выделяем четвертый столбец. Так как матрица $\mathbf{C}_0$ не содержит ни одного $\mathbf{X}_0$-неполного невыделенного нуля, то сразу переходим к третьему этапу. Затем снова переходим к первому этапу, отыскиваем $\mathbf{X}_0$-неполный невыделенный нуль $c_{32}^{(1)}$ и отметив его штрихом, переходим ко второму этапу. Цепочка, построенная на втором этапе, состоит из одного элемента $x_{32}^{(0)}$. Поэтому $x_{32}^{(1)} = x_{32}^{(0)} + \theta_1 = 1$.

Вторая итерация

$$\mathbf{C}_1 = \begin{array}{c}\times\\ \\ \times\end{array}\left\|\begin{array}{cccc} & + & & + \\ -1 & 2 & \dot{0}\uparrow & 4 \\ \dot{0} & -1 & 2 & \ddot{0} \\ 1 & \ddot{0} & -1 & 2 \end{array}\right\| = \mathbf{C}_2; \quad \begin{array}{c||cccc||c} & & & & & \delta_i \\ \mathbf{X}_1 = & 3 & 0 & 0 & 0 & 3 \\ & 0 & 1 & 0 & 2 & 0 \\ & 0 & 1 & 1 & 0 & 1 \\ \hline \delta_j & 1 & 0 & 3 & 0 & \end{array} \to \mathbf{X}_2 =$$

$$= \begin{array}{c||cccc||c} & & & & & \delta_i \\ & 3 & 0 & 2 & 0 & 1 \\ & 0 & 1 & 0 & 2 & 0 \\ & 0 & 1 & 1 & 0 & 1 \\ \hline \delta_j & 1 & 0 & 1 & 0 & \end{array}$$

$$\theta_2 = 2 \quad \Delta_2 = 4.$$

Третья итерация

$$\mathbf{C}_2 = \begin{array}{c||cccc||l} & & + & & \oplus & \\ \times & -1 & 2 & \ddot{0} & 4 & \\ & \ddot{0}' & -1 & 2 & \dot{0}^* & \\ \times & 1 & \ddot{0} & -1 & 2 & h=1 \end{array} \quad + \rightarrow \mathbf{C}_3 = \begin{array}{c||cccc||} \times & -2 & 2 & -1 & 3 \\ & \dot{0} & \ddot{0} & 2 & 0 \\ \times & \dot{0}\uparrow & \ddot{0} & -2 & 1 \end{array}$$

$$\mathbf{X}_2 = \begin{array}{c||cccc||cl} & & & & & \delta_i & \\ & 3 & 0 & 2 & 0 & 1 & \\ & 0 & 1 & 0 & 2 & 0 & \\ & 0' & 1 & 1 & 0 & 1 & \theta_3 = 1 \\ \delta_j & 1 & 0 & 1 & 0 & & \end{array} \quad \rightarrow \mathbf{X}_3 = \begin{array}{c||cccc||c} & & & & & \delta_i \\ & 3 & 0 & 2 & 0 & 1 \\ & 0 & 1 & 0 & 2 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ \delta_j & 0 & 0 & 1 & 0 & \end{array}$$

$$\Delta_3 = 2.$$

Третья итерация состоит из первого и третьего этапов при $h = 1$, а также из первого и второго этапов, причем цепочка второго этапа содержит только один элемент $x_{31}^{(2)}$.

Четвертая итерация

Первый и третий этапы — Первый и третий этапы

$$C_3 = \begin{array}{c||cccc||l} & + & + & & + & \\ \times & -2 & 2 & -1 & 3 & \\ & \dot{0} & \ddot{0} & 2 & \ddot{0} & \\ \times & \dot{0} & \ddot{0} & -2 & 1 & h=2 \end{array} \longrightarrow C_3 = \begin{array}{c||cccc||l} & + & \oplus & & \oplus & \\ \times & -2 & 2 & -3 & 3 & \\ & \dot{0} & \ddot{0}^* & 0' & \ddot{0}^* & + \rightarrow \\ & \dot{0} & \ddot{0} & -4 & 1 & h=1 \end{array}$$

Первый и третий этапы — Первый этап

$$\overline{\overline{C}}{}_3^{(2)} = \begin{array}{c||cccc||l} & \oplus & \oplus & & \oplus & \\ \times & -2 & 1 & -4 & 2 & \\ & 1 & \ddot{0}^* & 0' & \ddot{0}^* & + \\ & 0^* & -1 & -5 & 0' & + \end{array} \xrightarrow{h=1} C_3^3 = \begin{array}{c||cccc||l} & + & \oplus & & + & \\ \times & -3 & 0' & -5 & 1 & \\ & 1 & \ddot{0}^* & \dot{0}' & \ddot{0} & + \\ & \dot{0} & -1 & -5 & 0 & \end{array}$$

Второй этап

$$X_3 = \begin{array}{c||cccc||c} & & & & & \delta_i \\ & 3 & 0 & 2 & 0 & 1 \\ & 0 & 1 & 0 & 2 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ \hline \delta_j & 0 & 0 & 1 & 0 & \theta_1 = 1 \end{array} \longrightarrow X_4 = \begin{array}{c||cccc||c} & & & & & \delta_i \\ & 3 & 1 & 2 & 0 & 0 \\ & 0 & 0 & 1 & 2 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ \hline \delta_j & 0 & 0 & 0 & 0 & \end{array}$$

Поскольку суммарная невязка $\Delta_4 = 0$, то $\mathbf{X}_4$ — оптимальный план. Целевая функция $L_{\min} = 3 \cdot 1 + 1 \cdot 4 + 2 \cdot 2 + 1 \cdot 4 + 2 \cdot 1 + 1 \cdot 3 + 1 \cdot 2 + 1 \cdot 1 = 25$.

Обоснование венгерского метода для Т-задачи

На предварительном этапе строят матрицу $\mathbf{X}_0$, элементы которой удовлетворяют условиям:

$$\sum_{j=1}^{n} x_{ij}^{(0)} \leqslant a_i, \qquad i = 1, 2, \ldots, m; \tag{3.26}$$

$$\sum_{i=1}^{m} x_{ij}^{(0)} \leqslant b_j, \qquad j = 1,\ 2,\ \ldots,\ n; \tag{3.27}$$

$$0 \leqslant x_{ij} \leqslant d_{ij}. \tag{3.28}$$

Первый этап здесь отличается от такого же этапа при решении Т-задачи только тем, что выделению штрихом подлежат не все нули матрицы $\mathbf{C}_k$, а только $\mathbf{X}_k$-неполные нули. Это изменение сделано для того, чтобы иметь возможность увеличивать элементы матрицы $\mathbf{X}_k$, отвечающие нулям матрицы $\mathbf{C}_k$, не нарушая условий (3.28).

Второй этап. В соответствии с формулами (3.23) элементы матрицы $\mathbf{X}_k$, отвечающие 0^*, уменьшаются на величину θ_k, а элементы матрицы $\mathbf{X}_k$, отвечающие $0'$, увеличиваются на величину θ_k, причем θ_k не превосходит по величине наименьшего из уменьшаемых элементов и минимального уклонения увеличиваемых элементов $x_{ij}^{(k)}$ от соответствующих значений d_{ij}.

Следовательно, если элементы матрицы $\mathbf{X}_k$ удовлетворяли ограничениям (3.28), то и элементы матрицы $\mathbf{X}_{k+1}$ также будут удовлетворять этим ограничениям.

Если цепочка начинается на элементе $0'_{\lambda_1\mu_1}$ и заканчивается на элементе $0'_{\lambda_s\mu_s}$, то справедливы следующие соотношения:

$$\sum_{j=1}^{n} x_{ij}^{k+1} = \begin{cases} \sum\limits_{j} x_{ij}^{(k)}, & \text{если } i \neq \lambda_1; \\ \sum\limits_{j} x_{ij}^{(k)} + \theta_k, & \text{если } i = \lambda_1; \end{cases}$$
$$\sum_{i=1}^{m} x_{ij}^{k+1} = \begin{cases} \sum\limits_{i} x_{ij}^{(k)}, & \text{если } j \neq \mu_s; \\ \sum\limits_{i} x_{ij}^{(k)} + \theta_k, & \text{если } j = \mu_s. \end{cases} \tag{3.29}$$

Учитывая неравенство

$$\theta_k \leqslant \min\{\delta_{\lambda_1},\ \delta_{\mu_s}\},$$

получают, что если матрица $\mathbf{X}_k$ удовлетворяет условиям (3.26), (3.27), то и матрица $\mathbf{X}_{k+1}$ также удовлетворяет этим условиям.

На основании соотношений (3.29)

$$\Delta_{k+1} = \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2\sum_{i}\sum_{j} x_{ij}^{(k+1)} = \Delta_k - 2\theta_k. \tag{3.30}$$

Третий этап. В соответствии с правилом перехода от $\mathbf{C}_k$ к $\mathbf{C}'_k$ дважды выделенные элементы матрицы $\mathbf{C}_k$ увеличиваются на $h > 0$; невыделенные элементы матрицы $\mathbf{C}_k$ уменьшаются на h, а все остальные элементы этой матрицы сохраняют прежние значения, причем $\mathbf{C}'_k \sim \mathbf{C}_k$.

Покажем, что

$$\begin{cases} \text{если } c_{ij}^{(k+1)} > 0, \text{ то } x_{ij}^{(k+1)} = 0; \\ \text{если } c_{ij}^{(k+1)} < 0, \text{ то } x_{ij}^{(k+1)} = d_{ij}. \end{cases} \tag{3.31}$$

При $k = 1$ утверждение (3.31) справедливо по правилам построения $\mathbf{X}_0$ на предварительном этапе.

Докажем это утверждение методом индукции. Предположим, что неравенства (3.31) справедливы для некоторого k и докажем, что они сохраняются и для $(k + 1)$.

Первая часть доказательства. Итак, пусть в матрице $\mathbf{C}_k^{(')}$ элемент $c_{ij}^{(k,1)} > 0$. Тогда, очевидно, элемент $c_{ij}^{(k)} \geqslant 0$. Если $c_{ij}^{(k)} > 0$, то по предположению $x_{ij}^{(k)} = 0$, а потому и $x_{ij}^{(k+1)} = 0$, так как c_{ij} не входит в цепочку.

Если же $c_{ij}^{(k)} = 0$, то $c_{ij}^{(k,1)} = c_{ij}^{(k)} + h$ и, следовательно, $c_{ij}^{(k)}$ находится на пересечении выделенной строки и столбца. Это возможно тогда и только тогда, когда $c_{ij}^{(k)}$ — несущественный нуль, т. е. $x_{ij}^{(k)} = 0$. Таким образом, из условия $c_{ij}^{(k,1)} > 0$ следует $x_{ij}^{(k+1)} = 0 = x_{ij}^{(k)}$.

Вторая часть доказательства. Пусть теперь элемент $c_{ij}^{(k,1)} < 0$. Тогда по правилу определения параметра h элемент $c_{ij}^{(k)} \leqslant 0$. Если $c_{ij}^{(k)} < 0$, то по методу индукции $x_{ij}^{(k)} = d_{ij}$. Если же $c_{ij}^{(k)} = 0$, то этот нуль является невыделенным, так как при переходе к $\mathbf{C}_k'$ он уменьшился. Следовательно, $c_{ij}^{(k)}$ — полный нуль $\mathbf{X}_k$, т. е. $x_{ij}^{(k)} = d_{ij}$.

Таким образом, поскольку $c_{ij}^{(k,1)} < 0$, то

$$x_{ij}^{(k)} = d_{ij} = x_{ij}^{(k+1)}.$$

Так как элементы $c_{ij}^{(k,1)} < 0$, то они не входят в цепочку, а потому не изменяются. Итак, утверждение (3.31) доказано.

Докажем теперь, что при третьем исходе первого этапа условия задачи несовместны.

Пусть E_1 — множество выделенных (E_2 — множество невыделенных) позиций матрицы $\mathbf{C}_{k+1}$. Если $\mathbf{X}$ — произвольная матрица, элементы которой удовлетворяют неравенствам (3.26), (3.27), то

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{(i,j)\in E_1} x_{ij} + \sum_{(i,j)\in E_2} x_{ij} \leqslant \sum_{\mu=1}^{t}\sum_{i=1}^{m} x_{ij_\mu} + \sum_{\lambda=1}^{r}\sum_{j=1}^{n} x_{i_\lambda j} + \\ + \sum_{(i,j)\in E_1} d_{ij} \leqslant \sum_{\mu=1}^{t} b_{j_\mu} + \sum_{\lambda=1}^{r} a_{i_\lambda} + \sum_{(i,j)\in E_2} d_{ij}, \qquad (3.32)$$

где $i_1, i_2, \ldots, i_r$ — номера выделенных строк в матрице $\mathbf{C}_{k+1}$; $j_1, j_2, \ldots, j_t$ — номера выделенных столбцов в матрице $\mathbf{C}_{k+1}$.

Если $c_{ij}^{(k+1)}$ — дважды выделенный элемент матрицы $\mathbf{C}_{k+1}$, то $c_{ij}^{(k+1)} \geqslant 0$ (условие третьего исхода). Покажем, что тогда $x_{ij}^{(k)} = 0$.

Для случая $c_{ij}^{(k+1)} > 0$ это уже доказано, а для случая $c_{ij}^{(k+1)} = 0$ выражение $x_{ij}^{(k)} = 0$ следует из правила выделения строк первого этапа.

Итак, дважды выделенному элементу матрицы $\mathbf{C}_{k+1}$ отвечает $x_{ij}^{(k)} = 0$.

Следовательно,

$$\sum_{(i,j)\in E_1} x_{ij}^{(k)} = \sum_{\mu=1}^{t}\sum_{i=1}^{m} x_{ij_\mu}^{(k)} + \sum_{\lambda=1}^{r}\sum_{j=1}^{n} x_{i_\lambda j} = \sum_{\mu=1}^{t} b_{j_\mu} + \sum_{\lambda=1}^{r} a_{i_\lambda}. \qquad (3.33)$$

Если $c_{ij}^{(k+1)}$ — невыделенный элемент матрицы $\mathbf{C}_{k+1}$, то $c_{ij}^{(k+1)} \leqslant 0$ (условие третьего исхода).

Методом индукции доказано, что при $c_{ij}^{(k+1)} < 0$ $x_{ij}^{(k+1)} = d_{ij}$.

Если $c_{ij}^{(k+1)} < 0$ и элемент $c_{ij}^{(k)}$ — не выделен, то это полный нуль, т. е. $x_{ij}^{(k)} = d_{ij}$.

Следовательно,

$$\sum_{(i,j)\in E_2} x_{ij}^{(k)} = \sum_{(i,j)\in E_2} d_{ij}. \qquad (3.34)$$

Соотношения (3.32) — (3.34) приводят к неравенству

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \leqslant \sum_{(i,j)\in E_1} x_{ij}^{(k)} + \sum_{(i,j)\in E_2} x_{ij}^{(k)} = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^{(k)}. \qquad (3.35)$$

По условию

$$\Delta_k = 2\left(\sum_{i=1}^{m} a_i - \sum_{i}\sum_{j} x_{ij}^{(k)}\right) > 0$$

и в соответствии с неравенством (3.35)

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} < \sum_{i} a_i, \qquad (3.36)$$

т. е. ни при каком плане $\mathbf{X}$ условия (3.36) не переходят в равенства. Здесь $\mathbf{X} = \| x_{ij} \|$ — любая матрица, удовлетворяющая условиям (3.26)—(3.28).

Неравенство (3.36) указывает на несовместность условий T_d-задачи, так как для любого плана $\mathbf{X}$ в выражении (3.36) должно быть строгое равенство.

§ 4. ТРАНСПОРТНЫЕ СЕТИ

Основные понятия и определения

Выше нами рассматривались транспортные задачи, в которых предполагалось, что имеются отдельно пункты производства и потребления, причем перевозки от пунктов потребления к пунктам производства невозможны. Графически такая задача представлена на рис. 3.7.

Между тем, в большинстве практических задач имеются пункты, которые одновременно являются пунктами ввоза и вывоза продукта. Пример такой задачи приведен на рис. 3.8.

Следовательно, возникает необходимость расширить введенные в § 1 основные понятия Т-задачи на такой, более общий случай.

Такое обобщение приводит к понятию транспортной сети (Т-сети).

Пусть P — произвольное конечное множество пунктов p_i, K — совокупность некоторых упорядоченных пар $k_{ij} = (p_i, p_j)$, где $p_i, p_j \in \in P$.

Набор двух множеств P и K будем называть транспортной сетью (P, K) и говорить, что она задается или порождается множествами P и K.

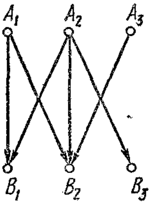


Рис. 3.7.

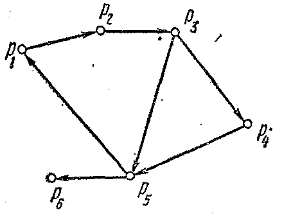


Рис. 3.8.

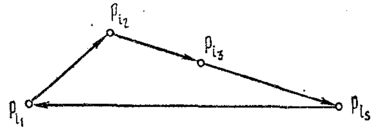


Рис. 3.9.

Элементы p_i называются пунктами Т-сети, а элементы $k_{ij} = (p_i, p_j)$ множества K — коммуникациями сети. k_{ij} связывает p_i и p_j, причем она начинается в p_i и заканчивается в p_j.

Произвольная Т-сеть может быть задана графически (рис. 3.8). Направление коммуникации отмечается стрелкой.

Т-сеть может задаваться с помощью матриц. Допустим, что сеть (P, K) содержит N пунктов. Рассмотрим квадратную матрицу $\mathrm{A}^{(N \times N)}$, элементы которой вычисляются по правилу:

$$a_{ij} = \begin{cases} 1, & \text{если } k_{ij} \in K, \\ 0, & \text{в противном случае.} \end{cases}$$

Матрица **А** однозначно определяет породившую ее сеть.

Рассмотрим последовательность коммуникаций

$$k_{i_1 j_1},\ k_{i_2 j_2},\ \ldots,\ k_{i_s j_s}$$

некоторой T-сети.

Если каждая пара коммуникаций $k_{i_1 j_1}$, $k_{i_2 j_2}$, стоящих рядом, имеет один общий пункт, а остальные пары коммуникаций не содержат общих пунктов, то такая последовательность коммуникаций называется маршрутом.

Если $p_{i_1} = p_{i_s}$, то маршрут является замкнутым (рис. 3.9). Если в любом пункте маршрута l, не являющемся крайним, одна коммуникация начинается, а другая заканчивается, то маршрут называется направленным.

Т-сеть называется *связной*, если любая пара ее пунктов может быть связана с помощью некоторого маршрута этой сети.

Функции на сети

Введем ряд функций, определенных на Т-сети.

1 Вводится функция *производства — потребления* $q\ (p_i)$, которая задается на множестве P всех пунктов сети (P, K). Если $q_i = q\ (p_i) > > 0$, то пункт p_i называется пунктом производства, если $q\ (p_i) < 0$,

то p_i — пункт потребления, если $q(p_i) = 0$, то p_i — перевалочный пункт.

2. Вводится функция *пропускных способностей* $d(k_{ij}) \geqslant 0$ для всех $k_{ij} \in K$; величина $d(k_{ij})$ называется пропускной способностью коммуникации k_{ij}. Если сеть (P, K) не содержит коммуникацию $k_{\lambda\mu}$, то можно условно считать, что $k_{\lambda\mu} \in K$, но $d_{\lambda\mu} = d(k_{\lambda\mu}) = 0$.

3. Рассмотрим сеть (P, K) на множестве коммуникаций которой определена функция $d(k_{ij})$. Выделим два пункта сети p_1 и p_N и назовем p_1 *источником*, а p_N — *стоком*.

Пусть E_i' — множество коммуникаций, исходящих из p_i, а E_i'' — множество коммуникаций, входящих в p_i. Введем следующее определение.

Функция $x_{ij} = x(k_{ij})$, для которой имеют место соотношения:

$$\sum_{k_{ij} \in E_i'} x(k_{ij}) - \sum_{k_{ji} \in E_i''} x(k_{ji}) = 0 \quad \text{для всех } i \neq 1,\ N, \tag{4.1}$$

$$x_1 = \sum_{k_{1j} \in E_1'} x(k_{1j}) - \sum_{k_{j1} \in E_1''} x(k_{j1}) \geqslant 0, \quad 0 \leqslant x(k_{ij}) \leqslant d(k_{ij}) \tag{4.2}$$

называется *потоком* на сети, направленным из p_1 в p_N. Сложив (4.1), (4.2) и равенство (4.3)

$$x_N = \sum_{k_{Nj} \in E_N'} x(k_{Nj}) - \sum_{k_{jN} \in E_i''} x(k_{jN}), \tag{4.3}$$

получим

$$x_1 + x_N = \sum_{i=1}^{N} \left(\sum_{k_{ij} \in E_i'} x(k_{ij}) - \sum_{k_{ji} \in E_i''} x(k_{ji}) \right). \tag{4.4}$$

Так как любая коммуникация исходит из одного пункта и заканчивается в другом, то все $x(k_{ij})$ дважды встречаются в правой части (4.4), один раз со знаком $+$, а другой раз со знаком $-$, и потому

$$x_1 + x_N = 0. \tag{4.5}$$

Величина $x_1 = -x_N$ называется *величиной потока* на сети.

Представим себе, что сеть (P, K) является сетью каналов с единственным источником p_1, стоком p_N, и канал k_{ij} способен пропускать не более d_{ij} единиц жидкости в единицу времени.

Тогда $x(k_{ij})$ — скорость протекания жидкости (газа) по коммуникации k_{ij}, x_1 — скорость притока, x_N — скорость отвода жидкости.

Условие $x_1 = -x_N$ представляет собой условие стационарности потока.

Введенная функция $x(k_{ij})$, $k_{ij} \in K$, очевидно, неотрицательна, и носит название *арифметического потока*.

При выяснении различных свойств потока иногда необходимо отказаться от свойства неотрицательности, для чего вводится понятие *алгебраического потока*.

Рассмотрим два пункта p_i и p_j, положим для них величину $x'(k_{ij})$ равной

$$x'(k_{ij}) = x(k_{ij}) - x(k_{ji}). \tag{4.6}$$

Если $k_{ij} \notin K$, $k_{ji} \in K$, то $x(k_{ij}) = 0$, $x(k_{ji}) > 0$ и $x'(k_{ij}) < 0$.

Расширим множество K, введя в него те коммуникации k_{ij}, для которых $k_{ij} \notin K$, для чего положим $d(k_{ij}) = 0$, обозначим его через K'.

Тогда
$$x'(k_{ij}) = -x'(k_{ji}) \tag{4.7}$$
и
$$x'(k_{ij}) \leqslant d(k_{ij}). \tag{4.8}$$

Учитывая (4.6), (4.7) и (4.8), условия 4.1, 4.2 и 4.4 можно записать в виде

$$\sum_{k_{ij}\in E_i} x'(k_{ij}) = \begin{cases} 0, & \text{если } i \neq 1,\ N, \\ x_1 \geqslant 0, & \text{при } i = 1, \\ x_N = -x_1, & \text{при } i = N, \end{cases} \tag{4.9}$$

причем $E_i = E_i' \cup E_i''$.

Функция $x'(k_{ij})$, определенная на множестве K' и удовлетворяющая (4.6)—(4.8) и (4.9), называется *алгебраическим потоком.*

Допустим, что на множестве P пунктов сети заданы $q(p_i)$, а на множестве K' определены $d(k_{ij})$.

Функция $x'(k_{ij})$, определенная на K', для которой справедливы соотношения

$$\sum_{k_{ij}\in E_i} x'(k_{ij}) = q(p_i), \tag{4.10}$$

$$x'(k_{ij}) \leqslant d(k_{ij}), \tag{4.11}$$

называется планом перевозок, совместимым с функциями $q(p_i)$ и $d(k_{ij})$.

Складывая соотношения (4.9) по i и учитывая, что каждая перевозка $x(k_{ij})$ встречается дважды в левой части, получим

$$\sum_{i=1}^{N}\left(\sum_{k_{ij}\in E_i'} x(k_{ij}) - \sum_{k_{ji}\in E_i''} x(k_{ji})) = \right)\sum_{i=1}^{N} q(p_i) = 0. \tag{4.12}$$

Равенство (4.12) представляет собой *условие баланса для сети* и означает, что суммарный объем производства всех пунктов сети равен суммарному потреблению этих же пунктов. (Сравните с аналогичным условием баланса для обычной Т-задачи).

Некоторые задачи на сетях

Задача о кратчайшем пути. Пусть задана Т-сеть, состоящая из $p_0, p_1, \ldots, p_{n+1}$ и коммуникаций $\{k_{ij}\}$. Обозначим длину коммуникации k_{ij} через $c(k_{ij}) = c_{ij}$. Если p_i и p_j не соединены, то полагаем $c_{ij} = \infty$. Из всех возможных путей, соединяющих p_0 и p_{n+1}, требуется найти путь наименьшей длины.

Аналитически задача записывается следующим образом.

Поставим в соответствие каждой паре пунктов p_i и p_j переменную

$$x_{ij} = \begin{cases} 1, & \text{если } k_{ij} \text{ используется в искомом пути,} \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда требуется найти такие числа $0 \leqslant x_{ij} \leqslant 1$, для которых линейная форма

$$\sum_{j=0}^{n+1} \sum_{i=0}^{n+1} c_{ij} x_{ij} = \min, \tag{4.13}$$

при условии

$$\sum_{j=0}^{n+1} x_{ij} - \sum_{j=0}^{n+1} x_{ji} = 0, \qquad i = 1, 2, \ldots, n; \tag{4.14}$$

$$\sum_{j=0}^{n+1} x_{0j} - \sum_{j=0}^{n+1} x_{j0} = 1; \tag{4.15}$$

$$\sum_{j=0}^{n+1} x_{n+1,j} - \sum_{j=0}^{n+1} x_{j,n+1} = -1, \tag{4.16}$$

где $0 \leqslant x_{ij} \leqslant 1 \qquad (i,\ j = 0,\ 1,\ \ldots,\ n+1)$.

Условие (4.14) означает, что для любого p_i, не являющегося начальным или конечным пунктом, число исходящих коммуникаций равно числу входящих. Поскольку все $c_{ij} > 0$, для минимизации (4.13) необходимо, чтобы из каждого p_i выходило не более одной коммуникации.

Условие (4.15) означает, что число коммуникаций, исходящих из p_0, на 1 больше числа входящих. С учетом условия минимизации это значит, что из p_0 исходит 1 коммуникация.

Аналогично (4.16) означает, что в конечный пункт p_{n+1} входит только одна коммуникация.

Алгоритм решения задачи о кратчайшем пути находит довольно широкое применение в системах СПУ при отыскании критического пути. Поскольку идея алгоритма основана на методе динамического программирования, решение данной задачи будет рассмотрено в главе 6 *при изложении основ динамического программирования.*

Задача о максимальном потоке. Пусть задана произвольная Т-сеть $(P,\ K)$ и функция пропускных способностей $d\ (k_{ij}) > 0$, определенная на множестве K. Перенумеруем пункты $\{p_i\}$ так, чтобы источник был обозначен через p_1, а сток — через p_N.

Рассмотрим всевозможные потоки, совместимые с $d\ (k_{ij})$. Задача состоит в отыскании такого потока $\mathbf{X}^* = \| x\ (k_{ij}) \|$, при котором $\overset{*}{x_1} = \max$. Эта задача носит название *задачи о максимальном потоке.*

Приведем дополнительно несколько простых, но полезных свойств потоков на сетях.

Обозначим

$$\sum_{k_{ij} \in E_i} x\ (k_{ij}) = x\ (p,\ P).$$

Откуда величина $x_1 = x\ (p_1,\ P) = -x\ (p_N,\ P)$.

Если $p_i \in P$, то поскольку $x(k_{ij}) = -x(k_{ji})$, $x(p_i, P) = 0$. Допустим, что множество P разбито на 2 части : $P = P' \cup P''$, причем $p_1 \in P'$, $p_N \in P''$.

Тогда имеет место следующая цепочка равенств:

$$x_1 = x(p_1, P) = \sum_{p_i \in P'} x(p_i, P) = x(P', P) =$$
$$= \underbrace{x(P', P')}_{0} + x(P', P'') = x(P', P''). \tag{4.17}$$

Итак, при любом разбиении P на P' и P'' и произвольном потоке $\mathbf{X} = \mathbf{X}(k_{ij})$ справедливо соотношение

$$x_1 = x(p_1, P) = x(P', P''), \tag{4.18}$$

где
$$p_1 \in P', \; p_N \in P'', \tag{4.19}$$

т. е. поток через сеть инвариантен к способу разбиения, лишь бы при этом выполнялось условие (4.19).

Назовем множество Г коммуникаций, исходящих из P' и заканчивающихся в P'', *разрезом* Т-сети с источником p_1 и стоком p_N.

Тогда под пропускной способностью разреза d (Г) будем понимать суммарную пропускную способность всех коммуникаций, составляющих разрез Г:

$$d(\Gamma) = \sum_{k_{ij} \in \Gamma} d(k_{ij}). \tag{4.20}$$

Пусть $x(k_{ij})$, $k_{ij} \in K'$ — любой поток сети (P, K), совместимый с $d(k_{ij})$, а Г — произвольный разрез сети. Тогда имеет место неравенство $x_1 \leqslant d(\Gamma)$.

Действительно,

$$x_1 = x(P', P'') = \sum_{k_{ij} \in \Gamma} x(k_{ij}) \leqslant \sum_{k_{ij} \in \Gamma} d(k_{ij}) = d(\Gamma). \tag{4.21}$$

Из (4.21) следует признак оптимальности потока: Если величина потока x_1^* оказывается равной пропускной способности некоторого разреза Г, то x_1^* — максимальный поток.

Назовем разрез Г* сети (P, K) минимальным, если $d(\Gamma^*) \leqslant d(\Gamma)$ для всех разрезов Г.

Справедлива следующая теорема Форда — Фалкерсона [44]. *Если существует максимальный поток на сети со значением* x_1^*, *то существует также и минимальный разрез этой сети с пропускной способностью* $d(\Gamma^*)$, *причем* $x_1^* = d(\Gamma^*)$.

Доказательство этой теоремы будет приведено в следующем разделе.

Алгоритм Форда—Фалкерсона

Алгоритм позволяет найти максимальный поток на сети, совместимый с некоторой функцией $d(k_{ij})$ [44, 58].

Вводим следующие определения.

Пусть $\mathbf{X} = x(k_{ij})$, $k_{ij} \in K$ — произвольный алгебраический поток с функцией $d(k_{ij}) = d_{ij}$. Коммуникация $k_{ij} \in K'$ называется насыщенной, если $x(k_{ij}) = d_{ij}$, если же $x(k_{ij}) < d_{ij}$, то она называется ненасыщенной.

Алгоритм Форда состоит из однотипных итераций, на каждой из которых либо устанавливается максимальность имеющегося потока, либо строится новый поток с величиной $x_1(r+1) > x_1(r)$.

Каждая итерация заключается в последовательном построении некоторого множества отмеченных пунктов сети.

Прежде всего отмечаем p_1. В процессе проведения итерации в множество отмеченных пунктов включаются все новые пункты.

Назовем совокупность операций, в результате которых число отмеченных пунктов увеличивается на 1 либо фиксируется, что множество отмеченных пунктов полностью определено, *отдельным шагом итерации*.

Допустим, что проделано t шагов, в результате которых получено множество P_t отмеченных пунктов сети. Очередной $(t+1)$-й шаг состоит в упорядоченном просмотре пунктов множества P_t и отыскании такого отмеченного p_λ, из которого выходит ненасыщенная коммуникация, которая заканчивается на некотором неотмеченном пункте.

Если такой пункт p_λ находится, то неотмеченный пункт p_μ, являющийся концом ненасыщенной коммуникации $k_{\lambda\mu}$, отмечается числом λ и $(t+1)$-й шаг заканчивается.

При этом возможен один из двух случаев:

а) p_μ не является стоком p_N,

б) p_μ является стоком p_N.

В случае а) переходим к следующему шагу итерации, положив

$$P_{t+1} = \{P_t, p_\mu\}.$$

Может оказаться, что, просмотрев все пункты P_t, мы не найдем такого p_μ, который связан ненасыщенной коммуникацией $k_{\lambda\mu}$ с некоторым $p_\mu \in (P \setminus P_t)$.

Назовем эту возможность случаем в).

Случай в) означает, что P_t — полное множество отмеченных пунктов.

Итак, каждый шаг итерации заканчивается одним из случаев а), б) или в).

В случае а) продолжаем итерацию, а в случае б) и в) фиксируем построение полного множества отмеченных пунктов, которое обозначим через P'. Поэтому через конечное число шагов итерации мы приходим к случаю б) или в).

Рассмотрим каждый из этих случаев отдельно.

Случай б). Он означает, что построено множество P', которое содержит сток p_N. Покажем, что это дает возможность построить поток $\mathbf{X}(r+1)$, величина которого больше величины $\mathbf{X}(r)$.

Образуем последовательность $\{p_{i_\alpha}\}$ пунктов множества P по следующему правилу.

Положим $p_{i_1} = p_N$. Если пункты $p_{i_1}, p_{i_2}, \ldots, p_{i_{s-1}}$ уже найдены, то индекс i_s, определяющий s-й пункт, положим равным числу, с помощью которого был отмечен пункт $p_{i_{s-1}}$. Формирование последовательности проводим до тех пор, пока не приходим к p_1.

В результате получим следующий маршрут из p_N в p_1:

$$p_N = p_{i_1}, p_{i_2}, \ldots, p_{i_{s-1}}, p_{i_s} = p_1.$$

По правилу образования P' (множества отмеченных пунктов) $k_{i_{\lambda+1}, i_\lambda}$ — ненасыщенные коммуникации при $1 \leqslant \lambda \leqslant s-1$. Положим

$$\theta = \min_{1 \leqslant \lambda \leqslant s-1} (d_{i_{\lambda+1} i_\lambda} - x_{i_{\lambda+1} i_\lambda}). \qquad (4.22)$$

Тогда новый поток $\mathbf{X}$ $(r+1)$ определяется соотношением

$$x_{ij}^{(r+1)} = \begin{cases} x_{ij}^{(r)} + \theta, \text{ если } k_{ij} = k_{i_{\lambda+1}, i_\lambda} \ (1 \leqslant \lambda \leqslant s-1), \\ x_{ij}^{(r)} \text{ — в противном случае.} \end{cases} \qquad (4.23)$$

Очевидно, что в силу (4.22) новый поток $\mathbf{X}$ $(r+1)$ также удовлетворяет ограничениям вида $x(k_{ij}) \leqslant d_{ij}$. При этом $x_1^{(r+1)} = x_1^{(r)} + \theta$, т. к. среди всех компонент $x(k_{ij})$ потока $\mathbf{X}$ $(r+1)$ изменилась лишь одна x $(k_{i_s i_{s-1}})$, увеличившись на θ.

Случай в). Он имеет место, когда построенное множество не содержит сток p_N, и, кроме того, оно не может быть расширено.

Покажем, что это свидетельствует о максимальности потока, т. е.

$$x_1 = x_{1\max}.$$

Пусть P'' множество неотмеченных пунктов. Очевидно, P' и P'' не имеют общих пунктов и, кроме того, $P' \cup P'' = P$, $p_1 \in P'$, а $p_N \in \in P''$.

Поэтому множество Γ коммуникаций, идущих от P' к P'', есть разрез сети (P, K').

Если $k_{ij} \in \Gamma$, то $x(k_{ij}) = d_{ij}$, (4.24)

т. к. в противном случае можно было бы присоединить p_j к множеству P_t.

Учитывая (4.20) и (4.24), получим

$$d(\Gamma) = \sum_{k_{ij} \in \Gamma} d(k_{ij}) = \sum_{k_{ij} \in \Gamma} x(k_{ij}) = x(P', P'') = x_1, \qquad (4.25)$$

откуда на основе признака оптимальности потока заключаем, что $x_1 = x_{1\max}$.

Теперь можно доказать теорему Форда — Фалкерсона о максимальном потоке. Пусть Γ^* — разрез, который определяется множествами P' и P'', построенными на последней итерации, которая закончилась случаем в), а x_1 — соответствующий максимальный поток.

Тогда согласно (4.25) получим

$$x_1 = x_{1\max} = d(\Gamma^*).$$

Но, с другой стороны, для любого разреза сети Γ справедливо соотношение (4.21)

$$x_1 \leqslant d(\Gamma).$$

Используя это неравенство, приходим к соотношению:

$$d(\Gamma^*) \leqslant d(\Gamma) \text{ для всех разрезов в сети } \Gamma. \qquad (4.26)$$

Таким образом, разрез Г* — минимален.

Отметим, что задача нахождения минимального разреза двойственна задаче о максимальном потоке, а теорема Форда — Фалкерсона является аналогом соответствующей теоремы двойственности для транспортной сети.

Для удобства алгоритм Форда — Фалкерсона разбивается на 2 части: (вспомогательную) предварительную и основную.

Такое разбиение сделано для того, чтобы в случае существования потоков, совместимых с функцией D и имеющих неограниченные значения (тогда $\theta = \infty$), неограниченность потока можно было бы установить за конечное число итераций.

Первая итерация проводится, отправляясь от нулевого потока.

Каждая последующая итерация проводится, исходя из потока, построенного на предыдущей итерации.

Сначала следуют итерации предварительной части. Перед началом каждой из них из сети удаляются насыщенные коммуникации. Итерации предварительной части проводятся до тех пор, пока одна из них не завершится исходом в) или исходом с) (т. е. когда $\theta = \infty$).

При исходе с) задача решения не имеет. При исходе в) необходимо перейти к основной части, перед началом которой восстанавливают все коммуникации, выброшенные на предыдущих итерациях. На протяжении всех итераций основной части алгоритма множество коммуникаций остается неизменным. (В случае, когда существование максимального потока гарантировано заранее, необходимость в разделении алгоритма на предварительную и основную части отпадает.)

Способы усовершенствования вычислительной схемы алгоритма. 1. Порядок просмотра отмеченных пунктов играет определенную роль для повышения эффективности алгоритма. Можно предложить следующие варианты просмотра:

а) каждому вновь отмеченному пункту присваивается номер, на 1 больше номера последнего просмотренного пункта;

б) просматривать множество неотмеченных пунктов, пока не обнаружится некоторый неотмеченный пункт, связанный ненасыщенной коммуникацией с одним из отмеченных пунктов. Тогда он включается в состав отмеченных.

2. Иногда целесообразно строить маршрут, двигаясь от p_N к p_1. Тогда во множество P' включаются последовательно пункты, из которых исходит хотя бы одна коммуникация, заканчивающаяся в одном из уже отмеченных пунктов.

3. Наконец, можно отыскивать маршрут из p_1 в p_N, состоящий из ненасыщенных коммуникаций, двигаясь из p_1 и p_N навстречу одновременно. При этом строится 2 множества P' и P'' отмеченных пунктов, из которых одно содержит p_1, а второе p_N. Как только обнаружен пункт, подлежащий включению в оба множества, искомый маршрут найден, и поток можно улучшить. Если же такой пункт найти нельзя,

а отмеченные множества нельзя больше расширить — то это свидетельствует об оптимальности потока.

4. После нахождения некоторого ненасыщенного маршрута находится величина θ, которую назовем *степенью ненасыщенности* маршрута.

В общем случае p_1 и p_N могут быть связаны многими ненасыщенными маршрутами. Поэтому разумно организовать такой процесс формирования ненасыщенного маршрута, который дает максимально возможное значение $\theta_{\max}$.

Для этого достаточно лишь некоторое незначительное изменение алгоритма, состоящее в следующем.

Каждый отмеченный пункт p_i (кроме p_1) снабжается дополнительным индексом θ_i, который равен максимальной степени ненасыщенности маршрута, ведущего из p_1 в p_i. Индекс θ_i вычисляется одновременно с присоединением p_i к P_t. Это делается так.

На каждом шаге итерации просматриваем все неотмеченные пункты и среди них выбираем те, в которых заканчивается одна из ненасыщенных коммуникаций.

Для каждого такого p_i вычисляется

$$\delta_i = \max_{\lambda} \{\min(\theta_\lambda;\ d_{\lambda i} - x_{\lambda i})\},$$

где максимум берется по всем из отмеченных пунктов p_λ, из которых выходят ненасыщенные коммуникации, заканчивающиеся в p_i.

Далее среди всех δ_i выбирается такой i', для которого $\delta_{i'} = \max_{i} \{\delta_i\}$. Тогда пункт $p_{i'}$ включается в число отмеченных, причем $\theta_{i'} = \delta_{i'}$.

Если $\max_{i} \{\delta_i\}$ достигается на нескольких индексах i, то отмечаются все соответствующие пункты, причем каждому из них приписывается число, равное $\max \delta_i$.

Маршрут, построенный таким образом, имеет максимальную степень ненасыщенности, что позволяет снизить общее число итераций по определению максимального потока.

Пример 3.7. Найти максимальный поток в сети, заданной на рис. 3.10.

Первая итерация. Исходным является 0-поток. В множество отмеченных пунктов включаются: p_1, p_2 (1), p_5 (2), p_8.

Здесь используем такой порядок просмотра, когда вновь отмеченный пункт получает минимальный свободный порядковый номер (т. е. просматривается в первую очередь).

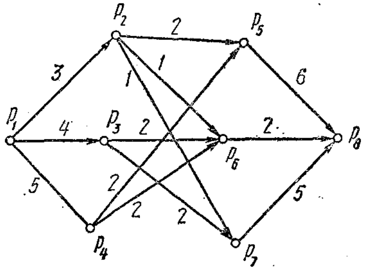


Рис. 3.10.

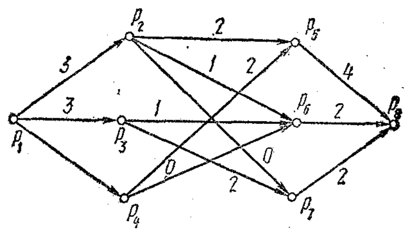


Рис. 3.11.

Итерация заканчивается исходом б).

Переходим к потоку X (1), полагая:

$$x_{12}^{(1)} = x_{25}^{(1)} = x_{58}^{(1)} = \min(3,\ 2,\ 6) = 2.$$

Выбрасываем из K насыщенную коммуникацию k_{25}.

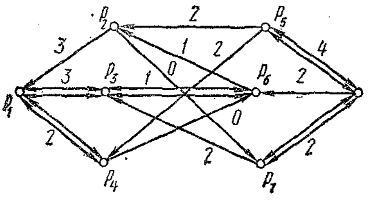


Рис. 3.12.

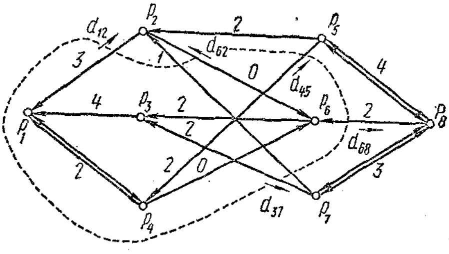


Рис. 3.13.

Вторая итерация. Множество $P' = \{p_1,\ p_2\,(1),\ p_6(2),\ p_8\,(6)\}$. Переходим к потоку X (2), изменяя X (1)

$$\theta = \min\{3,\ 2,\ 1,\ 2\} = 1,$$

$$x_{12}^{(2)} = x_{11}^{(1)} + \theta = 2 + 1 = 3; \quad x_{26} = x_{68} = 1.$$

Выбрасываем насыщенные коммуникации k_{12} и k_{26}.

Третья итерация $P' = \{p_1,\ p_3,\ p_6,\ p_8\}$,

$$\theta_3 = \min\{d_{13},\ d_{36},\ d_{68} - x_{68}^{(2)}\} = \min(4,\ 2,\ 1) = 1.$$

Тогда $x_{13}^{(3)} = x_{36}^{(3)} = 1$, $x_{68}^{(3)} = x_{68}^{(2)} + \theta = 1 + \theta = 2$, выбрасываем k_{68}.

Четвертая итерация $P' = \{p_1,\ p_3^{(1)},\ p_7^{(3)},\ p_8^{(7)}\}$.

Поток $x_{13}^{(4)} = 1 + \theta$; $x_{37}^{(4)} = x_{78}^{(4)} = \theta$, где $\theta = \min\{3,\ 2,\ 5\} = 2$, выбрасываем k_{37}.

Пятая итерация $P' = \{p_1,\ p_4\,(1),\ p_5\,(4),\ p_8\,(5)\}$.

Снова имеет место случай б) и переходим к X(5).

$$\theta = \min\{5,\ 2,\ 4\} = 2; \quad x_{14}^{(5)} = x_{45}^{(5)} = \theta; \quad x_{58}^{(5)} = 2 + \theta = 5.$$

Коммуникация k_{45} также исключается из K.

Шестая итерация. Строим множество отмеченных пунктов P'.

Последовательно отмечаем p_1; p_3 (1); p_6 (3); p_4 (1).

Дальнейшее расширение множества отмеченных пунктов невозможно. Итерация заканчивается исходом в) (см. рис. 3.11).

Переходим к основной части алгоритма, восстановив все удаленные коммуникации, при этом у насыщенных коммуникаций изменяем направление стрелок на противоположное, так как поток по ним может только уменьшаться, а двойными стрелками отметим ненасыщенные коммуникации с ненулевым потоком. Соответствующая Т-сеть приведена на рис. 3.12.

Седьмая итерация. На рис. 3.12 приведена Т-сеть $(P,\ K^{(6)})$, где $K^{(6)}$ — множество ненасыщенных коммуникаций $k_{ij} \in K'$.

Построим множество отмеченных пунктов P'.

$$P' = \{p_1,\ p_3\,(1),\ p_6,\ p_2,\ p_7,\ p_8\}.$$

Имеет место исход б). Маршрут состоит из ненасыщенных коммуникаций k_{13}, k_{36}, k_{62}, k_{27}, k_{78}.

Поток X(7) определяется следующими формулами:

$$\theta = \min\{d_{13} - x_{13},\ d_{36} - x_{36},\ x_{26},\ d_{27} - x_{27},\ d_{78} - x_{78}\} = \min\{1,\ 1,\ 1,\ 1,\ 3\} = 1,$$

$$x_{13}^{(7)} = 3 + \theta = 4; \quad x_{36}^{(7)} = 2; \quad x_{26} = 0; \quad x_{27} = 1; \quad x_{78} = 2 + \theta = 3.$$

Восьмая итерация. Строим множество отмеченных пунктов $P'=\{p_1, p_4, p_6, p_3\}$. Четвертый шаг заканчивается случаем в) (см. описание алгоритма).

Дальнейшее продолжение процесса невозможно. Следовательно, $x_1(7)$ — максимальный поток: $x_1(7) = x_{1\max} = 9$.

Минимальный разрез Г* сети состоит из коммуникаций (рис. 3.13)
$k_{12}, k_{45}, k_{68}, k_{62}, k_{37}$.
Очевидно,

$$d(\Gamma^*) = d_{12} + d_{45} + d_{68} + d_{62} + d_{37} = 3 + 2 + 2 + 0 + 2 = 9 = x^*_{1\max}.$$

§ 5. ТРАНСПОРТНАЯ ЗАДАЧА В СЕТЕВОЙ ПОСТАНОВКЕ

Постановка задачи

Пусть задана произвольная транспортная сеть (P, K). На множестве P пунктов сети определена функция производства и потребления $q(p_i)$; на множестве K заданы функция $d(k_{ij})$ пропускных способностей коммуникаций сети и функция транспортных издержек $c(k_{ij}) \geqslant 0$.

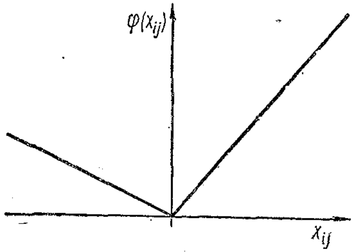


Рис. 3.14.

Рассмотрим транспортную задачу на сети, которая является обобщением обычной Т-задачи.

Транспортная задача на сети (P, K) с функциями $\mathbf{Q} = \|q(p_i)\|$, $\mathbf{D} = \|d(k_{ij})\|$ и $\mathbf{C} = \|c(k_{ij})\|$ может быть задана на множестве как арифметических, так и алгебраических планов.

Напомним, что функция $\mathbf{X} = \|x(k_{ij})\|$, удовлетворяющая условиям

$$x(p_i, P) = \sum_{k_{ij} \in E_i} x(k_{ij}) = q(p_i), \qquad i = 1, 2, \ldots, N, \tag{5.1}$$

$$x(k_{ij}) \leqslant d(k_{ij}) \qquad k_{ij} \in K', \tag{5.2}$$

$$x(k_{ij}) = -x(k_{ji}), \tag{5.3}$$

называется алгебраическим планом, совместимым с функциями $\mathbf{Q}$, $\mathbf{D}$ и $\mathbf{C}$. Множество K' получается из K добавлением таких коммуникаций $k_{ij} \notin K$, что $k_{ji} \in K$. При этом полагаем $x(k_{ij}) = d(k_{ij}) = 0$, если $k_{ij} \notin K$.

Образуем с помощью функции $c(k_{ij})$ систему функций $\varphi_{ij}(x_{ij})$, определяющих стоимость перевозки x_{ij} по коммуникации k_{ij}:

$$\varphi_{ij}(x_{ij}) = \begin{cases} c_{ij}x_{ij}, & \text{если } x_{ij} \geqslant 0; \\ -c_{ji}x_{ij}, & \text{если } x_{ij} < 0. \end{cases}$$

Функция $\varphi_{ij}(x_{ij})$ в общем случае кусочно-линейная выпуклая вниз с одним изломом (рис. 3.14).

Транспортная задача на сети (P, K) с функциями производства — потребления $\mathbf{Q} = \|q(p_i)\|$, пропускных способностей $\mathbf{D} = \|d(k_{ij})\|$ и транспортных издержек $\mathbf{C}$ состоит в построении такого плана перевозок $\mathbf{X} = \|x(k_{ij})\|$, $k_{ij} \in K'$, удовлетворяющего условиям (5.1) —

(5.3), для которого суммарные транспортные расходы

$$\Phi(\mathbf{X}) = \frac{1}{2} \sum_{k_{ij} \in K'} \varphi_{ij}(x_{ij}) \tag{5.4}$$

достигают минимума.

Задача (5.4) не является задачей ЛП из-за нелинейности функций Ф (X), но может быть легко приведена к линейной. Для этого достаточно перейти от алгебраических планов к арифметическим, и целевая функция становится линейной.

Транспортная задача на сеть (P, K) с функциями **Q**, **D** и **C** в терминах арифметических планов состоит в минимизации целевой функции

$$L(\mathbf{X}) = \sum_{k_{ij} \in K} c_{ij} x(k_{ij}), \tag{5.5}$$

при условиях

$$x(p_i, P) - x(P, p_i) = q(p_i), \qquad i = 1, 2, \ldots, N, \tag{5.6}$$

$$0 \leqslant x(k_{ij}) \leqslant d(k_{ij}), \qquad k_{ij} \in K. \tag{5.7}$$

Если $\mathbf{X} = \| x(k_{ij}) \|$, $k_{ij} \in K$ — арифметический план, то $\mathbf{У} = y(k_{ij}) = x(k_{ij}) - x(k_{ji})$, $k_{ij} \in K'$ — алгебраический план. Аналогичным образом каждому алгебраическому плану $\mathbf{У} = \|y(k_{ij})\|$ отвечает арифметический план

$$\mathbf{X} = x(k_{ij}) = \begin{cases} y(k_{ij}), & \text{при } y(k_{ij}) \geqslant 0, \\ 0, & \text{при } y(k_{ij}) < 0. \end{cases}$$

Легко проверить, что значения функций (5.4) и (5.5) на соответствующих друг другу алгебраическом **У** и арифметическом **X** планах равны, если $x(k_{ij}) \cdot x(k_{ji}) = 0$ для $k_{ij} \in K$. Поэтому задачи (5.1)—(5.3) и (5.5)—(5.7) эквивалентны.

В дальнейшем для удобства транспортную задачу на сети с функциями **Q**, **D** и **C** будем обозначать $T(q, d, c)$.

Для разрешимости задачи $T(q, d, c)$ необходимо и достаточно, чтобы существовал по крайней мере один план, удовлетворяющий условиям (5.2), (5.3) или (5.6), (5.7).

Необходимое условие разрешимости задачи $T(q, d, c)$ — это *обобщенное условие баланса* для сети $\sum_{p_i \in P} q(p_i) = 0$.

Произвольную задачу $T(q, d, c)$ можно легко свести к случаю, когда $q(p_i) = 0$ для всех p_i, кроме источника и стока. Для этого достаточно: 1) ввести два новых пункта p_0 и p_{N+1}; 2) из p_0 направить коммуникацию во все пункты производства; 3) из каждого пункта потребления направить коммуникацию в p_{N+1}, 4) доопределить $d(k_{ij})$ следующим образом:

$$d(k_{0j}) = q(p_j), \qquad d(k_{i,\,N+1}) = -q(p_i),$$

где p_j — пункты производства, а p_i — пункты потребления, 5) положить $c(k_{0j}) = c(k_{i,N+1}) = 0$.

Рассмотрим транспортную сеть $(\bar{P}, \bar{K})$ с функциями $\bar{\mathbf{Q}}$, $\bar{\mathbf{D}}$ и $\bar{\mathbf{C}}$, где $\bar{P} = \{P, p_0, p_{N+1}\}$; $\bar{K} = \{K, k_{0j}, k_{i,N+1}\}$, $\bar{\mathbf{D}}$ и $\bar{\mathbf{C}}$ указанные продолжения функций **D** и **C** на множество $\bar{K}$;

$$\bar{q}(p_i) = \begin{cases} \sum\limits_{q(p_\lambda)>0} q(p_\lambda), & \text{если } i = 0; \\ \sum\limits_{q(p_s)<0} q(p_s), & \text{если } i = N+1; \\ 0, & \text{для } 1 \leqslant i \leqslant N. \end{cases} \tag{5.8}$$

Задача $T(\bar{q}, \bar{d}, \bar{c})$ эквивалентна задаче $T(q, d, c)$. Действительно, каждому плану **X** задачи $T(q, d, c)$ отвечает план $\bar{\mathbf{X}}$ задачи $T(\bar{q}, \bar{d}, \bar{c})$, для которого

$$\bar{x}_{ij} = \begin{cases} q(p_j), & \text{если } i = 0,\ q(p_j) > 0; \\ -q(p_i), & \text{если } j = N+1,\ q(p_i) < 0; \\ x_{ij}, & \text{в остальных случаях,} \end{cases} \tag{5.9}$$

причем оба плана связаны с одинаковыми транспортными издержками. С другой стороны, для любого плана $\bar{\mathbf{X}}$ задачи $T(\bar{q}, \bar{d}, \bar{c})$

$$\bar{x}_{0j} = d_{0j} = q(p_j); \quad q(p_j) > 0,$$

$$x_{i,N+1} = d_{i,N+1} = -q(p_i); \quad q(p_i) < 0. \tag{5.10}$$

Следовательно, система перевозок $x_{ij} = \bar{x}_{ij}$ $(i \neq 0,\ j \neq N+1)$ составляет план задачи $T(q, d, c)$, причем переход от плана $\bar{\mathbf{X}}$ к плану **X** не изменяет суммарных транспортных расходов, так как $c_{0j} = c_{i,N+1} = 0$.

Критерии оптимальности плана задачи $T(q, d, c)$. Сформулируем критерий оптимальности плана задачи.

Свяжем с пунктом p_i сети (P, K) число u_i, называемое его потенциалом. Справедлив следующий критерий оптимальности арифметического плана задачи $T(q, d, c)$.

Теорема 3.6. *Для оптимальности плана* $\mathbf{X} = x(k_{ij})$, $k_{ij} \in K$ *задачи* $T(q, d, c)$ *необходимо и достаточно существование чисел* u_1, u_2, …, u_N *таких, что*

$$u_j - u_i = c(k_{ij}), \quad \text{если } 0 < x(k_{ij}) < d(k_{ij}); \tag{5.11}$$

$$u_j - u_i \leqslant c(k_{ij}), \quad \text{если } x(k_{ij}) = 0; \tag{5.12}$$

$$u_j - u_i \geqslant c(k_{ij}), \quad \text{если } (k_{ij}) = d(k_{ij}). \tag{5.13}$$

Сравнив условия (5.11)—(5.13) с соответствующими условиями оптимальности опорного плана T_d-задачи, убеждаемся в их полной аналогии. Поэтому справедливость теоремы 3.6 может быть доказана аналогично теореме 3.3.

Сформулируем теперь аналогичный критерий оптимальности для алгебраического плана задачи $T(q, d, c)$.

Теорема 3. 7. *Для оптимальности алгебраического плана* $X = x(k_{ij})$ $k_{ij} \in K'$ *задачи* (5.1)—(5.4) *необходимо и достаточно существование чисел* u_1, u_2, …, u_N *таких, что*

$$u_j - u_i = c(k_{ij}), \quad \text{если } 0 < x(k_{ij}) < d(k_{ij}); \tag{5.14}$$

$$u_j - u_i = -c(k_{ji}), \quad \text{если } -d(k_{ji}) < x(k_{ij}) < 0; \tag{5.15}$$

$$u_j - u_i \geqslant c(k_{ij}), \quad \text{если } x(k_{ij}) = d(k_{ij}); \tag{5.16}$$

$$u_j - u_i \leqslant -c(k_{ji}), \quad \text{если } x(k_{ij}) = -d(k_{ji}); \tag{5.17}$$

$$-c(k_{ji}) \leqslant u_j - u_i \leqslant c(k_{ij}), \quad \text{если } x(k_{ij}) = 0. \tag{5.18}$$

Заметим, что в условиях (5.14)—(5.18) величина $c(k_{ij}) = \infty$, если $k_{ij} \notin K$.

Опорные планы транспортной сети. Понятие опорного плана, введенное ранее при изучении Т-задачи, может быть легко распространено на случай сети.

Определение. *Арифметический план* $X = x(k_{ij})$, $k_{ij} \in K$ *называется опорным, если нельзя составить замкнутый маршрут из коммуникаций плана, для которых*

$$0 < x(k_{ij}) < d(k_{ij}). \tag{5.19}$$

Для алгебраического плана Т-сети признак опорности выглядит следующим образом.

Алгебраический план $X = x(k_{ij})$, $k_{ij} \in K'$ называется опорным, если невозможно составить направленный замкнутый маршрут из коммуникаций $k_{ij} \in K'$, для которых выполняется условие

$$x(k_{ij}) \neq d_{ij},\ 0, -d_{ji}. \tag{5.20}$$

Отметим, что коммуникации, удовлетворяющие (5.19) для арифметического плана и (5.20) для алгебраического, называют *основными*. Поэтому опорный план можно определить как план, из основных коммуникаций которого нельзя составить замкнутый маршрут.

Объединим в множество K_x основные коммуникации опорного плана X.

Опорный план X называется невырожденным, если сеть (P, K_x) связана, т. е. если любые два пункта P могут быть соединены маршрутом, состоящим из основных коммуникаций $k_{ij} \in K_x$.

Заметим, что в соответствии с определением основных коммуникаций алгебраического плана пункты p_i и p_j либо вообще не связаны коммуникациями множества K_x, либо соединены двумя противоположными коммуникациями множества K_x.

Справедлива следующая теорема.

Теорема 3.8. *Множество основных коммуникаций K_x состоит ровно из $(N-1)$ пар противоположных коммуникаций, где N — общее число пунктов сети.*

Назовем пункты Т-сети, связанные хотя бы одной коммуникацией, соседними. Введем следующие обозначения: P_1 — множество пунктов сети (P, K_x), соседних пункту p_1, P_i $(i = 1, 2...)$ — множество пунктов сети (P, K_x), соседних одному из пунктов множества P_{i-1}.

Очевидно, множества P_i $(i = 1, 2...)$ не имеют общих пунктов, ибо в противном случае неизбежно существовал бы замкнутый маршрут из основных коммуникаций $k_{ij} \in K_x$, а это противоречит признаку опорности плана X.

Отсюда, в частности, вытекает существование числа t, начиная с которого (при $i > t$) множества P_i не содержат ни одного пункта.

Из связности сети (P, K) вытекает, что

$$\bigcup_{i=0}^{t} P_i = P, \tag{5.21}$$

где $P_0 = p_1$; $\underset{\text{при } i \neq j}{P_i \cap P_j} = \varnothing$.

Поставим в соответствие каждому пункту p_λ ($p_\lambda \in P_i$) пару противоположных коммуникаций $k_{\lambda\mu}$, $k_{\mu\lambda}$ из K_x, где μ — номер пункта, соседнего с p_λ, т. е. $p_\mu \in P_{i-1}$. Множество всех таких коммуникаций обозначим через $\tilde{K}_x$.

Учитывая доказанное равенство (5.21), получим, что $\tilde{K}_x$ состоит из ($N - 1$) пар противоположных коммуникаций. Поэтому доказательство утверждения сводится к проверке равенства

$$\tilde{K}_x = K_x \tag{5.22}$$

Пусть k_{ij} — произвольная коммуникация $k_{ij} \in K_x$. В соответствии с правилом определения K_x, всегда существует некоторый маршрут l, связывающий пункты p_i и p_j. Теперь допустим, что $k_{ij} \notin \tilde{K}_x$. Но тогда коммуникации маршрута l и коммуникация $k_{ij} \notin l$ составили бы замкнутый маршрут, что противоречит опорности плана X. Остается принять, что если $k_{ij} \in K_x$, то $k_{ij} \in \tilde{K}_x$.

А так как это справедливо для всех k_{ij}, то равенство (5.22) справедливо.

Итак, K_x состоит из ($N - 1$) пар коммуникаций, причем для любого пункта $p_s \in P_i$, существует единственный соседний пункт p_μ ($p_\mu \in P_{i-1}$).

Множества P_i могут быть использованы для нахождения направленного маршрута из p_1 в произвольный пункт $p \in P_i$.

Действительно, находим последовательно пункт p_{i_1}, соседний с p_1, p_{i_2} — соседний с p_{i_1} и т. д., на i-м шаге приходим к $p \in P_i$. Маршрут, проходящий через пункты $p_1 = p_{i_0}, p_{i_1}, p_{i_2}, \dots, p_{i_s}$, будет искомым.

Метод потенциалов для транспортной задачи на сети

Метод потенциалов, разработанный для обычной транспортной задачи, естественным образом обобщается на сетевую транспортную задачу.

Процесс решения задачи T (q, d, c) методом потенциалов состоит из предварительного этапа и конечного числа итераций. На предварительном этапе строится начальный опорный план. Каждая итерация состоит из двух этапов. На первом этапе проверяется на оптимальность ранее полученный опорный план. Если он не удовлетворяет условиям оптимальности, то на втором этапе строится новый, более экономичный опорный план.

Будем сначала предполагать, что опорные планы задачи T (q, d, c) — невырожденные. Необходимые замечания относительно общего случая сделаем позднее.

Описание $(r+1)$-й итерации.

Допустим, что уже проведено r итераций и получен невырожденный опорный план $\mathbf{X} = x(k_{ij})$, $k_{ij} \in K$.

Первый этап. Выбираем произвольный пункт множества P, например p_1, и, начиная с него, рассчитываем предварительные потенциалы $u_i = u(p_i)$, $p_i \in P$ плана $\mathbf{X}$ согласно соотношениям (5.14), (5.15)

$$u(p_j) - u(p_i) = \begin{cases} c_{ij} = c(k_{ij}) & \text{при } x(k_{ij}) > 0, \ k_{ij} \in K_x; \\ -c_{ji} = -c(k_{ji}) & \text{при } x(k_{ij}) < 0, \ k_{ij} \in K_x. \end{cases} \tag{5.23}$$

Предварительные потенциалы u_i вычисляются последовательно для пунктов, принадлежащих P_0, P_1, ..., и т. д. в соответствии с правилом:

$$u_1 = u(p_1) = 0;$$

$$u_s - u_t = \begin{cases} c(k_{ts}) & \text{при } x(k_{ts}) > 0; \\ -c(k_{st}) & \text{при } x(k_{ts}) < 0, \end{cases} \tag{5.24}$$

где $p_s \in P_i$, а $p_t \in P_{i-1}$, причем p_t, p_s — соседние пункты сети $(P,\ K_x)$.

Вследствие свойств множеств P_i, на основе соотношений (5.24) можно однозначно вычислить значения u_i для всех $p_i \in P$. Далее для каждой неосновной коммуникации $k_{ij} \in K'$ составляем разность потенциалов $u_j - u_i$. Если выполняются условия

$$\left.\begin{array}{ll} -c_{ji} \leqslant u_j - u_i \leqslant c_{ij} & \text{при } x(k_{ij}) = 0; \\ u_j - u_i \geqslant c_{ij} & \text{при } x(k_{ij}) = d_{ij}; \\ u_j - u_i \leqslant -c_{ji} & \text{при } x(k_{ij}) = -d_{ji}, \end{array}\right\} \tag{5.25}$$

то в соответствии с критерием оптимальности (теорема 3.8) план $\mathbf{X}$ — оптимальный. Если же хотя бы для одной неосновной коммуникации условие (5.25) будет нарушено, то переходим ко второму этапу. Поскольку для алгебраического плана условия (5.25) либо одновременно соблюдаются для пары противоположных коммуникаций k_{ij} и k_{ji}, либо не соблюдаются ни для одной из них, данные условия достаточно проверять для одной из них (k_{ij} или k_{ji}).

Второй этап. Рассматриваются пары пунктов сети, для которых нарушены условия (5.25). Среди них выбирается некоторая пара p_λ, p_μ. Критерием выбора может служить, например, величина невязки нарушенного требования системы (5.25).

Дальнейшее течение этапа зависит от того, какое из условий (5.25) будет нарушено. Рассмотрим каждый из возможных случаев.

а) Пусть нарушена правая часть первого неравенства (5.25), т. е. имеет место $\quad u_\mu - u_\lambda > c_{\lambda\mu}$ при $x(k_{\lambda\mu}) = 0$.

Тогда строим направленный маршрут l из p_μ и p_λ, используя коммуникации $k_{ij} \in K_x$. Эта операция может быть осуществлена путем последовательного формирования множеств P_i, начиная с $P_0 = p_\mu$, до тех пор, пока пункт p_λ не попадет в множество P_s при некотором s. Пусть направленный маршрут l из μ в λ имеет вид

$$k_{\mu i_1},\ k_{i_1 i_2},\ \ldots,\ k_{i_{s-1} i_s},\ k_{i_s \lambda}.$$

Присоединив к маршруту l коммуникацию $k_{\lambda\mu}$, получим замкнутый маршрут γ.

Тогда новый план X' образуется из текущего плана X путем изменения составляющих потока $x(k_{ij})$ вдоль замкнутого маршрута согласно соотношению:

$$x'(k_{ij}) = \begin{cases} x(k_{ij}) + \theta, & \text{если } k_{ij} \in \gamma; \\ x(k_{ij}) - \theta, & \text{если } k_{ji} \in \gamma; \\ x(k_{ij}), & \text{во всех остальных случаях.} \end{cases} \qquad (5.26)$$

Параметр θ выбирается максимально возможным, но так, чтобы не были нарушены пропускные способности $d(k_{ij})$ коммуникаций и составляющие $x(k_{ij})$ не изменили знака.

Таким образом,

$$\theta = \min_{k_{ij} \in \gamma} \delta(k_{ij}), \qquad (5.27)$$

где

$$\delta(k_{ij}) = \begin{cases} d_{ij} - x(k_{ij}) & \text{при } x(k_{ij}) > 0, \ k_{ij} \in l; \\ -x(k_{ij}) & \text{при } x(k_{ij}) < 0, \ k_{ji} \in l; \\ d_{\lambda\mu} & \text{при } k_{ij} = k_{\lambda\mu}. \end{cases} \qquad (5.28)$$

Нетрудно показать, что $\mathbf{X}' = x'(k_{ij})$, определяемый соотношениями (5.26)—(5.27), будет также опорным планом задачи $T(q, d, c)$, так как при этом условия (5.1)—(5.3) будут выполняться.

Переход к новому плану $\mathbf{X}'$ приводит к изменению суммарных издержек Φ на величину $\Delta\Phi$.

$$\Delta\Phi = \Phi(\mathbf{X}') - \Phi(\mathbf{X}) = \sum_{\substack{x(k_{ij})>0 \\ k_{ij}\in l}} c_{ij}(x'(k_{ij}) - x(k_{ij})) - \sum_{\substack{x(k_{ij})<0 \\ k_{ij}\in l}} -c_{ji}(x'(k_{ij}) -$$

$$- x(k_{ij})) + c_{\lambda\mu}(x'(k_{\lambda\mu}) - x(k_{\lambda\mu})) = \theta\left[\sum_{\substack{x(k_{ij})>0 \\ k_{ij}\in l}} c_{ij} - \sum_{\substack{x(k_{ij})<0 \\ k_{ij}\in l}} c_{ji} + c_{\lambda\mu}\right]. \qquad (5.29)$$

Но учитывая далее соотношения (5.23) для $k_{ij} \in l$, получаем

$$\Phi(\mathbf{X}') - \Phi(\mathbf{X}) = \theta\left[\sum_{k_{ij}\in l} (u_j - u_i) + c_{\lambda\mu}\right] = \theta\,[u_{i_1} - u_\mu + u_{i_2} - u_{i_1} + \cdots$$

$$\cdots + u_{i_s} - u_{i_{s-1}} + u_\lambda - u_{i_s} + c_{\lambda\mu}) = \theta\,[c_{\lambda\mu} - (u_\mu - u_\lambda)]. \qquad (5.30)$$

Но по предположению $c_{\lambda\mu} < u_\mu - u_\lambda$, а параметр $\theta > 0$, как минимум среди положительных чисел. Поэтому

$$\Phi(\mathbf{X}') - \Phi(\mathbf{X}) = \theta\,[c_{\lambda\mu} - (u_\mu - u_\lambda)] < 0.$$

Следовательно, план X' будет более экономичным, чем предыдущий.

б) Допустим, что нарушено второе условие системы (5.25):

$$u_\mu - u_\lambda < c_{\lambda\mu}, \qquad (k_{\lambda\mu}) = d_{\lambda\mu}.$$

В этом случае план $\mathbf{X}'$ образуется из плана $\mathbf{X}$ путем увеличения на θ $(\theta > 0)$ перевозок вдоль коммуникаций замкнутого маршрута γ_1, состоящего из $k_{\mu\lambda}$ и маршрута l_1 из p_λ в p_μ. Замкнутый маршрут γ_1 отличается от γ только направлением. План X' вычисляется по формулам (5.26), (5.27), (5.28), в которых замкнутый маршрут γ заменен на γ_1.

Выполнив выкладки, аналогичные (5.29), (5.30), получим, что

$$\Phi(\mathrm{X}') - \Phi(\mathrm{X}) = \theta\,[(u_\mu - u_\lambda) - c_{\lambda\mu}] < 0,$$

причем X' также является опорным планом задачи.

Остальные 2 возможных случая нарушения условий (5.25) сводятся к уже рассмотренным случаям а) и б).

Действительно, пусть не соблюдается левая часть первого неравенства, т. е. $-c_{\mu\lambda} > u_\mu - u_\lambda$ при $x\,(k_{\lambda\mu}) = 0$. Но путем элементарных преобразований это соотношение сводится к виду $u_\lambda -$ $- u_\mu > c_{\mu\lambda}$ при $x\,(k_{\mu\lambda}) = 0$, а это случай а) для коммуникации $k_{\mu\lambda}$.

Если наконец нарушено последнее условие (5.25), т. е.

$$u_\mu - u_\lambda > -c_{\mu\lambda} \qquad \text{при } x\,(k_{\lambda\mu}) = -d_{\mu\lambda},$$

то, переходя к коммуникации $k_{\mu\lambda}$, приходим к случаю б), также рассмотренному выше:

$$u_\lambda - u_\mu < c_{\mu\lambda},\quad x\,(k_{\mu\lambda}) = d_{\mu\lambda}.$$

Таким образом, каждая итерация, содержащая второй этап, приводит к уменьшению целевой функции Φ. Следовательно, в предположении, что опорные планы X задачи невырожденные, оптимальный план $\mathrm{X}_{\text{опт}}$ может быть получен методом потенциалов за конечное число шагов.

Определение начального опорного плана

Рассмотрим один из возможных способов получения исходного опорного плана задачи $T\,(q, d, c)$.

Приведем задачу $T\,(q, d, c)$ к эквивалентной задаче с одним пунктом производства p_0 и одним пунктом потребления p_{N+1}, где $q\,(p_0) =$ $= \sum\limits_{q(p_i)>0} q\,(p_i) = -q\,(p_{N+1})$, в соответствии с правилом, указанным в начале § 5 на с. 191. Далее найдем максимальный поток $x_{1\max}$ на сети (P_1K) с источником p_0 и стоком p_{N+1}. Возможны 2 случая:

а) $x_{1\max} = \sum\limits_{q(p_i)>0} q\,(p_i)$;

б) $x_{1\max} < \sum\limits_{q(p_i)>0} q\,(p_i)$.

В случае а) компоненты потока X, отвечающие коммуникациям $k_{ij} \in$ $\in K'$, составляют план X задачи $T\,(q, d, c)$. Случай б) указывает на несовместимость условий задачи.

План X, к которому приходят в случае а), может оказаться неопорным. В таком случае осуществляется переход от плана X к опорному плану X_1 путем последовательного разрушения замкнутых цепочек. Процесс построения опорного плана X_1 по произвольному плану X состоит в следующем.

Пусть γ — замкнутый маршрут в плане X_1, составленный из его основных коммуникаций. Введем план X (θ), который связан с планом X соотношениями:

$$x(k_{ij},\ \theta) = x(k_{ij}) + \theta, \quad \text{если } k_{ij} \in \gamma.$$

Поскольку значения перевозок $x(k_{ij})$, $k_{ij} \in \gamma$ лежат внутри интервалов линейности функций $\varphi_{ij}(x_{ij})$, при малых абсолютных значениях параметра θ справедливо соотношение

$$\Phi(\mathrm{X}(\theta)) = \Phi(\mathrm{X}) + \beta\theta, \quad \text{где } \theta = \mathrm{const}.$$

Если $\beta < 0$, то будем θ увеличивать до тех пор, пока одна из перевозок маршрута γ не станет равной своему граничному значению (d_{ij}, 0 или $-d_{ij}$).

Если $\beta \geqslant 0$, необходимо уменьшить θ, придавая ему отрицательные значения до тех пор, пока опять одна из перевозок маршрута γ не станет равной одному из своих граничных значений. В обоих случаях приходим к новому плану X′, в котором отсутствует замкнутый маршрут и число основных коммуникаций меньше, а значение суммарных транспортных издержек не больше, чем в плане X. Если план X′ не является опорным, то повторяем описанный выше процесс, исходя из нового замкнутого маршрута γ_1.

Через несколько шагов мы обязательно придем к опорному плану X′, для которого $\Phi(\mathrm{X}') \leqslant \Phi(\mathrm{X})$.

Конечность числа шагов следует из того, что на каждом из них число основных коммуникаций сокращается.

При построенки начального плана X_0 полезно использовать аналог метода минимального элемента, используемый при нахождении опорных планов Т-задач, а именно: при выборе маршрута от источника p_0 к стоку p_{N+1} в первую очередь следует выбирать наиболее (выгодный) экономичный маршрут с минимальными суммарными издержками. Это позволяет найти более экономичный план X_0 и сократить (в среднем) общее число итераций.

Рассмотрим случай, когда опорный план оказывается вырожденным. В таком случае к числу основных коммуникаций K_x добавляются в качестве базисных коммуникации, соответствующие нулевым перевозкам $k_{ij} \in K_\varepsilon$ так, чтобы сеть $(P,\ K_x \bigcup K_\varepsilon)$ стала связной и не существовало бы ни единого замкнутого маршрута. Далее, применив ε-метод, описанный для вырожденных планов Т-задач, решаем задачу как невырожденную.

Заметим, что если число основных коммуникаций равно m ($m < N - 1$), то число дополнительно вводимых базисных коммуникаций r будет равно $r = N - 1 - m$.

Пример 3.8. Требуется решить транспортную задачу на сети, условия которой представлены на рис. 3.15.

При этом на рисунке используем следующие обозначения:

на каждой коммуникации k_{ij} в кружке число слева обозначает c_{ij}, а справа — d_{ij}. Примем, что матрицы C и D — симметричны, т. е. $c_{ij} = c_{ji}$, $d_{ij} = d_{ji}$.

Объемы производства равны $q(p_1) = 5$; $q(p_2) = 4$,

объемы потребления равны $q(p_5) = -3$, $q(p_6) = -6$.

Пункты p_3 и p_4 — перевалочные и для них $q(p_3) = q(p_4) = 0$.

Предварительный этап. Проверяем необходимое условие разрешимости $\sum_{i=1}^{6} q(p_i) = 5 + 4 - 3 - 6 = 0$, т. е. условие баланса выполняется. Строим начальный опорный план.

Так как $N = 6$, то число базисных элементов плана равно $N - 1 = 5$. Построим начальный план $\mathbf{X}_0 = \| x_{ij} \|$, используя подход, описанный в алгоритме нахождения максимального потока. Найденные величины потоков приводятся на рис. 3.16.

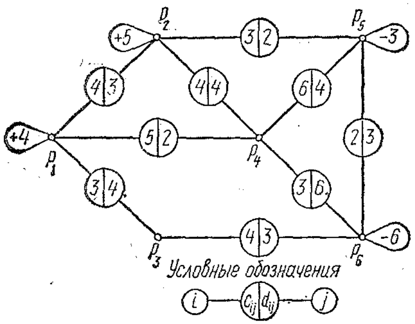


Рис. 3.15.

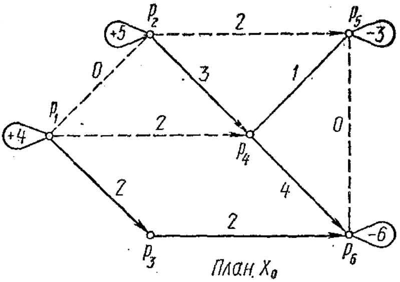


Рис. 3.16.

Так как $x_{12} = x_{65} = 0$, $x_{25} = d_{25} = 2$ и $x_{14} = d_{14} = 2$, то коммуникации k_{12}, k_{65}, k_{25} и k_{14} не являются базисными (основными). Все остальные коммуникации плана $\mathbf{X}_0$ $\{k_{13}, k_{36}, k_{46}, k_{24}, k_{45}\}$ являются основными, и так как сеть, составленная из этих коммуникаций, не содержит циклов и связана, то план $\mathbf{X}_0$ — опорный и невырожденный.

Первая итерация. Первый этап. Вычисляем предварительные потенциалы, используя схему Т-сети. Примем $u_1 = 0$. Тогда получим $u_3 = u_1 + c_{13} = 3$;

$$u_6 = u_3 + c_{36} = 3 + 4 = 7, \quad u_4 = u_6 - c_{46} = 7 - 3 = 4,$$
$$u_5 = u_4 + c_{45} = 10, \qquad u_2 = u_4 - c_{24} = 4 - 4 = 0.$$

Составляем разности потенциалов для всех неосновных коммуникаций и проверяем условия оптимальности

$$u_2 - u_1 = 0 < c_{12} = 4, \quad x_{12} = 0;$$
$$u_5 - u_2 = 10 > c_{25} = 2, \quad x_{25} = d_{25} = 2;$$
$$u_4 - u_1 = 4 < c_{14} = 5, \quad x_{14} = d_{14} = 2;$$
$$-c_{65} = -2 > u_6 - u_5 = 7 - 10 < c_{56} = 2, \quad x_{56} = 0.$$

Как видим, условия нарушены для коммуникаций k_{14} и k_{65}. Следовательно, переходим ко второму этапу.

Второй этап. Выбираем коммуникацию k_{14} $(x_{14} = d_{14})$. Строим направленный маршрут l из p_1 в p_4 из основных коммуникаций. Очевидно, он равен: $l = \{k_{13}, k_{36}, k_{64}\}$. Добавив к маршруту l коммуникацию k_{14}, получим замкнутый маршрут $\gamma = (l, k_{14})$. Находим $\theta_1 = \min\{x_{46}, d_{36} - x_{36}, d_{13} - x_{13}; x_{14}\} = \min\{4, 1, 2\} = 1$. Вычисляем новый план $\mathbf{X}^{(1)}$ согласно соотношениям (5.26):

$$x^{(1)}(k_{46}) = x(k_{46}) - \theta_1 = 4 - 1 = 3; \quad x_{14}^{(1)} = x_{14} - \theta_1 = 2 - 1 = 1;$$
$$x^{(1)}(k_{13}) = x(k_{13}) + \theta_1 = 2 + 1 = 3;$$
$$x^{(1)}(k_{36}) = 2 + 1 = 3;$$
$$x^{(1)}(k_{46}) = 4 - \theta_1 = 3.$$

Так как $x_{36}^{(1)} = d_{36}$, а $x_{14}^{(1)} < d_{14}$, то новому опорному плану соответствуют основные коммуникации k_{14}, k_{13}, k_{24}, k_{45}, k_{46}. Этот план $X^{(1)}$ приведен на рис. 3.17 (основные коммуникации указаны сплошными линиями, а в скобках — величины издержек c_{ij}).

Вторая итерация. Первый этап. Рассчитываем потенциалы, отвечающие плану $X^{(1)}$. Полагая $u_1 = 0$, получим

$$u_3 = 3, \ u_4 = u_1 + c_{14} = 5; \quad u_6 = u_4 + c_{46} = 5 + 3 = 8;$$
$$u_5 = u_4 + c_{45} = 11;$$
$$u_2 = u_4 - c_{24} = 5 - 4 = 1.$$

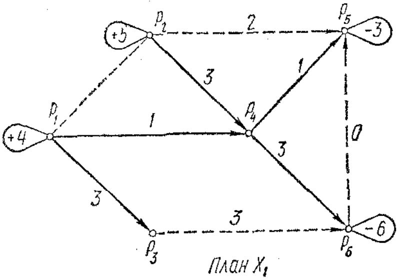


Рис. 3.17.

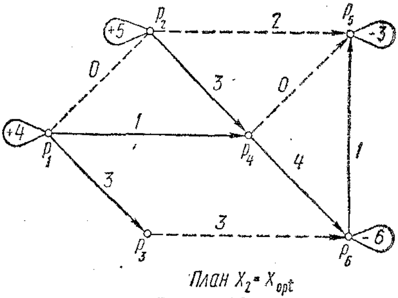


Рис. 3.18.

Проверяем условия оптимальности для всех неосновных коммуникаций:

а) для $k_{12} : u_2 - u_1 = 1 - 0 = 1 < c_{12} = 4, \quad x_{12} = 0;$

для $k_{25} : u_5 - u_2 = 10 > c_{25} = 2 \quad x_{25} = d_{25};$

для $k_{36} : u_6 - u_3 = 8 - 3 = 5 > c_{36} = 4 \quad x_{36} = d_{36};$

для $k_{65} : u_5 - u_6 = 11 - 8 = 3 > c_{56} = 2 \quad x_{65} = 0.$

Условия нарушены для коммуникации k_{65}.

Следовательно, переходим ко второму этапу.

Второй этап. Строим маршрут из p_5 в p_6. Он равен $l = \{k_{54}, k_{46}\}$. Тогда замкнутый маршрут $\gamma = \{k_{54}, k_{46}, k_{65}\}$.

Вычисляем $\theta_2 = \min \{x_{45}, d_{46} - x_{46}, d_{65}\} = \min \{1; 6—3; 3\} = 1$.

Определяем план $X^{(2)}$, изменяя поток вдоль замкнутого маршрута l на величину θ_2:

$$x_{45}^{(2)} = x_{45}^{(1)} - \theta_2 = 0; \quad x_{65}^{(2)} = \theta_2 = 1;$$
$$x_{46}^{(2)} = x_{46}^{(1)} + \theta_2 = 3 + 1 = 4.$$

Этому плану соответствуют схема перевозок и величины потоков, приведенные на рис. 3.18.

Третья итерация. Первый этап. Рассчитываем новые потенциалы, соответствующие плану $X^{(2)}$. Полагая $u_1 = 0$, получим $u_3 = 3$,

$$u_4 = u_1 + c_{14} = 5, \quad u_6 = u_4 + c_{46} = 8, \quad u_5 = u_6 + c_{65} = 10,$$
$$u_2 = u_4 - c_{24} = 5 - 4 = 1.$$

Проверим выполнение условий оптимальности (5.14) — (5.18):

$$u_2 - u_1 = 1 - 0 = 1 < c_{12} = 4, \quad x_{12} = 0;$$
$$u_6 - u_3 = 8 - 3 = 5 > c_{36} = 4, \quad x_{36} = d_{36} = 3;$$
$$u_5 - u_4 = 10 < c_{45} = 6, \quad x_{45} = 0;$$
$$u_5 - u_2 = 10 - 1 = 9 > c_{25} = 2; \quad x_{25} = d_{25}.$$

Таким образом, все условия выполняются и потому план $X^{(2)}$ оптимальный.

§ 6. АЛГОРИТМ ДЕКОМПОЗИЦИИ ДЛЯ РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ

Пусть имеется транспортная задача вида:
найти

$$\min_{x_{ij}} \sum_i \sum_j c_{ij} x_{ij} \tag{6.1}$$

при ограничениях

$$\sum_j x_{ij} = a_i, \qquad i = \overline{1, m}, \tag{6.2}$$

$$\sum_i x_{ij} = b_j, \qquad j = \overline{1, n}, \tag{6.3}$$

$$x_{ij} \geqslant 0. \tag{6.4}$$

Будем предполагать, что выполняется условие баланса $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. Если число пунктов потребления значительно превышает число пунктов производства ($n \gg m$), то появляется возможность учесть структурные особенности матрицы условий задачи и целесообразно применить метод декомпозиции для Т-задачи (он был предложен Вильямсом). При этом множество ограничений (6.3) соответствует ограничениям подзадачи, а ограничения (6.2) рассматриваются как связывающие ограничения. Следовательно, координирующая задача будет иметь только m ограничений, что значительно упрощает ее решение.

Рассмотрим основную идею метода. Пусть $x_{ij}^{(k)}$ — компоненты крайней точки множества ограничений (6.3), (6.4). Представим любое решение этой системы уравнений в виде

$$\sum_k \delta_k \cdot x_{ij}^{(k)} = x_{ij}, \tag{6.5}$$

где $\delta_k \geqslant 0$; $\sum_k \delta_k = 1$.

Подставим выражение (6.5) в (6.1) и (6.2). В результате получим следующую координирующую задачу.
Найти

$$\min \sum_k z_k \delta_k \tag{6.6}$$

при ограничениях

$$\sum_k p_{ik} \cdot \delta_k = a_i, \qquad i = \overline{1, m}, \tag{6.7}$$

$$\delta_k \geqslant 0, \tag{6.8}$$

$$\text{где } z_k = \sum_{i,j} c_{ij} x_{ij}^{(k)}, \tag{6.9}$$

$$p_{ik} = \sum_j x_{ij}^{(k)}. \tag{6.10}$$

Заметим, что условие выпуклости $\sum_k \delta_k = 1$ нет необходимости вводить в ограничения, так как оно учитывается через условие баланса. Действительно,

$$\sum_i a_i = \sum_i \sum_k \delta_k p_{ik} = \sum_k \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^{(k)} \delta_k = \sum_k \delta_k \sum_j \sum_i x_{ij}^{(k)} = \sum_k \delta_k \sum_{j=1}^{n} b_j$$

и в силу условия баланса получим $\sum_k \delta_k = 1$.

Предположим, что координирующая задача решена и найден вектор относительных оценок ограничений $\Lambda = [\lambda_i]\ i = 1, m$. Чтобы составить подзадачу, определим *симплекс-разность* для плана $X_k = \|x_{ij}^k\|$,

$$f_k = z_k - \sum_i \lambda_i p_{ik} = \sum_i \sum_j (c_{ij} - \lambda_j)\, x_{ij}^{(k)}. \tag{6.11}$$

Так как $x_{ij}^{(k)}$ — крайние точки множества (6.4)—(6.5), то нахождение $\min f_k$ равносильно решению подзадач вида

$$\min_{x_{ij}} \sum_i \sum_j (c_{ij} - \lambda_i)\, x_{ij}, \tag{6.12}$$

при ограничениях

$$x_{ij} \geqslant 0, \quad \sum_{i=1}^{m} x_{ij} = b_j \qquad j = \overline{1, n}. \tag{6.13}$$

Так как функционал (6.12) аддитивен, а каждая переменная x_{ij} находится лишь в одном из уравнений (6.13), задача (6.12) распадается на n независимых задач для каждого j

$$\min_{\{x_{ij}\}} \sum_i (c_{ij} - \lambda_i)\, x_{ij}, \tag{6.14}$$

при условиях (6.13).

Решение задачи (6.12) чрезвычайно просто, для этого необходимо найти минимальный коэффициент $(c_{ij} - \lambda_i)$ в каждом столбце. Если $\min_i (c_{ij} - \lambda_i) = c_{s_i j} - \lambda_{s_i}$,

$$\text{то } x_{ij}^{(k)} = \begin{cases} b_j & \text{при } i = s_i, \\ 0, & \text{в противном случае при } i \neq s_i, \ j = \overline{1, n}, \end{cases} \tag{6.15}$$

где $x_{ij}^{(k)}$ — компоненты оптимального решения X_k задачи (6.12).

Итак, искомое решение (6.15) характеризуется тем, что каждый пункт потребления j закрепляется за наиболее дешевым (в смысле затрат) пунктом потребления.

Учитывая (6.15), найдем *минимальную симплекс-разность* по формуле

$$f^0 = \min_{x_k} f_k = \sum_{j=1}^{n} b_j [\min_i (c_{ij} - \lambda_i)] = \sum_{j=1}^{n} b_j (c_{s_i j} - \lambda_{s_i}). \tag{6.16}$$

Если $\min_k f_k < 0$, то вектор X_k должен быть введен в базис. С этой целью находим координаты вектора $\mathbf{P}_k = [p_{ik}]$ согласно (6.10) и одновременно коэффициент z_k, и вводим вектор $\mathbf{P}_k$ в базис координирующей задачи.

Если же

$$\min_{x_k} f_k = 0, \tag{6.17}$$

то это есть признак оптимальности текущего базиса $\{\mathbf{P}_{s_1}, \mathbf{P}_{s_2}, \ldots, \mathbf{P}_{s_m}\}$ координирующей задачи, так как введением ни одного из столбцов в базис значение целевой функции уменьшить нельзя.

Допустим, что выполняется условие (6.17), $\{\delta^*_{s_1}\}_{i=\overline{1,\,m}}$ — текущее допустимое базисное решение координирующей задачи, и $\mathbf{X}_{s_i}$ — опорный план подзадачи (6.12), отвечающий переменной δ_{s_i}. Тогда оптимальное решение исходной Т-задачи (6.1)—(6.4) определится соотношением

$$\mathbf{X}_{\text{опт}} = \sum_{i=1}^{m} \mathbf{X}_{s_i} \delta^*_{s_i}. \tag{6.18}$$

Дадим теперь формальное описание алгоритма декомпозиции для Т-задачи.

Алгоритм состоит из предварительного этапа и конечного числа однотипных итераций.

На предварительном этапе определяют начальный базис координирующей задачи $\{\mathbf{P}_1^{(0)}, \mathbf{P}_2^{(0)}, \ldots, \mathbf{P}_m^{(0)}\}$ и соответствующие ему относительные оценки.

В частности, в качестве начального базиса можно выбрать систему из векторов $\{\mathbf{P}_k\}_{k=\overline{1,\,m}}$с компонентами p_{ik}, определяемыми из условия

$$p_{ik} = \begin{cases} \sum\limits_i a_i = \sum\limits_j b_j = T & \text{при } i = k, \\ 0 & \text{при } i \neq k \quad i = \overline{1,\,m}. \end{cases} \tag{6.19}$$

Такой набор векторов $\{\mathbf{P}_k\}$ соответствует опорным планам $\mathbf{X}_1, \ldots, \mathbf{X}_r \ldots, \mathbf{X}_m$ системы ограничений (6.3)—(6.4), компоненты которых вычисляются по формуле

$$x_{ij}^{(k)} = \begin{cases} b_j, & \text{при } i = k; \\ 0, & \text{в противном случае.} \end{cases} \tag{6.20}$$

Для каждого $\mathbf{P}_k$ находим коэффициент целевой функции z_k согласно (6.9). С учетом (6.20) получим окончательно

$$z_k = \sum_i \sum_j c_{ij} x_{ij}^{(k)} = \sum_j c_{kj} b_j. \tag{6.21}$$

Как известно, для базисных векторов $\mathbf{P}_k$ справедливо соотношение

$$\mathbf{P}_k^T \Lambda = \sum_{i=1}^{m} p_{ik} \lambda_i = z_k, \qquad k = \overline{1,\,m}. \tag{6.22}$$

В общем случае для нахождения искомых значений $\{\lambda_i^{(0)}\}$ необходимо решать систему уравнений (6.22). Однако в силу условия (6.19) она распадается на m независимых уравнений вида

$$\sum_{i=1}^{m} p_{ik}\lambda_i = p_{kk}\lambda_k = T\lambda_k = z_k, \tag{6.23}$$

откуда $\lambda_k = \frac{z_k}{T}$, $k = \overline{1, m}$.

На этом предварительный этап заканчивается и переходим на первую итерацию. Каждая итерация, кроме последней, состоит из двух этапов. Пусть уже проведено r итераций и найден базис координирующей задачи $\{\mathbf{P}_{si}\}$ $i = \overline{1, m}$ и соответствующие ему оценки $\lambda_i^{(r)}$. На первом этапе решается подзадача вида

$$\min_{x_{ij}} \sum_i \sum_j (c_{ij} - \lambda_i^{(r)})\, x_{ij} \tag{6.24}$$

при условии $\sum_i x_{ij} = b_j$.

Ее решение x_{ij} определяется по формулам (6.15). Найдя $\mathbf{X}_k$, соответствующее ему значение целевой функции f_k^0 вычисляем согласно (6.16). Если $f_k^0 = 0$, то текущее базисное решение координирующей задачи $\{\delta_{si}^*\}$ $i = \overline{1, m}$ оптимально, а оптимальное решение исходной задачи определяется по формуле (6.18)

$$\mathbf{X}_{\text{опт}} = \sum_i \mathbf{X}_{s_i}\delta_{s_i}^*.$$

Если $f_k^0 < 0$, то текущее д. б. р. координирующей задачи может быть улучшено и переходим ко второму этапу.

Второй этап. Определяем компоненты вектора $\mathbf{P}_k$ (соответствующего плану $\mathbf{X}_k$), который должен быть введен в базис координирующей задачи по формуле (6.10), а его коэффициент в целевой функции z_k — по формуле (6.9).

Составляем новую координирующую задачу (6.6)—(6.8), в базис которой вводим вектор $\mathbf{P}_k$ и новую переменную δ_k. Решаем ее методом обратной матрицы и определяем новый базис $\left\{\mathbf{P}_{s_i}^{(r+1)}\right\}$ $i = \overline{1, m}$, а также новые значения оценок $\left\{\lambda_i^{(r+1)}\right\}$.

На этом очередная итерация заканчивается.

В заключение еще раз подчеркнем, что преимущества метода декомпозиции по сравнению с обычными алгоритмами решения транспортной задачи проявляются при $m \ll n$.

В этом случае координирующая задача состоит только из m ограничений и решение ее нетрудоемко, что же касается решений подзадач вида (6.12), то, как было показано, их находят почти без вычислений.

В частности, решение задачи размером 15 × 3000 на ЦВМ методом декомпозиции потребовало примерно в полтора раза меньше времени в сравнении с одним из классических методов [18, 33].

Пример 3.9. Решить методом декомпозиции транспортную задачу со следующими условиями:
$a_1 = 40$, $b_1 = 25$, $b_2 = 15$, $b_3 = 18$, $b_4 = 20$, $b_5 = 12$, $b_6 = 10$, $a_2 = 60$.

$$\mathbf{C} = \begin{Vmatrix} 1 & 2 & 3 & 6 & 5 & 4 \\ 8 & 10 & 4 & 3 & 2 & 5 \end{Vmatrix}.$$

Проверяем условие баланса: $\sum\limits_i a_i = \sum\limits_j b_j = 100 = T$.

Предварительный этап. В качестве начального базиса координирующей задачи выберем

$$\{\mathbf{P}_1, \mathbf{P}_2\} = \begin{Vmatrix}100 & 0\\ 0 & 100\end{Vmatrix}.$$

Тогда вектору $\mathbf{P}_1$ отвечает начальный план $\mathbf{X}_1 = \begin{Vmatrix}25 & 15 & 18 & 20 & 12 & 10\\ 0 & 0 & 0 & 0 & 0 & 0\end{Vmatrix}$, а вектору $\mathbf{P}_2$ — план $\mathbf{X}_2 = \begin{Vmatrix}0 & 0 & 0 & 0 & 0 & 0\\ 25 & 15 & 18 & 20 & 12 & 10\end{Vmatrix}$.

Компоненты планов вычисляются согласно (6.20).

В соответствии с (6.21) определяем коэффициенты z_1 и z_2 целевой функции для векторов $\mathbf{P}_1$ и $\mathbf{P}_2$:

$$z_1 = \sum_{i=1}^{2}\sum_{j=1}^{4} c_{ij}x_{ij}^{(1)} = 329; \quad z_2 = \sum_i\sum_j c_{ij}x_{ij}^{(2)} = 556.$$

Используя соотношение (6.24), находим вектор относительных оценок условий задачи $\Lambda = [\lambda_0\lambda_2]$:

$$\lambda_1 = \frac{z_1}{T} = 3{,}29; \quad \lambda_2 = \frac{z_2}{T} = \frac{556}{100} = 5{,}56.$$

Первая итерация. Первый этап. Определяем матрицу подзадачи $\mathbf{C}_1$, используя значения λ_1 и λ_2.

$$\mathbf{C}_1 = \| c_{ij} - \lambda_i \| = \begin{Vmatrix}-2{,}29 & -1{,}29 & -0{,}29 & 2{,}71 & 1{,}71 & 0{,}71\\ 2{,}44 & 4{,}44 & -1{,}56 & -2{,}56 & -3{,}56 & -0{,}56\end{Vmatrix}.$$

Решаем подзадачу вида (6.12) при условии (6.13) с матрицей $\mathbf{C}_1$ и находим план $\mathbf{X}_3$ согласно соотношению (6.15):

$$\mathbf{X}_3 = \begin{Vmatrix}25 & 15 & 0 & 0 & 0 & 0\\ 0 & 0 & 18 & 20 & 12 & 10\end{Vmatrix}.$$

Отметим, что эта процедура сводится к нахождению минимального элемента в каждом j-м столбце матрицы $\mathbf{C}_1$, после чего в плане $\mathbf{X}_3$ полагаем соответствующий элемент равным b_j.

Находим симплекс-разность для плана $\mathbf{X}_3$:

$$f_3^0 = \sum_{j=1}^{n} b_j \min_i \, [c_{ij} - \lambda_i] = -204{,}2.$$

Так как $f_3^0 < 0$, переходим ко второму этапу.

Второй этап. Определяем компоненты $\mathbf{P}_3$ по формулам (6.10)

$$p_{13} = \sum_j x_{1j}^{(3)} = 40; \quad p_{23} = \sum_j x_{2j}^{(3)} = 60;$$

$$z_3 = \sum_i\sum_j c_{ij}x_{ij} = 1\cdot 25 + 15\cdot 2 + 18\cdot 4 + 20\cdot 3 + 12\cdot 2 + 10\cdot 5 = 261.$$

Вектор P_3 необходимо ввести в базис координирующей задачи. Эту задачу будем решать методом обратной матрицы (см. § 5, гл. 2). С этой целью определяем начальную обратную матрицу $\mathbf{B}^{-1}$

$$\mathbf{B}^{-1} = [\mathbf{P}_1, \mathbf{P}_2]^{-1} = \frac{1}{100}\begin{Vmatrix}1 & 0\\ 0 & 1\end{Vmatrix} = [\mathbf{e}_1, \mathbf{e}_2]$$

и столбец свободных членов

$$\mathbf{E}_0 = \mathbf{B}^{-1}\mathbf{A}_0 = \frac{1}{100}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}40\\ 60\end{bmatrix} = \begin{bmatrix}0{,}40\\ 0{,}60\end{bmatrix}.$$

Заполняем главную часть начальной таблицы для координирующей задачи (табл. 3.7) ($m = 2$).

Компоненты столбца P_3' находим согласно соотношениям $p_{13}' = \sum_{j=1}^{2} e_{ij}p_{j3}$ или в матричном виде: $\mathbf{P}_3' = \mathbf{B}^{-1}\,\mathbf{P}_3$.

В строке $m+1$ по столбцам e_1 и e_2 записываем λ_1, λ_2, а по столбцу $\mathbf{P}_3$: $\Delta_3 = -f_3^0 = 204{,}2$.

Выполнив одну итерацию метода, приходим к табл. 3.8.

Итак, мы нашли новый базис $\{\mathbf{P}_1, \mathbf{P}_3\}$, соответствующее ему решение $\{\delta_1^* = 0;\ \delta_2^* = 1\}$. Заметим, что оно получилось вырожденным.

Вторая итерация. Первый этап. Используя оценки, найденные в табл. 3.8, $\lambda_1^{(1)} = 3{,}29$, $\lambda_2^{(1)} = 2{,}16$, находим матрицу $\mathbf{C}_2$.:

$$\mathbf{C}_2 = \| c_{ij} - \lambda_i^{(1)} \| = \begin{bmatrix}-2{,}29 & -1{,}29 & -0{,}29 & 2{,}71 & 1{,}71 & 0{,}71\\ 5{,}84 & 7{,}84 & 1{,}84 & 0{,}84 & -0{,}16 & 2{,}84\end{bmatrix}.$$

Решаем подзадачу (6.12) с матрицей $\mathbf{C}_2$ и находим оптимальный план $\mathbf{X}_4$:

$$\mathbf{X}_4 = \begin{bmatrix}25 & 15 & 18 & 0 & 0 & 10\\ 0 & 0 & 0 & 20 & 12 & 0\end{bmatrix}.$$

Определяем $f^0(\mathrm{X}_4) = \sum_{j=1}^{6} b_j \min_i (c_{ij} - \lambda_i) = -59{,}84$.

Так как $f^0(x_4) < 0$, то вектор $\mathbf{P}_4$ необходимо ввести в базис координирующей задачи.

Второй этап. Определяем компоненты $\mathbf{P}_4$ в соответствии с (6.10):

$$\mathbf{P}_4 = \begin{bmatrix}68\\ 32\end{bmatrix}; \quad z_4 = \sum_i \sum_j c_{ij}x_{ij}^{(4)} = 233.$$

Находим компоненты разложения вектора $\mathbf{P}_4$ по векторам базиса:

$$p_{i4}^{(1)} = \sum_{j=1}^{2} e_{ij}p_{j4}, \qquad i = \overline{1,\ 2}.$$

Заполняем главную часть табл. 3.9, для чего используем табл. 3.8.

Выполнив итерацию метода Жордана — Гаусса, приходим к табл. 3.10.

Итак, новый базис $\{\mathbf{P}_4, \mathbf{P}_3\}$, а соответствующее ему решение $\delta_1 = 0$; $\delta_2 = 1{,}0$ и вектор оценок $\Lambda^{(2)} = \{\lambda_1^{(2)} = 2{,}02;\ \lambda_2^{(2)} = 3{,}0\}$.

Третья итерация. Первый этап. Определяем матрицу $\mathbf{C}_3$

$$\mathbf{C}_3 = \| c_{ij} - \lambda_i^{(2)} \| = \begin{Vmatrix}-1{,}02 & -0{,}02 & 0{,}98 & 3{,}98 & 2{,}98 & 1{,}98\\ 5{,}0 & 7{,}0 & 1{,}0 & 0 & -1{,}0 & 2{,}0\end{Vmatrix}.$$

Решаем подзадачу с матрицей $\mathbf{C}_3$ и находим план $\mathbf{X}_5$

$$\mathbf{X}_5 = \begin{Vmatrix}25 & 15 & 18 & 0 & 0 & 10\\ 0 & 0 & 0 & 20 & 12 & 0\end{Vmatrix} = \begin{Vmatrix}68\\ 32\end{Vmatrix}.$$

Вычисляем величину симплекс-разности для $\mathbf{X}_5$:

$$f^0(\mathrm{X}_5) = \sum_j b_j \min_i (c_{ij} - \lambda_i^{(2)}) = 0.$$

Таблица 3.7

N	C_X	B_X	e_0	e_1	e_2	$\mathbf{P}_3^1$	Θ
1	329	P_1	0,40	0,01		0,40	1
2	556	P_2	0,60		0,01	$\boxed{0,60}$	1
$m+1$			465,2	3,29	5,56	204,2	

Таблица 3.8

N	C_x	B_x	e_0	e_1	e_2
1	329	P_1	0	0,01	$-\frac{0,4}{60}$
2	261	P_3	1,0		$\frac{1}{60}$
			261	3,29	2,16

Таблица 3.9

N	C_x	B_x	e_0	e_1	e_2	$\mathbf{P}_4^1$	Θ
1	329	P_1	0	0,01	$-\frac{0,4}{60}$	$\boxed{0,47}$	0
2	261	P_3	1,00	—	$\frac{1}{60}$	$\frac{32}{60}$	$\frac{60}{32}$
			261	3,29	2,16	59,84	

Таблица 3.10

N	C_x	B_x	e_0	e_1	e_2
1	233	P_4	0	0,021	—0,014
2	261	P_3	1,00	—0,011	0,0242
			261	2,02	3,0

Так как $f^0(X_5) = 0$, то текущий базис координирующей задачи оптимален, $\delta_4^* = 0$, $\delta_3^* = 1$.

Оптимальный план исходной Т-задачи вычисляется по формуле (6.18)

$$X_{opt} = \delta_3^* X_3 + \delta_4^* X_4 = X_3 = \begin{Vmatrix} 25 & 15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 20 & 12 & 10 \end{Vmatrix}.$$

Вопросы для самопроверки

1. Сформулируйте условие разрешимости транспортной задачи.
2. Назовите признак оптимальности плана Т-задачи и поясните смысл его условий.
3. Что называется опорным планом Т-задачи?
4. Как проверить произвольный план на опорность?
5. Какой план называется вырожденным? Как установить, является ли опорный план вырожденным?
6. Дайте обоснование метода потенциалов, рассматривая его как вариант симплекс-метода.
7. Как решать Т-задачу методом потенциалов на максимум? Как при этом изменится правило выбора элемента x_{ij}, вводимого в базис, и признак оптимальности опорного плана?
8. В каком случае применение метода минимального элемента приводит к сокращению числа итераций по сравнению с методом северо-западного угла?
9. Почему при решении Т-задачи венгерским методом цепочка в матрице X строится обязательно из нулевых элементов?
10. Как изменится предварительный этап венгерского метода при решении Т-задачи на максимум?
11. В чем состоят особенности венгерского метода по сравнению с методом потенциалов?
12. Могут ли элементы c_{ij} оказаться отрицательными в процессе решения Т-задачи венгерским методом?
13. Дайте обоснование венгерского метода для решения Т-задачи.
14. Как свести открытую модель Т-задачи к закрытой?
15. Сформулируйте основные свойства потока на сети.
16. Как записывается условие баланса для транспортной сети?
17. Сформулируйте теорему Форда—Фалкерсона и докажите ее справедливость.
18. Сформулируйте признак оптимальности алгебраического плана транспортной сети, сравните его с соответствующим признаком для Т-задачи.
19. Признак опорности алгебраического плана транспортной сети.
20. Как найти начальный опорный план для сети?
21. Когда целесообразно использование декомпозиционного метода для решения транспортных задач?

Глава 4. ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

Многие задачи исследования операций, такие как распределения ресурсов, сетевого планирования и управления, календарного планирования, описываются математическими моделями дискретного программирования.

Рассмотрим общую задачу максимизации:
найти

$$\max f(x_1, x_2, \ldots, x_n), \tag{1}$$

при условиях

$$\begin{aligned} &g_1(x_1, x_2, \ldots, x_n) \leqslant 0, \\ &\ldots\ldots\ldots\ldots\ldots \\ &g_m(x_1, x_2, \ldots, x_n) \leqslant 0; \end{aligned} \tag{2}$$

и

$$X = (x_1, x_2, \ldots, x_n) \in D, \tag{3}$$

где D — некоторое множество.

Если множество D является конечным или счетным, то условие (3) — это условие дискретности, и данная задача является задачей *дискретного программирования.*

Чаще всего условие дискретности разделено по отдельным переменным следующим образом:

$$x_j \in D_j, \quad j = 1, 2, \ldots, n, \tag{4}$$

где D_j — конечное (или счетное) множество.

Если вводится ограничение x_j — целые числа ($j = 1, 2, \ldots, n$), то приходят к задачам целочисленного программирования (ЦП), которое является частным случаем дискретного программирования.

В задачах дискретного программирования область допустимых решений является невыпуклой и несвязной. Поэтому отыскание решения таких задач сопряжено со значительными трудностями. В частности, невозможно применение стандартных приемов, состоящих в замене дискретной задачи ее непрерывным аналогом, в дальнейшем округлении найденного решения до ближайшего целочисленного. Например, рассмотрим следующую задачу ЛП:

Пример 4.1. Найти $\max(x_1 - 3x_2 + 3x_3)$

$$\begin{cases} 2x_1 + x_2 - x_3 \leqslant 4; \\ 4x_1 - 3x_2 \leqslant 2; \\ -3x_1 + 2x_2 + x_3 \leqslant 3, \end{cases}$$

где $x_1, x_2, x_3 \geqslant 0$, x_j — целые числа ($j = 1, 2, 3$).

Игнорируя условие целочисленности, находим оптимальный план симплекс-методом:

$$x_{1\,\text{опт}}=\frac{1}{2}, \quad x_{2\,\text{опт}}=0, \quad x_{3\,\text{опт}}=4\frac{1}{2}.$$

Проверка показывает, что никакое округление компонент этого плана не дает допустимого решения, удовлетворяющего ограничениям этой задачи.

Искомое целочисленное решение задачи таково:

$$x_{1\,\text{опт}}=2; \quad x_{2\,\text{опт}}=2; \quad x_{3\,\text{опт}}=5.$$

Таким образом, для решения задач дискретного программирования необходимы специальные методы.

Методы решения задач дискретного программирования по принципу подхода к проблеме делят на 3 группы: 1) методы отсечения или отсекающих плоскостей; 2) метод ветвей и границ; 3) метод случайного поиска и приближенные эвристические методы [28].

§ 1. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

По структуре математической модели задачи дискретного программирования разделяют на следующие классы: 1) задачи с неделимостями; 2) экстремальные комбинаторные задачи; 3) задачи на несвязных и на невыпуклых областях; 4) задачи с разрывными целевыми функциями.

Задачи с неделимостями

Математические модели задач с неделимостями основаны на требовании целочисленности переменных $\{x_i\}$, вытекающем из физических условий практических задач.

К таким задачам относится задача об определении оптимальной структуры производственной программы, где $\{x_1, x_2, \ldots, x_n\}$ — объемы выпуска продукции.

Эта задача заключается в отыскании

$$\max_{\{x_i\}} \sum_{j=1}^{n} c_j x_j, \tag{1.1}$$

при

$$\sum_{j=1}^{n} a_{ij} x_i \leqslant b_i, \qquad (i = 1, 2, \ldots, m), \tag{1.2}$$

$$x_1 \geqslant 0, \; x_2 \geqslant 0, \; \ldots, \; x_n \geqslant 0; \quad x_j \text{ — целые при } j \in J. \tag{1.3}$$

Если $J = N = (1, 2, \ldots, n)$, то задача называется полностью целочисленной, в противном случае, если $J \neq N$ — частично целочисленной.

Задача о ранце. Одной из наиболее распространенных задач целочисленного программирования является так называемая задача о ранце.

Постановка задачи. Турист готовится к длительному переходу в горах. В рюкзаке он может нести груз, масса которого не более W *кг*. Этот груз может включать n видов предметов, каждый предмет типа j, массой wj *кг*, $j = 1, 2, \ldots, n$. Для каждого вида предмета турист определяет его ценность E_j во время перехода. Сколько предметов каждого типа он должен положить в рюкзак, чтобы суммарная ценность снаряжения была максимальной?

Обозначим через x_j — количество предметов j-го типа в рюкзаке. Тогда математическая модель задачи такова:

$$\max \sum_{j=1}^{n} Ex_j \tag{1.4}$$

при ограничениях

$$\sum_{j=1}^{n} w_j x_j \leqslant W, \quad x_j \text{ — целое}, \tag{1.5}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n.$$

Экстремальные комбинаторные задачи

В данных задачах необходимо найти экстремум некоторой целевой функции, заданной на конечном множестве, элементами которого служат перестановки из n символов (объектов).

Одной из наиболее простых задач этого класса является *задача о назначениях:*

найти такую перестановку $(p_1, p_2, \ldots, p_n)$ из чисел $1, 2, 3, \ldots, n$, при которой достигается $\min \sum\limits_{i=1}^{n} c_{ip_i}$ по всем перестановкам $(p_1, p_2, \ldots$ $\ldots, p_n)$.

Каждая такая перестановка может быть представлена точкой в n^2-мерном эвклидовом пространстве или в виде матрицы $X^{n \times n} = \| x_{ij} \|$.

Вводим переменные: $\begin{cases} x_{ij} = 1, \text{ если } i\text{-й механизм предназначен для } j\text{-й работы;} \\ x_{ij} = 0 \text{ в противном случае.} \end{cases}$

Очевидно, $\sum\limits_{i=1}^{n} x_{ij} = 1; \quad j = 1, 2, \ldots, n,$

$$\sum_{j=1}^{n} x_{ij} = 1; \qquad i = 1, 2, \ldots, n. \tag{1.6}$$

Тогда задача заключается в нахождении таких чисел $\{x_{ij}\}$, при которых достигается $\min \Sigma\Sigma\, c_{ij} x_{ij}$ при ограничениях (1.6).

Хотя условия целочисленности $\{x_{ij}\}$ в ограничениях (1.6) в явном виде нет, оно выполняется, поскольку задача о назначениях является частным случаем Т-задачи.

Задача о коммивояжере. Имеется $(n + 1)$ город. Задана матрица $C = \| c_{ij} \|$ расстояний между городами. Выезжая из исходного города A_0, коммивояжер должен побывать во всех остальных городах по одному разу и вернуться в город A_0.

Определить, в каком порядке следует объезжать города, чтобы суммарное пройденное расстояние было минимально.

Введем переменные $x_{ij} = \begin{cases} 1, \text{ если коммивояжер переезжает из } A_i \text{ в } A_j; \\ 0 \text{ в противном случае.} \end{cases}$

Математическая модель задачи имеет следующий вид: найти

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} \tag{1.7}$$

при условиях

$$\sum_{j=0}^{n} x_{ij} = 1 \qquad i = 1,\ 2,\ \ldots,\ n; \tag{1.8}$$

$$\sum_{i=0}^{n} x_{ij} = 1 \qquad j = 1,\ 2,\ \ldots,\ n \tag{1.9}$$

$$u_i - u_j + n x_{ij} \leqslant n - 1 \qquad (i,\ j = 1,\ 2,\ \ldots,\ n,\ i \neq j), \tag{1.10}$$

где u_i, u_j — произвольные целые и неотрицательные числа.

Условие (1.8) означает, что коммивояжер выезжает из каждого города один раз, а условие (1.9) — что он въезжает один раз в каждый город, кроме начального.

Если ограничить задачу только условиями (1.8) и (1.9), то она эквивалентна задаче о назначениях, план которой не обязан быть цикличным. Иначе говоря, путь коммивояжера при этом можно представить как ряд несвязанных подциклов, в то время как его путь в действительности состоит из одного цикла.

Покажем, что для любого цикла, начинающего в A_0, можно найти u_i, удовлетворяющие условию (1.10). Пусть $u_i = p$, если коммивояжер посещает город A_i на p-м этапе. Отсюда следует, что $u_i - u_j \leqslant n - 1$ для всех i и j, и, таким образом, условие (1.10) выполняется при $x_{ij} = 0$.

При $x_{ij} = 1$ условие (1.10) выполняется как строгое равенство:

$$u_i - u_j + n x_{ij} = p - (p + 1) + n = n - 1.$$

Задачи о покрытии относятся к классу экстремальных комбинаторных задач на графах.

Типичная задача о покрытии состоит в следующем. Дан граф Г. Требуется найти его минимальное покрытие, т. е. такую минимальную совокупность ребер, чтобы любая вершина графа была инцидентна некоторому ребру, входящему в покрытие.

Обозначим вершины графа i ($i = 1,\ 2,\ \ldots,\ m$), а ребра j ($j = 1,\ 2,\ \ldots,\ n$).

Граф характеризуется матрицей инциденций вершин и ребер $A = \| a_{ij} \|$, где

$$a_{ij} = \begin{cases} 1, \text{ если вершина } i \text{ инцидентна ребру } j; \\ 0, \text{ в противном случае.} \end{cases}$$

Вводим набор переменных $\{x_j\}$ таких, что

$$x_j = \begin{cases} 1, & \text{если ребро } j \text{ входит в покрытие;} \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда нахождение минимального покрытия эквивалентно следующей задаче:

найти

$$\min \sum_{j=1}^{n} x_j \tag{1.11}$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant 1, \tag{1.12}$$

$$i = 1,\ 2,\ \dots\ m.$$

Задачи на несвязных и невыпуклых областях

Данные задачи представляют собой модели линейного программирования, в которых к обычным условиям присоединены некоторые дополнительные условия, превращающие область допустимых решений в невыпуклую или несвязную.

Пусть к задаче ЛП присоединено условие $x_i x_{i+1} = 0$ или более общее условие $\sum\limits_{i=0}^{p} x_i x_{i+1} = 0$ $(p \leqslant n - 1)$. На первый взгляд кажется, что добавление этих условий превращает задачу в нелинейную. В действительности, этими условиями допустимое множество решений разбивается на несвязные области.

В частном случае при дополнительных условиях: $x_1 x_2 = 0$, $x_1 + x_2 \geqslant 1$ получаем несвязную область, приведенную на рис. 4.1.

Задачи на несвязных областях. Пусть переменная x_{j_0} задачи ЛП имеет ограничение сверху и, кроме того, ограничение:

$$\text{либо } x_{j_0} \leqslant a, \quad \text{либо } x_{j_0} \geqslant b, \tag{1.13}$$

где $0 \leqslant a < b \leqslant k_{j_0}$.

Задачи такого типа с дополнительными логическими условиями вида «или-или» будем называть дихотомическими.

Вводим целочисленную переменную y_{j_0}, принимающую значения 0 и 1. Дополнительная переменная y_{j_0} в целевую функцию не включается.

Тогда

$$\begin{cases} x_{j_0} - b + b y_{j_0} \geqslant 0; \\ -x_{j_0} + a y_{j_0} + k_{j_0} (1 - y_{j_0}) \geqslant 0; \\ y_{j_0} = \begin{cases} 0, \\ 1. \end{cases} \end{cases} \tag{1.14}$$

Система (1.14) эквивалентна ограничениям (1.13). Если $y_{j_0} = 1$, то $0 \leqslant x_j \leqslant a$. Если $y_{j_0} = 0$, то $x_{j_0} \geqslant b$; $x_{j_0} \leqslant k_{j_0}$.

Рассмотрим более общий случай. Пусть в задачах математического программирования с допустимой областью решений G имеется альтернативное ограничение:

$$\text{либо } h(x_1, x_2, \ldots, x_n) \geqslant 0, \text{ либо } k(x_1, x_2, \ldots, x_n) \geqslant 0, \tag{1.15}$$

где $h(x_1, x_2, \ldots, x_n)$ и $k(x_1, x_2, \ldots, x_n)$ — заданные функции.

Предположим, что нам известны нижние границы функций $h(X)$ и $k(X)$ на области G, обозначаемые соответственно $h_{\min}$ и $k_{\min}$.

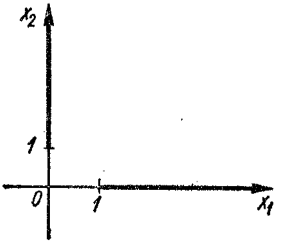


Рис.4.1.

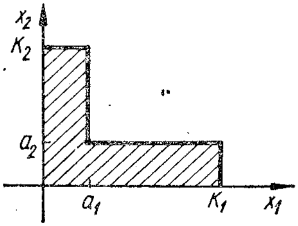


Рис. 4.2.

Введем вспомогательную переменную $y = \begin{cases} 0 \\ 1 \end{cases}$ и рассмотрим систему неравенств:

$$\begin{cases} h(x_1, x_2, \ldots, x_n) - h_{\min} y \geqslant 0, \\ k(x_1, x_2, \ldots, x_n) - k_{\min}(1 - y) \geqslant 0. \end{cases} \tag{1.16}$$

Эта система эквивалентна альтернативному ограничению (1.15).

Итак, данная задача при введении вспомогательной переменной становится задачей дискретного программирования.

Задачи на невыпуклых областях. Введение дополнительных дискретных переменных позволяет осуществить переход от задачи оптимизации на невыпуклой области к задаче на области, представляющей собой сумму выпуклых множеств.

Пусть имеется задача ЛП с дополнительным ограничением (рис. 4.2):

$$x_1 \leqslant k_1, \quad x_2 \leqslant k_2, \tag{1.17}$$

причем либо $x_1 \leqslant a_1$, либо $x_2 \leqslant a_2$. (1.18)

Введя переменную y, можно охарактеризовать эту область системой неравенств:

$$\begin{cases} 0 \leqslant x_1 \leqslant a_1 + (k_1 - a_1) y; \\ 0 \leqslant x_2 \leqslant k_2 - (k_2 - a_2) y; \\ y = \begin{cases} 0, \\ 1. \end{cases} \end{cases} \tag{1.19}$$

В более общем случае при решении задач может быть задана система множеств (T_1, T'_1), (T_2, T'_2), ..., (T_p, T'_p), причем решение принадлежит либо $\bigcap\limits_{k=1}^{p} T_k$, либо $\bigcap\limits_{k=1}^{p} T'_k$.

Пусть для каждой пары множеств (T_i, T'_i) области T_i и T'_i описываются соответственно системами неравенств:

$$\left.\begin{array}{l} g_{1i}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0, \\ g_{2i}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0, \\ \dots\dots\dots\dots\dots \\ g_{ki}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0; \end{array}\right\} \quad (1.20) \qquad \left.\begin{array}{l} h_{1i}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0, \\ h_{2i}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0, \\ \dots\dots\dots\dots\dots \\ h_{ki}(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0. \end{array}\right\} \quad (1.21)$$

Кроме того, пусть известны числа $\{g_{ki\,\min}\}$, $\{h_{ki\,\min}\}$ — нижние границы для функций g_{ki} $(x_1, x_2, \dots, x_n)$ и h_{ki} $(x_1, x_2, \dots, x_n)$ соответственно.

Вводим дополнительную целочисленную переменную $y = \{0,\ 1\}$. Тогда альтернативные ограничения:

$$\left.\begin{array}{l} g_{1i}(x_1,\ x_2,\ \dots,\ x_n - g_{1i\,\min} y \geqslant 0, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ g_{ki}(x_1,\ x_2,\ \dots,\ x_n) - g_{ki\,\min} y \geqslant 0, \\ h_{1i}(x_1,\ x_2,\ \dots,\ x_n) - h_{1i\,\min}(1 - y) \geqslant 0, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ h_{ki}(x_1,\ x_2,\ \dots,\ x_n) - h_{ki\,\min}(1 - y) \geqslant 0. \\ y = \begin{cases} 0, \\ 1. \end{cases} \end{array}\right\} \quad (1.22)$$

Теперь рассмотрим случай условных или логических ограничений.

Пусть имеется логическое ограничение:

$$\text{если} \quad h(x_1,\ x_2,\ \dots,\ x_n) > 0, \quad \text{то} \quad k(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0. \qquad (1.23)$$

Запишем его в эквивалентном виде:

$$\left.\begin{array}{l} а)\ \text{либо}\ h(x_1,\ x_2,\ \dots,\ x_n) > 0, \quad k(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0; \\ б)\ \text{либо}\ h(x_1,\ x_2,\ \dots,\ x_n) \leqslant 0. \end{array}\right. \qquad (1.24)$$

Введя переменную $y = \{0,\ 1\}$, приходим к следующей системе

$$\begin{array}{l} h(x_1,\ x_2,\ \dots,\ x_n) > h_{\min} y; \\ k(x_1,\ x_2,\ \dots,\ x_n) \geqslant k_{\min} y; \\ h(x_1,\ x_2,\ \dots,\ x_n) \leqslant h^*(1 - y), \end{array} \qquad (1.25)$$

где $h^* = \sup\limits_G h$, $h_{\min} = \inf\limits_G \{h\ (x_1, x_2, \dots, x_n)\}$, $k_{\min} = \inf\limits_G \{k\ (x_1, x_2, \dots$ $\dots, x_n)\}$, G — область определения h $(x_1, x_2, \dots, x_n)$ и k $(x_1, x_2, \dots, x_n)$.

Рассмотрим другой случай ограничений.

$$\left.\begin{array}{l} \text{Если}\ h(x_1,\ x_2,\ \dots,\ {}_n) \geqslant 0, \ \text{то}\ k(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0; \\ \text{если}\ h(x_1,\ x_2,\ \dots,\ x_n) < 0, \ \text{то}\ l(x_1,\ x_2,\ \dots,\ x_n) \geqslant 0. \end{array}\right. \qquad (1.26)$$

Введем переменные y_1, y_2 и получим

$$\left.\begin{aligned} &h_{\min} y_1 \leqslant h(x_1, x_2, \ldots, x_n) < h^* y_2; \\ &k(x_1, x_2, \ldots, x_n) \geqslant k_{\min} y_1; \\ &l(x_1, x_2, \ldots, x_n) \geqslant l_{\min} y_2; \\ &y_1 + y_2 = 1, \end{aligned}\right\} \tag{1.27}$$

где $y_i = \begin{cases} 0, \\ 1 \end{cases} \quad (i = 1,\ 2).$

Таким образом, при введении логических переменных задачи с логическими ограничениями сводятся к задачам дискретного программирования.

Задачи с разрывными целевыми функциями

Наиболее изучена из этого класса задач Т-*задача* с *фиксированными доплатами* [37]:
найти

$$\min \sum_i \sum_j c_{ii}(x_{ij})$$

при условиях

$$\sum_{j=1}^{n} x_{ij} = a_i; \quad \sum_{i=1}^{m} x_{ij} = b_j; \quad c_{ij}(x_{ij}) = \begin{cases} 0, & \text{если } x_{ij} = 0 \\ c_{ij} x_{ij} + d_{ij}, & \text{если } x_{ij} > 0, \end{cases} \tag{1.28}$$

где c_{ij} — транспортные издержки по перевозке единицы груза;
d_{ij} — фиксированная доплата за аренду транспортных средств.

Вводим вспомогательные переменные y_{ij} следующим образом:

$$y_{ij} = \begin{cases} 0, \\ 1, \end{cases} \quad x_{ij} \leqslant M_{ij} y_{ij}, \tag{1.29}$$

где $M_{ij} = \min\ \{a_i,\ b_j\}$.

Тогда целевая функция имеет вид

$$\sum_i \sum_j c_{ij} x_{ij} + d_{ij} y_{ij}. \tag{1.30}$$

Задача (1.30) эквивалентна исходной.

Действительно, при $y_{ij} = 0$ переменные $x_{ij} = 0$; а при $y_{ij} = 1$ неравенства (1.29) становятся несущественными, так как в любом опорном плане они выполняются. Таким образом, эта задача является задачей частично-целочисленного программирования.

Теперь рассмотрим более общую математическую модель с разрывной целевой функцией на примере задачи о смесях.

Примем следующие обозначения: a_{ij} — содержание элемента i в единице компонента j; b_i — ограничение снизу на содержание каждого элемента в смеси; c_j — стоимость закупки единицы компонента, d_j — фиксированная стоимость заказа j-го компонента; x_j — количество j-го компонента смеси.

Требуется составить наиболее дешевую смесь, удовлетворяющую ограничениям по содержанию каждого компонента.

Формально задача сводится к минимизации $\sum_{j=1}^{n} c_j(x_j)$ при условиях

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i \qquad (i = 1,\ 2,\ \dots,\ m), \tag{1.31}$$

где

$$x_j \geqslant 0,$$

$$c_j(x_j) = \begin{cases} 0 & \text{при } x_j = 0, \\ c_jx_j + d_j & \text{при } x_j > 0. \end{cases} \tag{1.32}$$

Предположим, что в дополнение к условиям (1.31) задана для x_j верхняя граница:

$$x_j \leqslant k_j \qquad (j = 1,\ 2,\ \dots,\ n). \tag{1.33}$$

Тогда запишем задачу минимизации в следующем виде: найти

$$\min \sum_j (c_jx_j + d_jy_j) \tag{1.34}$$

при условии (1.31) и дополнительных условиях

$$y_j = \left\{\begin{matrix} 0, \\ 1 \end{matrix}\right. \quad (0 \leqslant x_j \leqslant k_jy_j).$$

Эквивалентность частично-целочисленной задачи (1.34) и исходной задачи может быть проверена аналогично Т-задаче с фиксированными доплатами.

Задачи, сводящиеся к целочисленным

Некоторые задачи, формально не являющиеся целочисленными, имеют целочисленные решения при любых целочисленных исходных данных. К таким задачам относится Т-задача, которая формально не является целочисленной, так как отсутствует условие: $\{x_{ij}\}$ — целые числа.

При анализе решений Т-задачи Данцигом установлена следующая теорема: при любых целых значениях a_1, a_2, ..., a_m и b_1, b_2, ..., b_n Т-задача всегда имеет целочисленный оптимальный план, независимо от значений $\{c_{ij}\}$.

Доказательство теоремы основано на двух положениях:

существует исходный опорный целочисленный план;

при переходе от одного опорного плана к другому целочисленность сохраняется.

Действительно, из этих положений сразу следует целочисленность всех опорных планов, и в частности оптимального.

Целочисленность планов Т-задачи связана со структурой ее матрицы ограничений и имеет место для всех ее частных случаев, важнейшими из которых являются задача о назначениях и задача о потоке на сети.

Таким образом, Т-задача при целочисленных условиях сводится к задаче ЦП.

§ 2. МЕТОД ОТСЕКАЮЩИХ ПЛОСКОСТЕЙ[1]

Для группы методов отсечения, или отсекающих плоскостей, характерна так называемая «регуляризация» задачи, состоящая в погружении исходной области допустимых решений в объемлющую ее выпуклую область, т. е. во временном отбрасывании условий дискретности. Затем к получившейся регулярной задаче применяются стандартные методы оптимизации. Если полученное в результате решение удовлетворяет требуемым условиям дискретности, то задача решена. В противном случае необходим дальнейший переход к целочисленному решению, причем этот переход не может быть получен простым округлением компонент нецелочисленного решения. Впервые идея такого перехода описана в работах Данцига, Фулкерсона и Джонсона.

Для задачи целочисленного программирования (ЦП) соответствующая идея, положенная в основу метода отсекающих плоскостей, высказана Данцигом.

Допустим необходимо решить задачу ЦП:
найти

$$\max \sum_{j=1}^{n} c_j x_j \tag{2.1}$$

при условиях

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad (i = 1,\ 2,\ \ldots,\ m); \tag{2.2}$$

$$x_j \geqslant 0, \qquad x_j \text{ — целые числа } (j = 1,\ 2,\ \ldots,\ n). \tag{2.3}$$

Отбросив на время условие целочисленности, найдем оптимальный опорный план. Если окажется, что он удовлетворяет также и условиям целочисленности, то данный план является искомым.

В противном случае нужно сформировать *дополнительное ограничение, которому заведомо удовлетворяет любой целочисленный план, но не удовлетворяет найденный оптимальный нецелочисленный план.* Такое ограничение называют *правильным отсечением.* Геометрически правильное отсечение означает гиперплоскость, отсекающую от выпуклого многогранника (или многогранного множества) соответствующей задачи ЛП некоторый многогранник, содержащий все целочисленные планы.

Таким образом, исходную систему ограничений дополняют правильным отсечением, после чего разыскивают оптимальное решение новой ЛП-задачи.

Если дополнительное ограничение сформировано удовлетворительно. то можно полагать, что через несколько итераций будет найдено искомое решение задачи ЦП либо будет зафиксирована несовместность ее условий.

Рассмотрим задачу (2.1), (2.2), (2.3) в предположении, что все $\{a_{ij}\}$, $\{b_i\}$ и $\{c_j\}$ — целые числа.

---

[1] Метод отсечения, или отсекающих плоскостей, разработан Р. Гомори в 1957—1958 гг.

Пусть $\{A_1, A_2, \ldots, A_m\}$ — линейно независимые векторы ограничений. Очевидно, матрица $A = [A_1, A_2, \ldots, A_m]$ невырождена, и поэтому существует обратная матрица $A^{-1} = [A_1, A_2, \ldots, A_m]^{-1}$.

Умножив ограничения задачи $\sum_{j=1}^{n} A_j x_j = b$ на матрицу A^{-1}, получим

$$\sum_{j=1}^{n} A^{-1} A_j x_j = A^{-1} b = X_0. \tag{2.4}$$

Это соотношение в развернутой форме записи имеет вид:

$$x_i + \sum_{j=m+1}^{n} x_{ij} x_j = x_{i0} \qquad (i = 1, \ldots, m), \tag{2.5}$$

откуда

$$x_i = x_{i0} - \sum_{j=m+1}^{n} x_{ij} x_j, \tag{2.6}$$

где x_i — базисная переменная, а x_{i0} определяют из соотношения (2.4).

Подставляя выражение (2.6) в (2.1), получим:

$$L(x) = \sum_{i=1}^{m} \left(x_{i0} - \sum_{j=m+1}^{n} x_{ij} x_j \right) c_i + \sum_{j=m+1}^{n} c_j x_j = \sum_{i=1}^{m} c_i x_{i0} -$$
$$- \sum_{j=m+1}^{n} \left(\sum_{i=1}^{m} c_i x_{ij} - c_j \right) x_j = \Delta_0 - \sum_{j=m+1}^{n} \Delta_j x_j. \tag{2.7}$$

Теперь решим задачу (2.1), (2.2), (2.3) симплекс-методом или двойственным симплекс-методом.

Допустим, что оптимальным оказался базис $[A_{s_1}, A_{s_2}, \ldots, A_{s_m}]$. Элементы последней таблицы обозначим через x_{ij}, причем $\Delta_j = x_{0j}$, а столбец свободных членов обозначим через x_{i0}. Выберем один из них, например, x_{i0}, и, начиная с i-й строки последней таблицы, составим дополнительное ограничение.

Вводим обозначения: $[x_{ij}]$ — целая часть, а $\gamma_{ij} = x_{ij} - [x_{ij}]$ — дробная часть числа x_{ij}, причем $x_{ij} \geqslant [x_{ij}]$.

Пусть X — произвольный план задачи (2.1), (2.2), (2.3). Его компоненты связаны уравнением (2.6). Подставим $x_{ij} = [x_{ij}] + \gamma_{ij}$ в выражение (2.6) и после несложных преобразований получим:

$$x_i - [x_{i0}] + \sum_{j=m+1}^{n} [x_{ij}] x_j = \gamma_{i0} - \sum_{j=m+1}^{n} \gamma_{ij} x_j. \tag{2.8}$$

Соотношение (2.8) справедливо для любого плана данной задачи.

Допустим, что X — целочисленный план. Тогда, учитывая целочисленность левой части соотношения (2.8), имеем $\gamma_{i0} - \sum_{j=m+1}^{n} \gamma_{ij} x_j$ — целое число или $\sum_{j=m+1}^{n} \gamma_{ij} x_j = \gamma_{i0} + \xi$, где ξ — целое число.

Очевидно, дробная часть $0 \leqslant \gamma_{ij} < 1$, $0 \leqslant \gamma_{i0} < 1$. Следовательно, $\sum\limits_{j=m+1}^{n} \gamma_{ij}x_j \geqslant 0$. Если допустить, что целое число $\xi \leqslant -1$, то получаем неравенство $\sum\limits_{j=m+1}^{n} \gamma_{ij}x_j + 1 \leqslant \gamma_{i0}$, откуда $\gamma_{i0} \geqslant 1$, а это противоречит условию $0 \leqslant \gamma_{i0} < 1$. Поэтому принимаем ξ неотрицательным числом, т. е. признаем неравенство

$$\sum_{j=m+1}^{n} \gamma_{ij}x_j \geqslant \gamma_{i0}. \qquad (2.9)$$

Нецелочисленный план **X** не удовлетворяет неравенству (2.9), поскольку $\gamma_{i0} > 0$, а левая часть неравенства (2.9) равна нулю, так как $x_j = 0$ для всех небазисных компонент $j = m + 1, m + 2, \ldots, n$. В то же время любой целочисленный план удовлетворяет (2.9) как строгому равенству, так как $\gamma_{i0} = 0$ для всех i.

Итак, ограничение (2.9) является правильным отсечением. Его записывают в эквивалентной форме

$$x_{n+1} = -\gamma_{i0} + \sum_{j=m+1}^{n} \gamma_{ij}x_j, \quad x_{n+1} \geqslant 0, \qquad (2.10)$$

или

$$-\gamma_{i0} = \sum_{j=m+1}^{n} -\gamma_{ij}x_j + x_{n+1}, \qquad (2.11)$$

и добавляют к системе линейных ограничений задачи. После этого отыскивают новый оптимальный план, используя двойственный симплекс-метод.

Переменные новой задачи $\{x_1, x_2, \ldots, x_{n+1}\}$. Ее условия уже разрешены относительно базисных переменных плана $\mathbf{X} = \{x_1, x_2, \ldots \ldots, x_m\}$ и новой переменной x_{n+1}, и, следовательно, имеем псевдоплан с базисными компонентами:

$$x_i = x_{i0} \qquad (i = 1, \ldots, m); \quad x_{n+1} = -\gamma_{i0}.$$

Таблица данного псевдоплана образуется дописыванием к таблице, отвечающей найденному плану **X**; строки с элементами

$$x_{n+1,j} = \begin{cases} -\gamma_{i0} \text{ при } j = 0, \\ -\gamma_{ij} \text{ при } j \in J, \text{ (где } J \text{ — множество индексов} \\ \qquad\qquad\qquad \text{небазисных векторов),} \\ 0 \qquad\qquad\quad \text{в остальных случаях.} \end{cases} \qquad (2.12)$$

Одновременно к таблице добавляют единичный вектор $\mathbf{A}_{n+1}$ такой, что $\mathbf{A}_{n+1}^{T} = [\underbrace{0,\ 0,\ \ldots,\ 0}_{n},\ 1]$.

К этому псевдоплану применяют двойственный симплекс-метод. На первой итерации из базиса обязательно выводят вектор, отвечающий переменной x_{n+1}, так как в столбце a_{i0} таблицы имеется один отрицательный элемент $x_{n+1} = -\gamma_{i0} < 0$.

Условимся называть *большой итерацией метода отсекающих плоскостей* несколько итераций алгоритма двойственного симплекс-метода, приводящих от псевдоплана с дробными компонентами в столбце a_{i0} к последующей оптимальной таблице.

В зависимости от исхода большой итерации различают следующие три случая:

1. В столбце a_{i0} все x_{i0} — целые числа, причем $x_{i0} \geqslant 0$. Это условие определяет оптимальность найденного плана.

2. Получен новый план, где все $x_{i0} \geqslant 0$, но не все x_{i0} — целые числа. Тогда необходимо сформировать новое правильное отсечение и перейти к очередной большой итерации.

3. Получен некоторый промежуточный псевдоплан, где имеется элемент $x_{i0} < 0$ такой, что $x_{ij} \geqslant 0$ для всех $j = 1, 2, \ldots, n$. Это признак неразрешимости задачи.

Переменные x_{n+t}, вводимые в задачу в начале каждой большой итерации, называются дополнительными, а переменные x_j $(1 \leqslant j \leqslant n)$ — основными. Если дополнительная переменная является небазисной для некоторого промежуточного псевдоплана, то уравнение $x_{n+t} = 0$ входит в число соотношений, определяющих этот псевдоплан. Как только переменная x_{n+t} снова становится базисной, ее значение оказывается безразличным для основных переменных. Поэтому строку и столбец, отвечающие x_{n+t}, в соответствующей таблице вычеркивают.

С геометрической точки зрения это правило можно обосновать так. Если псевдоплан оказывается внутри полупространства $x_{n+t} \geqslant 0$, то дополнительное ограничение, определяемое гиперплоскостью $x_{n+t} = 0$, становится несущественным и потому опускается.

Двойственный симплекс-метод является основой для метода Гомори, так как он позволяет учитывать новые дополнительные ограничения (правильные отсечения) и переходить от текущего псевдоплана к новому оптимальному плану.

Замечания.

1. При определенных условиях можно гарантировать конечность алгоритма Гомори.

Достаточные условия конечности алгоритма Гомори установлены в следующей теореме [18]:

Если целевая функция задачи $L = \sum_{j=1}^{n} c_j x_j$ *ограничена снизу и сверху на множестве решений, причем множество оптимальных решений представляет собой выпуклый многогранник, и правильное отсечение образуется по строке таблицы с нецелочисленной компонентой* x_{i0}, *имеющей минимальный номер, а выбор вектора, вводимого в базис, определяют по величине* $\Theta_k = \min\limits_{j} \left\{ -\frac{\Delta_j}{x_{ij}} \bigg|_{x_{ij}<0} \right\}$, *то алгоритм Гомори является конечным.*

2. *Правильное отсечение можно формировать и по индексной строке.*

Пусть на k-м шаге элементы индексной строки таковы: $\Delta_0^k, \Delta_1^k, \ldots$ $\ldots, \Delta_n^k$, причем $\Delta_0^k = L(X_k)$ — нецелое число. Тогда обозначим γ_{0j} —

дробную часть элемента Δ_j: $\gamma_{0j} = \Delta_j - [\Delta\ ]$, $(j = 0, 1, \ldots, n)$. В этом случае правильное отсечение записывается так:

$$-\gamma_{00} = -\sum_{j\in J} \gamma_{0j} x_j + x_{n+1}. \qquad (2.13)$$

3. В отличие от обычной ЛП-задачи задача целочисленного программирования требует большого объема вычислений даже при малых m и n. Количество итераций существенно зависит от того, насколько удачно сформированы правильные отсечения.

Пример 4.2. Решить методом Гомори следующую задачу: найти

$$\max\,(x_1 + x_2)$$

при ограничениях

$$\begin{cases} 2x_1 + 11x_2 \leqslant 38; \\ x_1 + x_2 \leqslant 7; \\ 4x_1 - 5x_2 \leqslant 5, \end{cases}$$

$$0 \leqslant x_1,\ y_2;\quad x_1,\ x_2 \text{ — целые числа.}$$

Для данной задачи при отброшенном условии целочисленности оптимальный нецелочисленный план найден двойственным симплекс-методом (пример 2.10). Здесь это решение задано в виде табл. 4.1.

Образуем правильное отсечение по строке A_1 $(i = 1)$:

$$-\gamma_{i0} = x_{n+1} - \sum_{i\in J} \gamma_{i\in j_j}.$$

Допишем строку A_6 и столбец A_6 к данной таблице. Направляющая строка A_6, направляющий столбец A_5.

Выполнив один шаг двойственного симплекс-метода, приходим к табл. 4.2.

Найдя в столбце A_0 отрицательную компоненту $x_{30} = -3 < 0$, определим направляющую строку A_3. Направляющий столбец A_4. Выполнив один шаг двойственного симплекс-метода, приходим к табл. 4.3.

Так как в этой таблице все элементы $x_{i0} \geqslant 0$, то первая большая итерация закончена, и найден новый оптимальный, но нецелочисленный план. Поэтому формируем правильное отсечение по индексной строке.

Добавим к табл. 4.3 строку A_7 и столбец A_7. Выбрав направляющий элемент $a_{73} = -\dfrac{1}{11}$ и выполнив очередной шаг, приходим к новой табл. 4.4.

Так как в столбце A_0 имеется отрицательный элемент $x_{50} = -1$, то выполним следующую итерацию с направляющим элементом $x_{56} = -9$. Вектор A_6 должен быть снова введен в базис, но так как переменная x_6 была использована для формирования предыдущего правильного отсечения, то это позволяет после соответствующего симплекс-преобразования вычеркнуть (или опустить) столбец A_6 и строку A_6 в следующей новой табл. 4.5.

Закончилась вторая большая итерация. Так как в столбце имеются дробные компоненты, то формируем следующее новое правильное отсечение (строка A_8) по строке A_1. Направляющий элемент $a_{85} = -\dfrac{1}{9}$. Приходим к табл. 4.6.

Выполним очередной шаг двойственного симплекс-метода с направляющим элементом $a_{37} = -11$. Вектор A_7 нужно снова ввести в базис. При этом соответствующую строку и столбец вычеркивают и опускают в табл. 4.7.

Так как полученный план нецелочисленный, то формируем новое правильное отсечение по индексной строке, дописывая к табл.4.7 строку A_9 и столбец A_9. Направляющий элемент $a_{93} = -\dfrac{1}{11}$. Выполнив один шаг двойственного симплекс-метода, приходим к табл. 4.8.

Таблица 4.1

								↓	
	c_i			1	1	0	0	0	0
		B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	1	A_1	$\frac{40}{9}$	1	0	0	$^{5}/_{9}$	$^{1}/_{9}$	0
	0	A_3	1	0	0	1	—6	1	0
	1	A_2	$^{23}/_{9}$	0	1	0	$^{4}/_{9}$	$-^{1}/_{9}$	0
		Δ	7	0	0	0	1	0	0
←		A_6	$-^{4}/_{9}$	0	0	0	$-^{5}/_{9}$	$-^{1}/_{9}$	1

Таблица 4.2

							↓		
	c			1	1	0	0	0	0
		B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	1	A_1	4	1	0	0	0	0	1
←	0	A_3	—3	0	0	1	$\boxed{-11}$	0	9
	1	A_2	3	0	1	0	1	0	—1
		A_5	4	0	0	0	5	1	—9
		Δ	7	0	0	0	1	0	0

c_i			1
	B_x	A_0	A_1
1	A_1	4	1
0	A_4	$^{3}/_{11}$	0
1	A_2	$^{30}/_{11}$	0
	A_3	$^{29}/_{11}$	0

c_i			1
	B_x	A_0	A_1
1	A_1	$^{35}/_{9}$	1
0	A_4	1	0
1	A_2	$^{19}/_{9}$	0
	A_3	6	0

c_i			1
	B_x	A_0	A_1
1	A_1	3	1
0	A_4	$^{12}/_{11}$	0
1	A_2	$^{32}/_{11}$	0
	A_5	$^{83}/_{11}$	0

В этой таблице получен иско

x_1

($x_{3\,\text{опт}} = 10$; $x_{4\,\text{опт}} = 2$; $x_{5\,\text{опт}} =$
Целевая функция $L_{\max} = (x$

Анализ отсечений

Эффективность метода о
насколько удачно работает
мы получили правило постр
ределяемое формулой 2.11.
ным и могут быть построе
тивных отсечений, которые
нию, важно уметь определ
связаны с данным оптимальн
Используя результаты
жем, что множество отсечен

Следовательно, для любого целочисленного α на основании (2.15) и (2.16) можно написать

$$\alpha \mathbf{x}_B = \alpha \mathbf{B}^{-1}\mathbf{b} - \alpha \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \text{целое число}. \tag{2.17}$$

Используя очевидное соотношение $x = [x] + \{x\}$, получим, что

$$[\alpha \mathbf{B}^{-1}\mathbf{b}] + \{\alpha \mathbf{B}^{-1}\mathbf{b}\} - [\alpha \mathbf{B}^{-1}\mathbf{N}]\,\mathbf{x}_N - \{\alpha \mathbf{B}^{-1}\mathbf{N}\}\,\mathbf{x}_N = \text{целое число}. \tag{2.18}$$

Учитывая, что по предположению $[\alpha \mathbf{B}^{-1}\ \mathbf{b}]$, $[\alpha \mathbf{B}^{-1}\ \mathbf{N}]$ являются матрицами с целочисленными компонентами, а $\mathbf{x}_N$ — целочисленный вектор, получим, что дробные части выражения (2.17) должны удовлетворять требованию

$$\{\alpha \mathbf{B}^{-1}\mathbf{b}\} - \{\alpha \mathbf{B}^{-1}\mathbf{N}\}\,\mathbf{x}_N = \xi = \text{целое число}. \tag{2.19}$$

По определению дробной части вещественного числа имеем

$$0 \leqslant \{\alpha \mathbf{B}^{-1}\mathbf{b}\} < 1, \quad \text{а также } 0 \leqslant \{\alpha \mathbf{B}^{-1}\mathbf{N}\} < 1.$$

Поэтому с учетом неотрицательности вектора $\mathbf{x}_N$ заключаем, что целое число ξ в (2.19) должно быть неположительным, так как в противном случае при $\xi \geqslant 1$ мы бы получили $\{\alpha \mathbf{B}^{-1}\mathbf{b}\} \geqslant 1 + \{\alpha \mathbf{B}^{-1}\mathbf{N}\}\,\mathbf{x}_N$, что противоречит определению дробной части.

Таким образом, общее уравнение отсечений по Гомори имеет вид

$$\{\alpha \mathbf{B}^{-1}\mathbf{b}\} - \{\alpha \mathbf{B}^{-1}\mathbf{N}\}\,\mathbf{x}_N \leqslant 0. \tag{2.20}$$

Вводя обозначения $\mathbf{B}^{-1}\mathbf{b} = \| x_{i0} \|\ i \in I_б$, $\mathbf{B}^{-1}\mathbf{N} = \| a_{ij} \|$, $i \in I_б$, $j \in J_{неб}$ и подставляя в (2.20), получим следующее общее выражение для отсечений по Гомори

$$\left\{\sum_{i \in I_б} \alpha_i x_{i0}\right\} - \sum_{j \in J_{неб}} \left\{\sum_i \alpha_i a_{ij}\right\} x_j \leqslant 0. \tag{2.21}$$

Эта формула обобщает выражение (2.11).

Два различных отсечения по Гомори соответствуют двум различным векторам α_1 и α_2. Выбор компонентов вектора α произволен.

Возьмем для иллюстрации отсечение из табл. 4.1 для $\alpha_1 = [1\ 0\ 0]$. Из выражения (2.21) имеем

$$\left\{1 \cdot \frac{40}{9}\right\} - \left\{1 \cdot \frac{5}{9}\right\} x_4 - \left\{1 \cdot \frac{1}{9}\right\} x_5 \leqslant 0 \to \frac{4}{9} - \left(\frac{5}{9} x_4 + \frac{1}{9} x_5\right) \leqslant 0. \tag{2.22}$$

Вычислим теперь отсечение, соответствующее $\alpha_2 = [4\ 0\ 0]$. Получим

$$\left\{4 \cdot \frac{40}{9}\right\} - \left\{4 \cdot \frac{5}{9}\right\} x_4 - \left\{4 \cdot \frac{1}{9}\right\} \leqslant 0,$$

или

$$\frac{7}{9} - \frac{2}{9} x_4 - \frac{4}{9} x_5 \leqslant 0. \tag{2.23}$$

Сравнивая (2.23) с (2.22), убеждаемся, что, исходя из одного и того же ограничения и изменяя лишь вектор α, можно получать различные отсечения. Это объясняется тем, что отсечения по Гомори образуют группу, которая часто оказывается циклической. В этом случае все отсечения, входящие в одну и ту же циклическую группу, могут быть порождены одним и тем же ограничением.

Покажем, что отсечения, определяемые соотношением (2.20), образуют абелеву группу по отношению к вводимой ниже операции [1].

Обозначим через I отдельное отсечение по Гомори или соответствующее ему неравенство:

$$I_i : \{\alpha B^{-1} b\} \leqslant \{\alpha B^{-1} N\} x_N \qquad (2.24)$$

и определим операцию $*$ на множестве I отсечений по Гомори следующим образом.

Пусть

$$I_1 : \{\alpha_1 B^{-1} b\} \leqslant \{\alpha_1 B^{-1} N\} x_N, \qquad (2.25)$$

$$I_2 : \{\alpha_2 B^{-1} b\} \leqslant \{\alpha_2 B^{-1} N\} x_N. \qquad (2.26)$$

Тогда

$$I_1 * I_2 : \{(\alpha_1 + \alpha_2) B^{-1} b\} \leqslant \{(\alpha_1 + \alpha_2) B^{-1} N\} x_N. \qquad (2.27)$$

Таким образом, реализация операции $*$ есть построение нового отсечения (2.27), осуществляемое введением арифметической суммы векторов α_1 и α_2.

Покажем, что множество отсечений I по отношению к операции $*$ образует абелеву группу. Для этого достаточно показать, что множество векторов $A = \{\alpha_j\}$ образует группу по арифметическому сложению. Для этого нужно доказать у множества A существование следующих свойств: замкнутости, существование единицы, ассоциативности, обратного элемента и коммутативности.

1. *Замкнутость*. Множество A замкнуто по арифметическому сложению, так как если $\alpha_i^{(1)} \in Z$ и $\alpha_i^{(2)} \in Z$, то и $\alpha_i^{(1)} + \alpha_i^{(2)} \in Z$, $i = \overline{1, m}$, где Z — множество относительных целых чисел.

2. *Существование единицы*. Единицей в A по арифметическому сложению является элемент $0 = [\underbrace{0,\ 0,\ \ldots,\ 0}_{m}]$.

Единицей в множестве I будет отсечение

$$0 \leqslant 0^T x_N.$$

3. *Ассоциативность*. Если $\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)} \in Z$, $i = \overline{1, m}$, то $(\alpha_i^{(1)} + \alpha_i^{(2)}) + \alpha_i^{(3)} = \alpha_i^{(1)} + (\alpha_i^{(2)} + \alpha_i^{(3)})$ и $(\alpha^{(1)} + \alpha^{(2)}) + \alpha^{(3)} = \alpha^{(1)} + (\alpha^{(2)} + \alpha^{(3)})$. Отсюда же $(I_1 * I_2) * I_3 = I_1 * (I_2 * I_3)$.

4. *Существование обратного элемента*. Имеем для каждого $\alpha_i \in Z$ существует такой единственный β_i, что $\alpha_i + \beta_i = \beta_i + \alpha_i = 0$ $i = 1, m$, откуда $\beta_i = -\alpha_i$.

---

1) Определение и свойства абелевой группы рассмотрены в приложении 3.

Таким образом, каждому вектору $\alpha^1 \in A$ соответствует вектор $[-\alpha^1]$, который является его единственным обратным элементом. Тогда можно найти отсечение I_1^{-1}, обратное к I_1.

Имеем

$$I_1 : \{\alpha_1 B^{-1} b\} \leqslant \{\alpha_1 B^{-1} N\} x_N ;$$

$$I_1^{-1} : \{-\alpha_1 B^{-1} b\} \leqslant \{-\alpha_1 B^{-1} N\} x_N.$$

5. *Коммутативность.* Имеем для всех $\alpha_i^{(1)}, \alpha_i^{(2)} \in Z$; $\alpha_i^{(1)} + \alpha_i^{(2)} = \alpha_i^{(2)} + \alpha_i^{(1)}$, $i = \overline{1, m}$

и, следовательно,

$$\alpha^{(1)} + \alpha^{(2)} = \alpha^{(2)} + \alpha^{(1)}, \ \forall \alpha^{(1)}, \alpha^{(2)} \in A.$$

Откуда

$$I_1 * I_2 = I_2 * I_1.$$

Таким образом $(A, +)$ и $(I, *)$, представляют собой гомоморфные по отношению друг к другу абелевы группы.

Число элементов в группе отсечений по Гомори

Используя полученные выше результаты, найдем число различных отсечений типа (2.21), которые можно построить из всех следующих векторов

$$[\alpha] = [\alpha_1, \alpha_2, \ \ldots, \ \alpha_m] \in Z^m. \qquad (2.28)$$

Как известно (см. приложение 2), любая матрица $B^{(m \times m)}$ может быть приведена к приведенной нормальной форме Смита $\Delta^{(m \times m)}$

$$U^{(m \times m)} B V = \Delta^{(m \times m)}, \qquad (2.29)$$

где $U^{(m \times m)}$, $V^{(m \times m)}$ — невырожденные унимодулярные матрицы. Заметим, что если матрица $B^{(m \times m)}$ невырожденная, то и матрица $\Delta^{(m \times m)}$ будет также невырождена. В этом случае матрица $\Delta^{(m \times m)}$ в приведенной форме Смита будет иметь вид

$$\Delta^{m \times m} = \begin{bmatrix} \delta_1 & 0 & \ldots & 0 \\ 0 & \delta_2 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & \delta_n \end{bmatrix}, \qquad (2.30)$$

причем δ_i является делителем δ_{i+1}.

Умножая выражение (2.29) слева на U^{-1}, а справа на V^{-1}, получим

$$B = U^{-1} \Delta V^{-1}. \qquad (2.31)$$

Найдем обратную матрицу для матрицы B:

$$B^{-1} = V \Delta^{-1} U, \qquad (2.32)$$

где

$$\Delta^{-1} = \begin{bmatrix} 1/\delta_1 & 0 & \dots & 0 \\ 0 & 1/\delta_2 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 1/\delta_m \end{bmatrix}. \tag{2.33}$$

Подставляя выражение (2.32) в (2.20), получим

$$\{\alpha \mathbf{V}\Delta^{-1}\mathbf{U}\mathbf{b}\} \leqslant \{\alpha \mathbf{V}\Delta^{-1}\mathbf{U}\mathbf{N}\}\mathbf{x}_N. \tag{2.34}$$

Заметим, что матрица $\mathbf{V}$, будучи унимодулярной, является целочисленной. Тогда

$$\beta^{(1\times m)} = \alpha^{(1\times m)}\mathbf{V}^{(m\times m)} \tag{2.35}$$

представляет собой матрицу-строку из целочисленных компонент.

Подставляя (2.35) в (2.34), получаем

$$\{\beta\Delta^{-1}\mathbf{U}\mathbf{b}\} \leqslant \{\beta\Delta^{-1}\mathbf{U}\mathbf{N}\}\mathbf{x}_N. \tag{2.36}$$

Положим

$$\mathbf{U}^{(m\times m)}\mathbf{b}^{(m\times 1)} = \mathbf{T}^{(m\times 1)}, \tag{2.37}$$

$$\mathbf{U}^{(m\times m)}\mathbf{N}^{(m\times n)} = \mathbf{S}^{(m\times n)}, \tag{2.38}$$

и подставим выражения (2.37), (2.38) в соотношение (2.36). Получим

$$\{\beta\Delta^{-1}\mathbf{T}\} \leqslant \{\beta\Delta^{-1}\mathbf{S}\}\mathbf{x}_N^{(n\times 1)}. \tag{2.39}$$

Обозначим также

$$\mu = \beta^{(1\times m)}\Delta^{-1} = [\beta_1/\delta_1,\ \beta_2/\delta_2,\ \dots,\ \beta_m/\delta_m]. \tag{2.40}$$

Тогда выражение (2.39) примет вид

$$\{\mu\mathbf{T}\} \leqslant \{\mu\mathbf{S}\}\mathbf{x}_N. \tag{2.41}$$

Заметим, что $\mathbf{T}^{(m\times 1)}$ и $\mathbf{S}^{(m\times n)}$ являются матрицами с целочисленными элементами.

Воспользуемся следующим легко проверяемым свойством сложения по модулю 1:

если b — целое число, то $\{ab\} = \{\{a\}b\}$. (2.42)

На основании соотношения (2.42) неравенство (2.41) может быть записано в следующем эквивалентном виде:

$$\{\{\mu\}\mathbf{T}\} \leqslant \{\{\mu\}\mathbf{S}\}\mathbf{x}_N. \tag{2.43}$$

Максимально возможное число отсечений соответствует числу различных отсечений (2.43), которые можно получить для каждого вектора

$$\{\mu\} = \{\beta\Delta^{-1}\}. \tag{2.44}$$

Другими словами, если имеется δ_1 значений величины $\{\mu_1\} = \{\beta_1/\delta_1\}$, δ_2 значений величины $\{\mu_2\} = \{\beta_2/\delta_2\}$ и, наконец, δ_m значений величины $\{\mu_m\} = \{\beta_m/\delta_m\}$, то вектор $\{\beta\Delta^{-1}\}$ может иметь лишь $\delta_1 \times \times \delta_2 \times \ldots \times \delta_m$ различных значений и если исключить вектор $\{0\}$, то существует $|\delta_1 \cdot \delta_2 \cdot \ldots \cdot \delta_m| - 1$ различных ненулевых векторов $\{\beta\Delta^{-1}\}$.

Рассмотрим выражение (2.29). Матрица $\Delta^{(m\times m)}$ — диагональная и невырожденная. Ее детерминант равен

$$\det \Delta = \delta_1\delta_2 \ldots \delta_m. \tag{2.45}$$

Но поскольку $\det U = \pm 1$ и $\det V = \pm 1$, то $|\det \Delta| = |\det B|$.

Таким образом, число возможных отсечений по Гомори равно

$$\nu = |\det B| - 1. \tag{2.46}$$

Циклическая группа отсечений *. Если в матрице Δ $\delta_1 = \delta_2 = \ldots = \delta_m - 1 = 1$, то все множество векторов $\mu^{(m\times 1)}$ может быть записано в виде

$$\{\mu\} = [0\ 0\ \ldots\ \{\beta_m/\delta_m\}]. \tag{2.47}$$

В этом случае общее выражение (2.41) отсечений по Гомори примет вид

$$\{\{\beta_m/\delta_m\}\ t_m\} \leqslant \{\{\beta_m/\delta_m\} s_{m_1}\} x_{N_1} + \\ + \{\{\beta_m/\delta_m\} s_{m_2}\} x_{N_2} + \ldots + \{\{\beta_m/\delta_m\} s_{m_n}\} x_{N_n}. \tag{2.48}$$

Все отсечения (2.47) получают последовательным применением операции сложения по модулю 1, начиная со следующего отсечения при $\beta_m = 1$

$$\{t_m/\delta_m\} \leqslant \{s_{m_1}/\delta_m\} x_{N_1} + \{s_{m_2}/\delta_m\} x_{N_2} + \ldots + \{s_{m_n}/\delta_m\} x_{N_n}. \tag{2.49}$$

Отсечения, определяемые (2.48), образуют *циклическую группу* *, для которой отсечение (2.49) представляет собой порождающий элемент.

Пример 4.3. Найдем полное множество отсечений по Гомори для оптимального нецелочисленного плана, приведенного в табл. 4.1.

1. Базис, отвечающий оптимальному решению $(\overset{*}{x}_1, \overset{*}{x}_2, \overset{*}{x}_3)$, равен

$$B = [A_1A_2A_3] = \begin{bmatrix} 2 & 11 & 1 \\ 1 & 1 & 0 \\ 4 & -5 & 0 \end{bmatrix}.$$

Матрицу B необходимо привести к приведенной форме Смита. С этой целью умножим B на матрицу P_{13} справа, что приводит к перестановке первого и третьего столбцов, и получим

$$B' = BP_{13} = \begin{bmatrix} 2 & 11 & 1 \\ 1 & 1 & 0 \\ 4 & -5 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 11 & 2 \\ 0 & 1 & 1 \\ 0 & -5 & 4 \end{bmatrix}.$$

Далее, умножая B′ справа на матрицу вычитаний V_1, получим

$$D_1 = BP_{13}V_1 = \begin{bmatrix} 1 & 11 & 2 \\ 0 & 1 & 1 \\ 0 & -5 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & -11 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -5 & 4 \end{bmatrix}.$$

---

* Определение циклической группы см. в приложении 3.

Умножая D_1 слева на матрицу вычитаний U_1, а справа на матрицу V_2, получим приведенную форму Смита

$$D_2 = U_1 D_1 V_2 = \overset{U_1}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix}} \times \overset{D_1}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -5 & 4 \end{bmatrix}} \times \overset{V_2}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{bmatrix} = \Delta.$$

Таким образом $\delta_1 = 1$, $\delta_2 = 1$, $\delta_3 = 9$.

$$U = U_1, \quad V = P_{13} V_1 V_2.$$

Максимальное число отсечений, которые можно получить для решения в табл. 4.1, равно:

$$\nu = |\det B| - 1 = |\det \Delta| - 1 = 9 - 1 = 8.$$

2. Для построения полного множества отсечений находим вектор $y^{(m \times 1)}$ и матрицу $S^{(m \times (n-m))}$

$$y = Ub = U_1 b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix} \times \begin{bmatrix} 38 \\ 7 \\ 5 \end{bmatrix} = \begin{bmatrix} 38 \\ 7 \\ 40 \end{bmatrix}, \qquad (1)$$

$$S = UN = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 5 & 1 \end{bmatrix}, \qquad (2)$$

где $N = [A_4, A_5]$ — матрица для небазисных переменных x_4 и x_5 в исходных ограничениях примера 4.2.

Так как $\delta_1 = 1$, $\delta_2 = 1$, а $\delta_3 = 9$, то вектор μ будет равен

$$\mu = \left[\left\{ \frac{\beta_1}{\delta_1} \right\}; \ \left\{ \frac{\beta_2}{\delta_2} \right\}; \ \left\{ \frac{\beta_3}{\delta_3} \right\} \right] = \left[0 \ 0 \ \left\{ \frac{\beta_3}{9} \right\} \right],$$

где β_3 — произвольный целочисленный скаляр.

Таким образом, варьируя β_3, мы находим 8 ненулевых векторов μ:

$$\mu_1 = [0 \ 0 \ 1/9]; \quad \mu_2 = [0 \ 0 \ 2/9]; \quad \mu_3 = [0 \ 0 \ 3/9]; \quad \mu_4 = [0 \ 0 \ 4/9],$$
$$\mu_5 = [0 \ 0 \ 5/9]; \quad \mu_6 = [0 \ 0 \ 6/9]; \quad \mu_7 = [0 \ 0 \ 7/9]; \quad \mu_8 = [0 \ 0 \ 8/9]. \qquad (3)$$

Подставляя значения μ, а также выражения (1) и (2) в уравнение отсечений по Гомори вида

$$\{\{\mu\} y\} \leqslant \{\{\mu\} S\} x_N$$

и учитывая, что $x_N = [x_4, x_5]$, получим следующую полную группу из 8 отсечений по Гомори:

а) для $\mu = \mu_1$

$$\{40 \cdot 1/9\} \leqslant \{1/9 \cdot 5\} x_4 + \{1/9 \cdot 1\} x_5$$

или $4/9 \leqslant 5/9\, x_4 + 1/9\, x_5$;

б) для $\mu = \mu_2$:

$$\{40 \cdot 2/9\} \leqslant \{2/9 \cdot 5\} x_4 + \{2/9 \cdot 1\} x_5 \rightarrow 8/9 \leqslant 1/9 x_4 + 2/9 x_5;$$

в) $\mu = \mu_3$:

$$\left\{ 40 \cdot \frac{3}{9} \right\} \leqslant \left\{ \frac{3}{9} \cdot 5 \right\} x_4 + \left\{ \frac{3}{9} \cdot 1 \right\} x_5 \rightarrow \frac{3}{9} \leqslant \frac{6}{9} x_4 + \frac{3}{9} x_5;$$

г) $\mu = \mu_4$:

$$\left\{ 40 \cdot \frac{4}{9} \right\} \leqslant \left\{ \frac{4}{9} \cdot 5 \right\} x_4 + \left\{ \frac{4}{9} \cdot 1 \right\} x_5 \rightarrow \frac{7}{9} \leqslant \frac{2}{9} x_4 + \frac{4}{9} x_5;$$

г) $\mu = \mu_5$:

$$\left\{40 \cdot \frac{5}{9}\right\} \leqslant \left\{\frac{5}{9} \cdot 5\right\} x_4 + \left\{\frac{5}{9} \cdot 1\right\} x_5 \to \frac{2}{9} \leqslant \frac{7}{9} x_4 + \frac{5}{9} x_5;$$

е) $\mu = \mu_6$:

$$\left\{40 \cdot \frac{6}{9}\right\} \leqslant \left\{\frac{6}{9} \cdot 5\right\} x_4 + \left\{\frac{6}{9} \cdot 1\right\} x_5 \to \frac{6}{9} \leqslant \frac{3}{9} x_4 + \frac{6}{9} x_5;$$

ж) $\mu = \mu_7$:

$$\left\{40 \cdot \frac{7}{9}\right\} \leqslant \left\{\frac{7}{9} \cdot 5\right\} x_4 + \left\{\frac{7}{9} \cdot 1\right\} x_5 \to \frac{1}{9} \leqslant \frac{8}{9} x_4 + \frac{7}{9} x_5;$$

з) $\mu = \mu_8$:

$$\left\{40 \cdot \frac{8}{9}\right\} \leqslant \left\{\frac{8}{9} \cdot 5\right\} x_4 + \left\{\frac{8}{9} \cdot 1\right\} x_5 \to \frac{5}{9} \leqslant \frac{4}{9} x_4 + \frac{8}{9} x_5.$$

Заметим, что последнее отсечение совпадает с отсечением, которое формируется по строке A_2 в табл. 4.1.

§ 3. АСИМПТОТИЧЕСКОЕ ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

Формулировка асимптотической задачи

В последние годы усилия математиков были направлены на разработку данных, более эффективных методов линейного целочисленного программирования (ЛЦП).

Одним из таких методов является так называемый метод *асимптотического целочисленного* программирования, предложенный Гомори [32].

Рассмотрим задачу ЛЦП вида:

найти

$$\max_{\mathbf{x}} f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}, \tag{3.1}$$

$$\mathbf{A}^{(m \times n)} \mathbf{x} \leqslant \mathbf{b}, \tag{3.2}$$

$$x_j \geqslant 0, \quad j = 1, n, \tag{3.3}$$

$$x_j \text{ — целые.} \tag{3.4}$$

Отбросив на время условие целочисленности (3.4), найдем оптимальное решение задачи ЛП.

Обозначим через $\mathbf{B}^{(m \times m)}$ — базис оптимального решения задачи; $\mathbf{x}_B$ — вектор базисных переменных оптимального решения; $\mathbf{x}_N$ — вектор небазисных переменных; $\mathbf{N}$ — матрица ограничений, соответствующая переменным $\mathbf{x}_N$; $\mathbf{c}_B$, $\mathbf{c}_N$ — компоненты вектора $\mathbf{c}$, отвечающие переменным $\mathbf{x}_B$ и $\mathbf{x}_N$ соответственно.

Используя соотношение

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \tag{3.5}$$

и подставляя его в (3.3), получим

$$\max \mathbf{c}^T \mathbf{x} = \max (\mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N) = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} - (-\mathbf{c}_N^T + \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N})\,\mathbf{x}_N, \tag{3.6}$$

при условиях

$$\mathbf{x}_N \geqslant 0, \tag{3.7}$$

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N. \tag{3.8}$$

Если составляющими вектора $x_B^{(m\times 1)}$ являются целые числа, то он представляет собой оптимальное решение задачи (3.1)—(3.4). В противном случае применяют метод асимптотического программирования, излагаемый ниже.

Отметим, что этот метод не гарантирует получения оптимального решения во всех без исключения случаях.

Задача *асимптотического программирования,* порождаемая задачей целочисленного программирования, формулируется следующим образом:

$$\min g(\mathbf{x}) = (-\mathbf{c}_N^T + \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N})\, \mathbf{x}_N; \quad (3.9)$$

$$\mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \mathbf{B}^{-1}\mathbf{b}; \quad (3.10)$$

$$\mathbf{x}_N \geqslant \mathbf{0}; \quad (3.11)$$

$$\mathbf{x}_B \in Z^m, \ \mathbf{x}_N \in Z^n, \quad (3.12)$$

где Z — множество относительных целых чисел.

Модель (3.9)—(3.12) отличается от модели (3.1)—(3.4) тем, что в ней составляющие вектора $\mathrm{X_B}$ могут принимать и отрицательные значения.

Представим теперь задачу (3.9)—(3.12) в другой форме, положив для упрощения

$$\tilde{\mathbf{c}}_N^T = -\mathbf{c}_N^T + \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N}. \quad (3.13)$$

В этих обозначениях задача (3.9) приобретает вид

$$\min \tilde{\mathbf{c}}_N^T \mathbf{x}_N; \quad (3.14)$$

$$\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \geqslant 0; \quad (3.15)$$

$$\mathbf{x}_N^{(n\times 1)} \in \mathbf{H}^{(n)}, \text{ где } \mathbf{H} = \{0, 1, 2 \ \ldots\}. \quad (3.16)$$

Пусть $[\mathbf{x}_N/\mathbf{b}]$ — оптимальное решение задачи (3.14)—(3.16). Это решение остается неизменным для всех векторов $\mathbf{b}$ и $\mathbf{b}^1$ таких, что

$$\{\mathbf{B}^{-1}\mathbf{b}\} = \{\mathbf{B}^{-1}\mathbf{b}^1\}, \quad (3.17)$$

где знаком $\{\mathbf{x}\}$ обозначена дробная часть $\mathbf{x}$.

Это решение назовем *периодическим.*

Положив

$$\mathbf{x}^{(m+n)\times 1} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix},$$

можем записать решение асимптотической задачи (3.9—3.12) в следующем виде:

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_B^* \\ \mathbf{x}_N^* \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ 0^{(n\times 1)} \end{bmatrix} + \begin{bmatrix} -\mathbf{B}^{-1}\mathbf{N} \\ [1] \end{bmatrix} \times [\mathbf{x}_N/\mathbf{b}]. \quad (3.18)$$

Из соотношения (3.18) видно, что решение задачи асимптотического программирования есть сумма решения ЛП-задачи (первый член) и периодического члена (второй член).

Решение асимптотической задачи

Воспользуемся приведенной формой Смита Δ для матрицы $\mathbf{B}$. Если матрица $\mathbf{B}$ — невырожденная, то справедливо соотношение:

$$\mathbf{B}^{-1} = \mathbf{V}\Delta^{-1}\mathbf{U}. \tag{3.19}$$

Справедливо следующее, легко проверяемое свойство сложения по модулю 1: если b является целым числом, то

$$\{ab\} = \{\{a\}b\}. \tag{3.20}$$

Так как по условию $\mathbf{x}_N$ — целое, используя соотношение (3.20), ограничение (3.15) можно записать в виде:

$$\{\mathbf{B}^{-1}\mathbf{b}\} = \{\{\mathbf{B}^{-1}\mathbf{N}\}\mathbf{x}_N\}. \tag{3.21}$$

Учитывая, что матрица $\mathbf{V}$ состоит из относительных целых чисел, и используя свойство (3.20) после подстановки (3.19) в (3.21), получим

$$\{\Delta^{-1}\mathbf{U}\mathbf{b}\} = \{\{\Delta^{-1}\mathbf{U}\mathbf{N}\}\mathbf{x}_N\}. \tag{3.22}$$

Используя обозначение $\mathbf{U}\mathbf{b} = \mathbf{T} = [t_j]$ и $\mathbf{U}\mathbf{N} = \mathbf{S}$ и учитывая, что

$$\Delta^{-1} = \begin{bmatrix} 1/\delta_1 & 0 & \dots & 0 \\ 0 & 1/\delta_2 & \dots & 0 \\ 0 & 0 & \dots & 1/\delta_m \end{bmatrix},$$

получим из соотношения (3.22)

$$\left\{\frac{t_i}{\delta_i}\right\} = \left\{\frac{\mathbf{S}_i^{(1\times n)}}{\delta_i}\right\}\mathbf{x}_N^{(n\times 1)}\Big\} \quad i = 1 \dots m-1, \tag{3.23}$$

$$\left\{\frac{t_m}{\delta_m}\right\} = \left\{\left\{\frac{1}{\delta_m}\mathbf{S}_m\right\}\mathbf{x}_N\right\}, \tag{3.24}$$

где t_i — i-я компонента вектора $\mathbf{T}$; S_i — i-я строка матрицы $\mathbf{S}^{(m\times n)}$. Допустим, что матрица Δ такова, что $\delta_1 = \delta_2 = \dots = \delta_{m-1} = 1$. В этом случае, учитывая, что элементы матриц $\mathbf{T}$ и $\mathbf{S}$ так же, как и составляющие вектора $\mathbf{x}_N$, представляют собой относительные целые числа, нетрудно убедиться, что условия (3.23) будут выполняться для всех x_N, удовлетворяющих условию целочисленности.

Таким образом, асимптотическая задача (3.9) может быть упрощена и в случае, когда группа отсечений по Гомори является циклической (см. приложение 3), ее можно записать в виде:

$$\min g(\mathbf{x}_N) = \min \tilde{\mathbf{C}}_N^T\mathbf{x}_N = \tilde{c}_1x_1 + \tilde{c}_2x_2 + \dots + \tilde{c}_nx_n, \tag{3.25}$$

при условиях

$$\left\{\frac{t_m}{\delta_m}\right\} = \{\{1/\delta_m \cdot \mathbf{S}_m\}\{\mathbf{x}_N\}, \tag{3.26}$$

$$\mathbf{x}_N \in \mathbf{H}^n, \quad \mathbf{H} = \{0,1,2\dots\}. \tag{3.27}$$

Задача (3.25) при условиях (3.26)—(3.27) представляет собой известную задачу о ранце, которая рассмотрена в § 1 данной главы. Ограничение (3.26) имеет особый вид

$$\left\{ \frac{a_1}{\delta_m} x_1 + \frac{a_2}{\delta_m} x_2 + \ldots + \frac{a_n}{\delta_m} x_n \right\} = \left\{ \frac{b}{\delta_m} \right\}, \tag{3.28}$$

где все a_i и b — целые.

Это уравнение называют *уравнением по модулю* 1 [32].

Если оно имеет хотя бы одно допустимое решение, то оно допускает бесчисленное множество решений.

Для решения асимптотической задачи (3.25)—(3.27) можно применить метод динамического программирования.

Рассмотрим основные идеи этого метода.

Запишем задачу (3.25)—(3.27) в следующем виде:

$$\min g(\mathbf{x}_n) = \tilde{c}_1 x_1 + \tilde{c}_2 x_2 + \ldots + \tilde{c}_n x_n, \tag{3.29}$$

при условиях

$$\left\{ \frac{a_1}{\delta} x_1 + \frac{a_2}{\delta} x_2 + \ldots + \frac{a_n}{\delta} x_n \right\} = \left\{ \frac{b}{\delta} \right\}, \tag{3.30}$$

где $\delta \in \mathbf{H}$, $x_i \in \mathbf{H}$, $a_i < \delta$, $i = \overline{1, n}$.

Пусть $\Lambda_{n-1}(\xi)$ представляет собой оптимальное решение следующей задачи:

$$\Lambda_{n-1}(\xi) = \min (\tilde{c}_1 x_1 + \tilde{c}_2 x_2 + \ldots + \tilde{c}_{n-1} x_{n-1}), \tag{3.31}$$

при условиях

$$\left\{ \frac{a_1}{\delta} x_1 + \frac{a_2}{\delta} x_2 + \ldots + \frac{a_{n-1}}{\delta} x_{n-1} \right\} = \left\{ \xi \right\}. \tag{3.32}$$

$$x_i \in \mathbf{H}, \quad i = 1, 2 \ldots n-1,$$

где ξ — переменный параметр.

Учитывая соотношение (3.28), получим

$$\left\{ \frac{a_1}{\delta} x_1 + \frac{a_2}{\delta} x_2 + \ldots + \frac{a_{n-1}}{\delta} x_{n-1} \right\} = \left\{ \frac{b}{\delta} - \frac{a_n}{\delta} x_n \right\}. \tag{3.33}$$

Обозначим через $\Lambda_n\left(\frac{b}{\delta}\right)$ — оптимальное значение целевой функции (3.25) при ограничении (3.28). Тогда, используя выражение для $\Lambda_{n-1}(\xi)$, получим следующее реккурентное соотношение для шага n:

$$\Lambda_n\left(\frac{b}{\delta}\right) = \min_{x_n} \left(\tilde{c}_n x_n + \Lambda_{n-1}\left(\frac{b}{\delta} - \frac{a_n}{\delta} x_n\right)\right), \tag{3.34}$$

$$x_n \in \mathbf{H}, \quad \mathbf{H} = \{0, 1, 2, \ldots\}.$$

Это соотношение можно обобщить на случай произвольного k-го шага:

$$\Lambda_k(\xi) = \min_{x_k} \left(\tilde{c}_k x_k + \Lambda_{k-1}\left(\xi - \frac{a_k x_k}{\delta}\right)\right), \tag{3.35}$$

при условии

$$\left\{ \frac{a_1}{\delta} x_1 + \ldots + \frac{a_{k-1}}{\delta} x_{k-1} + \frac{a_k}{\delta} x_k \right\} = \{ \xi \}. \tag{3.36}$$

Определим множество значений, которые может принимать ξ в (3.36). Так как a_i и x_i — целые, то ξ должно иметь вид $\xi = \frac{\alpha}{\delta}$, где α — целое. Так как ξ не может принимать значений, отличных от $\frac{0}{\delta}$, $\frac{1}{\delta}, \ldots, \frac{\delta-1}{\delta}$, то $\alpha = 0, 1, \ldots, \delta - 1$ и $x_k = 0, 1, \ldots, \delta - 1$.

Задачу асимптотического программирования (3.29)—(3.30) решаем следующим образом. Исследуя рекуррентное соотношение (3.35), находим последовательно $\Lambda_1(\xi)$, $\Lambda_2(\xi)$, ..., $\Lambda_{k-1}(\xi)$, $\Lambda_k(\xi)$ и т. д. для всех возможных значений ξ: $\xi = 0, \frac{1}{\delta}, \ldots, \frac{\delta - 1}{\delta}$.

При этом на каждом шаге k используем значение $\Lambda_{k-1}(\xi)$, вычисленное на предыдущем k — 1-м шаге. Наконец, на последнем шаге при $k = n$ находим $\Lambda_n\left(\xi = \frac{b}{\delta}\right)$ и искомые значения всех переменных $\overset{0}{x}_n$, $\overset{0}{x}_{n-1}$, ..., $\overset{0}{x}_1$.

Условия, при которых решение асимптотической задачи является решением исходной задачи ЛЦП

Рассмотрим условия, при которых решение асимптотической задачи является решением соответствующей задачи ЛЦП. Для этого необходимо сформулировать необходимые и достаточные условия оптимальности.

Теорема 4.1. *Оптимальным решением задачи* (3.25) *является решение, для которого* $\sum_{i=1}^{n} x_{N_i} \leqslant \delta_m - 1$, где $\delta_m = \det \mathbf{B}$.

Доказательство. Для доказательства используем некоторые идеи теории графов. С этой целью сформулируем задачу о кратчайшем пути на графе, решение которой позволяет получить решение асимптотической задачи. Положим $\frac{s_{mp}}{\delta_m}$ равным p-му компоненту вектора $\left\{ \frac{1}{\delta_m} \cdot \mathbf{S}_m \right\}$ из ограничения (3.26).

Построим граф с δ_m вершинами, которые обозначим через 0, 1, ..., $(\delta_m - 1)$. Этим вершинам поставим в соответствие элементы $0/\delta_m$, $1/\delta_m$ ... $(\delta_m - 1)/\delta_m$ группы, являющейся циклической по отношению к сложению по модулю 1.

Если существует p $(1 \leqslant p \leqslant n)$, удовлетворяющее условию

$$\left\{ \frac{i}{\delta_m} + \frac{s_{mp}}{\delta_m} \right\} = \left\{ \frac{j}{\delta_m} \right\} = j/\delta_m, \tag{3.38}$$

то соединим i-ю и j-ю вершины ориентированной дугой и припишем этой дуге длину c_p.

Модель асимптотического программирования при таких условиях приводит к задаче о нахождении кратчайшего пути на графе от вершины 0 до вершины, соответствующей значению $\{ l_m/\delta_m \}$ в левой части уравнения (3.26).

В случае графа, длины дуг которого неотрицательны, кратчайший путь между двумя вершинами не содержит петель, т. е. у графа с δ_m вершинами в этом случае не может быть более $\delta_m - 1$ дуг. Полагая $x_i = 1$ каждый раз, когда при выборе кратчайшего пути используем дугу длиной c_i, получим доказательство сформулированной теоремы.

Следствие. *Обозначим через* $|\mathbf{x}^*_N|$ *длину вектора* $\mathbf{x}^*_N$. *Тогда*

$$|\mathbf{x}^*_N| \leqslant |\det \mathbf{B}| - 1. \tag{3.39}$$

Доказательство. Определим длину вектора $\mathbf{x}^*_N$

$$|\mathbf{x}^*_n| = |x^2_{N_1} + x^2_{N_2} + \dots + x^2_{N_n}|^{1/2}.$$

Поскольку $x_{N_i} \geqslant 0$ для всех $i = \overline{1, n}$, получаем

$$|\mathbf{x}^*_N| = (x^2_{N_1} + x^2_{N_2} + \dots + x^2_{N_n})^{1/2} \leqslant x_{N_1} + x_{N_2} + \dots + x_{N_n} = \sum_i x_{N_i}.$$

Используя результаты теоремы 4.1, получим, что следствие справедливо.

Теорема 4.2. *Для того, чтобы оптимальное решение* x^*_N *асимптотической задачи* (3.26) *являлось также и решением исходной задачи целочисленного программирования, достаточно выполнения следующего условия*

$$\mathbf{B}^{-1}\mathbf{b} \geqslant l \max (|\det \mathbf{B}| - 1) \cdot \mathbf{1}^{(m\times 1)}, \tag{3.40}$$

где $l_{\max}$ — длина наибольшего по норме из m векторов, формирующих строки матрицы $\mathbf{B}^{-1}\mathbf{N}$, а $\mathbf{1}$ — m-мерный единичный вектор.

Доказательство. Запишем выражение $\mathbf{B}^{-1}\mathbf{N}\,\mathbf{x}^*_N$ в развернутом виде

$$\mathbf{B}^{-1}\mathbf{N}\mathbf{x}^*_N = \begin{bmatrix} (\mathbf{B}^{-1}\mathbf{N})_1\,\mathbf{x}^*_N \\ (\mathbf{B}^{-1}\mathbf{N})_2\,\mathbf{x}^*_N \\ \vdots \\ (\mathbf{B}^{-1}\mathbf{N})_m\,\mathbf{x}^*_N \end{bmatrix}. \tag{3.41}$$

Для любых векторов $\mathbf{p}$ и $\mathbf{q}$ всегда выполняется неравенство Шварца

$$\mathbf{pq} \leqslant |\mathbf{p}|\,|\mathbf{q}|$$

и, следовательно,

$$\mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \leqslant \begin{bmatrix} |(\mathbf{B}^{-1}\mathbf{N})_1| \times |\mathbf{x}^*_N| \\ |(\mathbf{B}^{-1}\mathbf{N})_2| \times |\mathbf{x}^*_N| \\ \dots\dots\dots \\ |(\mathbf{B}^{-1}\mathbf{N})_m| \times |\mathbf{x}^*_N| \end{bmatrix} \tag{3.42}$$

Используя определение $l_{\max}$ и учитывая следствие из выражения (3.39), получаем

$$(\mathbf{B}^{-1})\,\mathbf{N}\mathbf{x}_N^* \leqslant \begin{bmatrix} l_{\max}(|\det B| - 1) \\ l_{\max}(|\det B| - 1) \\ l_{\max}(|\det B| - 1) \end{bmatrix} = l_{\max}(\det \mathbf{B} - 1) \cdot \mathbf{1}^{(m\times 1)}. \qquad (3.43)$$

Допустим теперь, что выполняется условие (3.40) теоремы 4.2, тогда с учетом выражения (3.43) получаем

$$\mathbf{B}^{-1}\mathbf{b} \geqslant l_{\max}(|\det \mathbf{B}| - 1) \cdot \mathbf{1} \geqslant \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N^*.$$

Таким образом, имеем

$$\mathbf{x}_B^* = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{X}_N^* \geqslant 0. \qquad (3.44)$$

Откуда следует, что вектор $[\mathbf{x}_B^*, \mathbf{x}_N^*]$ есть решение задачи (3.1)—(3.4), так как он удовлетворяет условию неотрицательности.

Геометрическая интерпретация. Дадим геометрическую интерпретацию условий (3.40). Неравенство

$$\mathbf{B}^{-1}\mathbf{y} \geqslant l_{\max}(|\det \mathbf{B}| - 1) \cdot \mathbf{1} \qquad (3.40)$$

определяет конус, содержащийся в конусе

$$(\mathbf{B}^{-1})^{(m\times m)}\,\mathbf{y} \geqslant 0.$$

Именно в этом случае согласно теореме 4.2 решение асимптотической задачи является одновременно и решением исходной задачи (3.1)—(3.4).

Это означает, что вектор $\mathbf{b}^{(m\times 1)}$, стоящий в правой части ограничений исходной задачи, должен лежать внутри конуса асимптотической задачи (3.40). При этом данный конус представляет собой множество точек, расположенных на расстоянии $d \geqslant l_{\max} \times \times (\det \mathbf{B}\,| - 1)$ от гиперплоскостей, образующих конус $\mathbf{B}^{-1}\mathbf{y} \geqslant 0$ (рис. 4.3). Заметим, что образующие конической поверхности задаются при этом вектор-столбцами матрицы $\mathbf{B}^{-1}$, а именно $(\mathbf{B}^{-1})_i$ и $(\mathbf{B}^{-1})_j$.

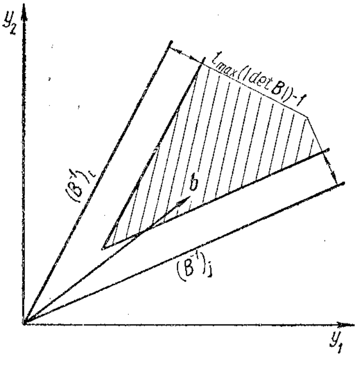


Рис. 4.3.

Описание алгоритма. Приведем формальное описание алгоритма асимптотического целочисленного программирования Гомори.

Пусть задача ЛЦП задана в виде (3.1)—(3.4).

1. Решаем соответствующую ей ЛП-задачу и находим базис **B**, отвечающий оптимальному решению.

Если $\mathbf{x}_B$ не удовлетворяет условию целочисленности, то проверяем условие (3.40) теоремы 4.2.

Если оно выполняется, то можно перейти к решению асимптотической задачи ЛЦП.

2. Записываем асимптотическую задачу вида (3.25) при ограничениях (3.26)—(3.27), где x_N — вектор небазисных переменных.

3. Решаем асимптотическую задачу (например, методом динамического программирования) и определяем ее оптимальное решение.

4. Используя соотношения (3.8), находим вектор базисных переменных x_B^*. Искомое решение исходной задачи (3.1) определяется как $x_0 = [x_B^*, \; x_N^*]$

В заключение подчеркнем основные преимущества перехода к решению асимптотической задачи. Эта задача в отличие от исходной имеет лишь одно ограничение вида (3.26) (вместо m ограничений исходной задачи), что может существенно упростить ее решение.

Заметим, что ограничение вида (3.26) представляет собой своего рода обобщенное правильное отсечение.

Пример 4.4. Проверим возможность перехода к асимптотической задаче для задачи ЛЦП, приведенной в табл. 4.1.

Условие перехода к асимптотической задаче таково:

$$B^{-1}b \geqslant l_{\max}(|\det B| - 1)\,\bar{1},$$

где $l_{\max}$ — длина наибольшего из m векторов, формирующих строки матрицы $B^{-1}N$.

Заметим, что $B^{-1}b$ — столбец A_0, а $B^{-1}N$ — столбцы из небазисных векторов A_4, A_5 в табл. 4. 1.

Таким образом,

$$B^{-1}b = \begin{bmatrix} 40/9 \\ 1 \\ 23/9 \end{bmatrix}, \quad B^{-1}N = \begin{bmatrix} 5/9 & 1/9 \\ -6 & 1 \\ 4/9 & -1/9 \end{bmatrix}.$$

Кроме того, ранее мы нашли, что $\det B = \det \Delta = 9$. Таким образом,

$$l_{\max} = \max\left(\sqrt{(5/9)^2 + (1/9)^2};\ \sqrt{6^2 + 1^2};\ \sqrt{\left(\frac{4}{9}\right)^2 + \left(-\frac{1}{9}\right)^2}\right) = \\ = \sqrt{6^2 + 1} = \sqrt{37} \approx 6,1.$$

Так как $\frac{40}{9} < 6,1\,(9-1)$, мы не можем гарантировать, что решение асимптотической задачи будет оптимальным решением исходной задачи.

Пример 4.5. Рассмотрим пример, когда решение асимптотической задачи оказывается решением связанной с ней исходной задачи целочисленного программирования.

Найти

$$\max(3x_1 + 4x_2) \tag{1}$$

при условиях:

$$\begin{cases} 3x_1 + 2x_2 \leqslant 24 \\ x_1 + 4x_2 \leqslant 27 \end{cases} \tag{2}$$

$$x_1,\ x_2 \geqslant 0; \tag{3}$$

$$x_1,\ x_2 = \text{целые} \tag{4}$$

1. Решаем нецелочисленную задачу ЛП (1) — (3) симплекс-методом, результаты последовательных итераций приводятся в табл. 4.9 — 4.11.

2. Проверим условие возможности перехода к асимптотической задаче. При этом используем результаты табл. 4.11:

$$B^{-1}b = A_0 = \begin{bmatrix} 4,2 \\ 5,7 \end{bmatrix}; \quad \det B = \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix} = 10; \quad \det B - 1 = 9;$$

$$B^{-1}N = [A_3\ A_4] = \begin{bmatrix} 0,4 & -0,2 \\ -0,1 & 0,3 \end{bmatrix}.$$

Тогда $l_{\max} = \max(\sqrt{0{,}42 + (-0{,}2)^2};\quad \sqrt{(0{,}1)^2 + (0{,}3)^2}) = \max(\sqrt{0{,}2};\quad \sqrt{0{,}1}) = \sqrt{0{,}2} \approx 0{,}45$.

Необходимо, чтобы выполнялось условие

$$\mathbf{B}^{-1}\mathbf{b} \geqslant l_{\max}\,(\,|\det \mathbf{B}\,| - 1)\cdot \bar{1}. \tag{5}$$

Таблица 4.9

c_i	B_x	A_0	A_1	A_2		A_4
0	x_3	24	3	2	1	0
0	x_4	27	1	$\boxed{4}$	0	1
	Δ	0	−3	−4	0	0

Таблица 4.10

c	B_x	A_0	A_1	A_2	A_3	A_4
0	x_3	$10\frac{1}{2}$	$\boxed{10/4}$		1	1/2
4	x_2	27/4	1/4	1		1/4
	Δ	27	−2	0	0	1

Таблица 4.11

c_i	B_x	A_0	A_1	A_2	A_3	A_4
3	x_1	4,2	1		0,4	−0,2
4	x_2	5,7		1	−0,1	0,3
	Δ	35,4	0	0	0,8	0,6

Подставляя в (5) найденные значения, находим, что

$$\begin{bmatrix} 4{,}2 \\ 5{,}7 \end{bmatrix} \geqslant 9 \cdot 0{,}45 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Таким образом, можно перейти к решению асимптотической задачи.

3. Находим значения $\tilde{c}_j$ для небазисных переменных x_3 и x_4 из индексной строки табл. 4.11. Они равны соответственно $\tilde{c}_3 = 0{,}8$; $\tilde{c}_4 = 0{,}6$.

4. Матрицу $\mathbf{B} = \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$ необходимо привести к приведенной форме Смита Δ. Используя метод, описанный в приложении, найдем такие унимодулярные матрицы $\mathbf{U}$ и $\mathbf{V}$, что $\mathbf{UBV} = \Delta$:

$$\overset{\mathbf{U}}{\begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix}} \times \overset{\mathbf{B}}{\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}} \times \overset{\mathbf{V}}{\begin{bmatrix} 1 & -4 \\ 0 & 1 \end{bmatrix}} = \overset{\Delta}{\begin{bmatrix} 1 & 0 \\ 0 & -10 \end{bmatrix}}$$

Таким образом, $\delta_1 = 1$; $\delta_2 = -10$. Определив $\mathbf{U}$, вычисляем вспомогательные вектор $\mathbf{Y}^{(2\times1)}$ и матрицу $\mathbf{S}^{(2\times2)}$:

$$\mathbf{y}^{(2\times1)} = \mathbf{U} \times \mathbf{b} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} \times \begin{bmatrix} 24 \\ 27 \end{bmatrix} = \begin{bmatrix} 27 \\ -57 \end{bmatrix}, \tag{6}$$

$$\mathbf{S}^{(2\times2)} = \mathbf{UN} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} \times \overset{A_3 A_4}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix}. \tag{7}$$

Найдем теперь компоненты y_m и S_m, учитывая, что $m = 2$, $y_2 = -57$, $S_2 = [1 \; -3]$.

4. Теперь можно составить асимптотическую задачу для задачи (1) — (4). Используя ее общее выражение

$$\min \tilde{\mathbf{c}}_N \mathbf{x}_N \tag{8}$$

при условиях

$$\left\{ y_m/\delta_m \right\} = \left\{ \left\{ \frac{1}{\delta_m} \times \mathrm{s}_m \right\} \times \mathbf{x}_N \right\}, \tag{9}$$

$$x_N \in \mathbf{H} \tag{10}$$

и подставляя в (8) — (10) значения δ_2, y_2 и S_2, получим

$$\min (\tilde{c}_3 x_3 + \tilde{c}_4 x_4) = \min (0{,}8x_3 + 0{,}6x_4) \tag{11}$$

при условии

$$\left\{ y_2/\delta_2 \right\} = \left\{ \left\{ 1/\delta_m \times s_{m_1} \right\} x_3 + \left\{ 1/\delta_m \times s_{m_2} \right\} x_4 \right\} \to$$

$$\to \left\{ \frac{-57}{-10} \right\} = \left\{ \left\{ -\frac{1}{10} \times [1, -3] \right\} \times \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \right\}. \tag{12}$$

Откуда получим

$$\frac{7}{10} = \left\{ \left\{ -\frac{1}{10} \right\} x_3 + \left\{ \frac{3}{10} \right\} x_4 \right\} = \left\{ \frac{9}{10} x_3 + \frac{3}{10} x_4 \right\}.$$

Итак, асимптотическая задача записывается в виде

$$\min_{x_3, x_4} (\tilde{c}_3 x_3 + \tilde{c}_4 x_4) = \min (0{,}8x_3 + 0{,}6x_4) \tag{13}$$

при условиях:

$$\left\{ \frac{9}{10} x_3 + \frac{3}{10} x_4 \right\} = \frac{7}{10}, \tag{14}$$

$$x_3, \; x_4 \in \mathbf{H}.$$

5. Решаем асимптотическую задачу методом динамического программирования. Первый шаг. Составляем основное рекуррентное соотношение

$$\Lambda_1 (\xi) = \min 0{,}8x_3 \tag{15}$$

при условии

$$\left\{ \frac{9}{10} x_3 \right\} = \left\{ \xi \right\}, \tag{16}$$

где $\xi = 0, \frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10}$. Результаты решения представлены в табл. 4.12.

Таблица 4.12

ξ	0	1/10	2/10	3/10	4/10	5/10	6/10	7/10	8/10	9/10
$\Lambda_1 (\xi)$	0	7,2	6,4	5,6	4,8	4,0	3,2	2,4	1,6	0,8
x_3^*	0	9	8	7	6	5	4	3	2	1

Заметим, что нахождение оптимального решения $x_3^*(\xi)$ не составляет труда, оно находится из условия: $\min x_3$, где x_3 удовлетворяет условию $\left\{\frac{9}{10}x_3-\xi\right\}=0$.

Второй шаг. Основное рекуррентное соотношение имеет вид

$$\Lambda_2(\xi)=\min_{x_4}\left(c_4x_4+\Lambda_1\left(\left\{\xi-\frac{3}{10}x_4\right\}\right)\right),$$

где $\xi=\frac{7}{10}$.

Обозначим $c_4x_4+\Lambda_1\left(\frac{7}{10}-\frac{3}{10}x_4\right)=\Omega_2(x_4)$.

Задавая x_4 последовательно значения 0, 1, ..., 9 и используя значения для $\Lambda_1\left(\left\{\frac{7}{10}-\frac{3}{10}x_4\right\}\right)=\Lambda_1(\xi)$ из табл. 4.12, вычисляем $\Omega_2(x_4)$ и результаты заносим в табл. 4.13.

Таблица 4.13

x_4	0	1	2	3	4	5	6	7	8	9
c_4x_4	0	0,6	1,2	1,8	2,4	3	3,6	4,2	4,8	5,4
$\Lambda_1(x_4)$	2,4	4,8	7,2	1,6	4,0	6,4	0,8	3,2	5,6	0
$\Omega_2(x_4)$	2,4	5,4	8,4	3,4	6,4	9,4	4,4	7,4	10,4	5,4

Из табл. 4.13 находим, что $\min\limits_{x_4}\Omega_2(x_4)=2,4=\Lambda_2$ достигается при $x_4^*=0$. Но тогда $\xi=\left\{\frac{7}{10}-\frac{3}{10}\cdot x_4\right\}=\frac{7}{10}$ и из табл. 4.12 определяем, что $x_3^*=3$.

Итак, мы нашли оптимальное решение асимптотической задачи $x_3^*=3$; $x_4^*=0$. Определим теперь соответствующие значения базисных переменных x_1 и x_2:

$$X_B=\begin{bmatrix}x_1\\x_2\end{bmatrix}=B^{-1}b-B^{-1}N\times\begin{bmatrix}x_3^*\\x_4^*\end{bmatrix}=\begin{bmatrix}4,2\\5,7\end{bmatrix}-\begin{bmatrix}0,4&-0,2\\-0,1&0,3\end{bmatrix}\times\begin{bmatrix}3\\0\end{bmatrix}=$$

$$=\begin{bmatrix}4,2\\5,7\end{bmatrix}-\begin{bmatrix}1,2\\-0,3\end{bmatrix}=\begin{bmatrix}3\\6\end{bmatrix}$$

Таким образом, найдено искомое решение задачи (1) — (4) $x_1^0=3$; $x_2^0=6$; $x_3^0=3$; $x_4^0=0$. При этом $L_{\max}=\max(3x_1+4x_2)=33$. Заметим, что при решении этой задачи методом отсекающих плоскостей пришлось бы несколько раз строить отсечения по Гомори и затратить значительно больше времени.

§ 4. МЕТОД ВЕТВЕЙ И ГРАНИЦ

Метод ветвей и границ[1] относится к группе комбинаторных методов дискретного программирования и является одним из наиболее распространенных методов этой группы.

Рассматриваем задачу дискретного программирования в общей форме:

минимизировать $z = f(x)$ (4.1)

при условиях $x \in G$,

где G — конечное множество.

В основу метода ветвей и границ положены следующие построения, позволяющие в ряде случаев существенно уменьшить объем перебора вариантов.

Вычисление нижней границы (оценки) min z. Часто удается найти нижнюю границу (оценку) целевой функции на множестве планов G (или на его некотором подмножестве) $\xi(G)$ — такую, что для всех $x \in G$ имеет место $f(x) \geqslant \xi(G)$.

Ветвление (разбиение на подмножества). Реализация метода связана с постоянным ветвлением множества планов G на дерево подмножеств.

Ветвление происходит по следующей схеме.

Нулевой шаг. Имеется исходное множество $G = G^{(0)}$. Некоторым способом его разбивают на конечное число подмножеств $G_1^{(1)}, G_2^{(1)}, \ldots, G_{r_1}^{(1)}$.

k-й шаг ($k \geqslant 1$). Имеются множества $G_1^{(k)}, G_2^{(k)}, \ldots, G_r^{(k)}$, еще не подвергавшиеся ветвлению. По определенному правилу (указанному ниже) среди них выбирают множество $G_{\nu(k)}^{(k)}$ и разбивают на конечное число подмножеств (рис. 4.4).

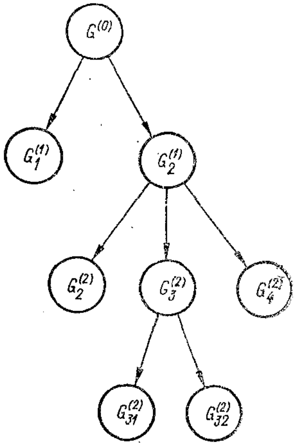


Рис. 4.4.

Затем еще не подвергавшиеся разбиению множества $G_1^{(k)}, G_2^{(k)}, \ldots, G_{\nu(k)+1}^{(k)}, \ldots, G_{\rho(k)}^{(k)}$, а также подмножества $G_{\nu(k),1}^{(k)}, G_{\nu(k),2}^{(k)}, \ldots, G_{\nu(k),s}$ переобозначают $G_1^{(k+1)}, G_2^{(k+2)}, \ldots, G_\rho^{(k+1)}$.

Пересчет оценок. Если $G_1 \subset G_2$, то $\min\limits_{x \in G_1} f(x) \geqslant \min\limits_{x \in G_2} f(x)$.

Поэтому, разбивая в процессе решения множество G_0 на подмножества $G_1, G_2, \ldots, G_s$ $\left(G_0 = \bigcup\limits_{i=1}^{s} G_i\right)$, всегда считают, что оценка для любого

---

[1] Впервые метод ветвей и границ был предложен в работе Лэнг и Дойг в 1960 г. применительно к задаче целочисленного программирования. Однако эта работа не оказала заметного влияния на развитие идей дискретного программирования. Второе рождение метода связано с работой Литтла, Мурти, Суини и Кэрел, 1963 г., посвященной задаче о коммивояжере.

из них не меньше оценки для исходного множества G_0, т. е. для всех множеств $\xi(G_i) \geqslant \xi(G_0)$. В конкретных ситуациях это неравенство для некоторых i может оказаться строгим неравенством.

Нахождение планов. Для конкретных задач могут быть указаны различные способы определения планов в последовательно разветвляемых подмножествах. Любой такой способ опирается на специфику задачи.

Признак оптимальности. Пусть $G = \bigcup\limits_{i=1}^{s} G_i$ и некоторый план $X_0 \in G_\nu$. Если при этом $f(X_0) = \xi(G_\nu) \leqslant \xi(G_i)$ для всех i, то X_0 — оптимальный план (доказательство следует из определения оценки).

Оценка точности приближенного решения. Пусть $G = \bigcup\limits_{i=1}^{s} G_i$, $\xi = \min\limits_{i=1,\ 2,\ \ldots,\ s} \xi(G_i)$. Если X — некоторый план задачи, то $\xi \leqslant \min\limits_{x \in G} f(X) \leqslant f(X)$.

Если разность $\Delta = f(X) - \xi$ невелика, то X можно принять за приближенное решение, а Δ будет оценкой его точности.

Описание алгоритма метода

Нулевой шаг. Вычисляют оценку $\xi(G) = \xi(G^{(0)})$. Если при этом находят такой план X_0, что $f(X_0) = \xi(G^{(0)})$, то X_0 — оптимальный план.

Если оптимальный план не найден, то некоторым способом разбивают множество $G^{(0)}$ на конечное число непересекающихся подмножеств $G^{(0)} = G_1^{(1)} \cup G_2^{(1)} \cup \ldots \cup G_{\rho_1}^{(1)}$ и переходят к первому шагу.

Первый шаг. Вычисляют оценки $\xi(G_i^{(1)})$ при $i = 1, 2, \ldots, \rho_1$. Если удается найти такой план X_0, что $X_0 \in G_\rho^{(1)}$ $(1 \leqslant \rho \leqslant \rho_1)$ и $f(X_0) = \xi(G_\rho^{(1)}) \leqslant \xi(G_i^{(1)})$ при $i = 1, 2, \ldots, \rho_1$, то X_0 — оптимальный план.

Если же оптимальный план не найден, то для дальнейшего разбиения выбирают наиболее перспективное множество $G_\nu^{(1)}$ по следующему правилу:

$$\xi(G_\nu^{(1)}) = \min_{i=1,2,\ldots,\rho_1} \xi(G_i^{(1)}). \tag{4.2}$$

Разбивают множество $G_\nu^{(1)}$ на несколько подмножеств: $G_\nu^{(1)} = \bigcup\limits_{i=1}^{s_1} G_{\nu,i}^{(1)}$. Еще не подвергавшиеся разбиению множества переобозначают $G_1^{(2)}, G_2^{(2)}, \ldots, G_{\rho_2}^{(2)}$ и переходят ко второму шагу.

k-й шаг $(k \geqslant 2)$. Вычисляют оценки $\xi(G_i^{(k)})$ при $i = 1, 2, \ldots, \rho_k$. Если удается найти такой план X_0, что $X_0 \in G_\rho^{(k)}$ $(1 \leqslant \rho \leqslant \rho_k)$ и $f(X_0) = \xi(G_\rho^{(k)}) \leqslant \xi(G_i^{(k)})$ для всех $i = 1, 2, \ldots, \rho_k$, то X_0 — оптимальный план.

Если же оптимальный план не найден, то снова выбирают наиболее перспективное множество $G_\nu^{(k)}$ по правилу:

$$\xi(G_\nu^{(k)}) = \min_{i=1,2,\ldots,\rho_k} \xi(G_i^{(k)}). \tag{4.3}$$

Разбивают множество $G^{(k)}_{v(k)}$ на несколько непересекающихся подмножеств $G^{(k)}_{v} = \bigcup\limits_{i=1}^{s_k} G^{(k)}_{v,i}$ и переходят к $(k+1)$-му шагу.

Примечание. При реализации описанной выше общей схемы метода ветвей и границ для отдельных задач дискретного программирования необходимо разрабатывать правила ветвления, способы вычисления оценок и нахождения планов, исходя из специфики конкретных задач.

Метод ветвей и границ для задачи целочисленного программирования

Рассматриваем частично целочисленную задачу ЛП: минимизировать

$$z = f(\mathbf{X}) = \sum_{j=1}^{n} c_j x_j \tag{4.4}$$

при условиях:

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant b_i \quad (i = 1, 2, \ldots, m), \tag{4.5}$$

$$0 \leqslant x_j \leqslant d_j \quad (j = 1, \ldots, n), \tag{4.6}$$

$$x_j \text{ — целые числа} \quad (j = 1, 2, \ldots, n_1). \tag{4.7}$$

Как и в методе отсекающих плоскостей, процесс начинают с решения непрерывной задачи ЛП. Если полученный при этом оптимальный план $\mathbf{X}_0$ не удовлетворяет условию (4.7), то значение целевой функции $\xi_0 = f(\mathbf{X}_0)$ дает нижнюю оценку для искомого решения, т. е. $\min z = \xi_0$.

Пусть некоторая переменная $x_{i_0} (1 \leqslant i_0 \leqslant n_1)$ не получила в плане $\mathbf{X}_0$ целочисленного решения. В целочисленном плане значение x_{i_0} следует либо уменьшить по крайней мере до $[x_{i_0}]$, либо увеличить по крайней мере до $[x_{i_0}] + 1$.

Если границы изменения x_{i_0} заранее не заданы, то их можно вычислить, решив для этого две вспомогательные задачи ЛП. Эти задачи состоят в максимизации и минимизации x_{i_0} при условиях (4.5) и (4.6).

Теперь для каждого фиксированного целочисленного значения x_{i_0} в найденном отрезке $(x_{i_0 \min}, x_{i_0 \max})$ находят $\min z$, решая задачу ЛП с ограничениями (4.5), (4.6) и с дополнительным ограничением $x_{i_0} \leqslant k_{i_0} = [x_{i_0}]$ или ограничением $x_{i_0} \geqslant k_{i_0} + 1$.

Таким образом, все указанные выше возможности можно представить в виде некоторого дерева, в котором вершина 0 отвечает плану $\mathbf{X}_0$, а каждая из соединенных с ней ветвью вершин отвечает оптимальному плану следующей задачи: минимизировать z при условиях (4.5), (4.6) и дополнительном условии, что переменной x_{i_0} дано значение $x_{i_0} \leqslant k_{i_0}$ или $x_{i_0} \geqslant k_{i_0} + 1$, где k_{i_0} — целое число. Каждой из таких вершин приписывают оценку $\xi = \xi(i_0, k)$, которая равна $\min z$ при указанных выше ограничениях. Очевидно, $\xi_0 \leqslant \xi(i_0, k)$ для всех k.

Если оптимальные планы полученных задач удовлетворяют условиям целочисленности, то план с минимальной оценкой $\xi_0 = \min\limits_k \xi(i_0, k)$ и будет оптимальным планом исходной задачи. В противном случае возникает необходимость в продолжении ветвления. При этом каждый раз для очередного ветвления выбирают вершину с наименьшей оценкой.

Любой маршрут в дереве от начальной вершины 0 до некоторой вершины определяет допустимую последовательность выбора целочисленных решений для переменных. Процесс продолжают до тех пор, пока продолжение ветвления становится невозможным.

Каждая конечная вершина отвечает некоторому допустимому целочисленному плану. Вершина с минимальной оценкой дает оптимальный план.

Описание алгоритма

Нулевая итерация. 1. Задание множества G_0. Множество G_0 определяется условиями (4.5) и (4.6).

2. Находим оптимальный план $\mathbf{X}_0$ задачи (4.4) при условиях (4.5), (4.6).

3. Вычисляем оценку $\xi(G_0) = f(\mathbf{X}_0)$.

Если $\mathbf{X}_0$ удовлетворяет условиям целочисленности, то он является искомым. Иначе переходим к первой итерации.

Первая итерация. 1. Ветвление. Выбираем некоторую нецелочисленную компоненту $x_i = x_{i0}$, $1 \leqslant i \leqslant n_1$. Множество G_0 разбивают на два непересекающихся подмножества $G_1^{(1)}$ и $G_2^{(1)}$ следующим образом:

$$G_1^{(1)} = \{\mathbf{X} : \mathbf{X} \in G_0 \text{ и } x_i \leqslant [x_{i_0}]\}; \tag{4.8}$$

$$G_2^{(1)} = \{\mathbf{X} : \mathbf{X} \in G_0 \text{ и } x_i \geqslant [x_{i_0}] + 1\} \tag{4.9}$$

2. Решаем ЛП-задачу (4.4) на множестве $G_1^{(1)}$ и находим оптимальный план $\mathbf{X}_1^{(1)}$. Соответственно решив задачу (4.4) на множестве $G_2^{(1)}$, находим план $\mathbf{X}_2^{(1)}$.

3. Вычисляем оценки $\xi(G_1^{(1)}) = f(\mathbf{X}_1^{(1)})$ и $\xi(G_2^{(1)}) = f(\mathbf{X}_2^{(1)})$.

4. Проверяем признак оптимальности.

Если $\mathbf{X}_i^{(1)}$ — целочисленный и $\xi(G_i^{(1)}) = \min\{\xi(G_1^{(1)}, \xi(G_2^{(1)})\}$, то $\mathbf{X}_i^{(1)}$ — искомый план. В противном случае переходим к следующей итерации и выполняем процедуру ветвления.

$k + 1$-я итерация. Пусть проведено k итераций, в результате которых построены подмножества $G_i^{(k)}$, $i = \overline{1, r_k}$, определены оценки $\xi(G_i^{(k)})$ и еще не найдено искомое решение. Тогда выбираем наиболее перспективное подмножество $G_\nu^{(k)}$ — такое, что

$$\xi(G_\nu^{(k)}) = \min_i \xi(G_i^{(k)}). \tag{4.10}$$

1. Производим ветвление $G_\nu^{(k)}$ на подмножества: $G_{\nu_1}^{(k)}$ и $G_{\nu_2}^{(k)}$ так, что $G_\nu^{(k)} = G_{\nu_1}^{(k)} \cup G_{\nu_2}^{(k)}$

С этой целью выберем некоторую нецелочисленную компоненту плана $X_\nu^{(k)}$, например, $x_{s,\nu}^{(k)} = x_{s0}$, $1 \leqslant s \leqslant n_1$.

Тогда множества $G_{\nu,1}^{(k)}$ и $G_{\nu,2}^{(k)}$ определяются из условий:

$$G_{\nu,1}^{(k)} = \{X : X \in G_\nu^{(k)} \cap x_s \leqslant [x_{s0}]\}, \qquad (4.11)$$

$$G_{\nu,2}^{(k)} = \{X : X \in G_\nu^{(k)} \cap x_s \geqslant [x_{s0}] + 1\}.$$

2. Решаем ЛП-задачи (4.4) на множествах $G_{\nu,1}^{(k)}$ и $G_{\nu,2}^{(k)}$ и находим соответствующие оптимальные планы $X_{\nu,1}^{(k)}$ и $X_{\nu,2}^{(k)}$ и соответствующие им оценки $\xi(G_{\nu,1}^{(k)}) = f(X_{\nu,1}^{(k)})$; $\xi(G_{\nu,2}^{(k)}) = f(X_{\nu,2}^{(k)})$.

3. Если $X_{\nu,1}^{(k)}$ удовлетворяет условию целочисленности, то вершина $G_{\nu,1}^{(k)}$ — концевая и дальнейшему разбиению не подвергается. Если при этом также $\xi(G_{\nu,1}^{(k)}) \leqslant \{\xi(G_i^{(k)})\}$ для всех висячих вершин $G_i^{(k)}$, то $X_{\nu,1}^{(k)}$ — искомый целочисленный план (это признак оптимальности). В противном случае необходимо продолжить процесс ветвления и дальше.

Укажем некоторые особенности применения метода ветвей и границ для задач ЦП.

1. Если все коэффициенты c_j целевой функции — целые при $1 \leqslant j \leqslant n_1$ и равны нулю при $j > n_1$, то оценку $\xi(G_\nu^{(k)})$ можно заменить на более сильную оценку $\xi'(G_\nu^{(k)}) = \,]f(X_\nu^{(k)})[$, где $]a[$ обозначено наименьшее целое, но не меньшее, чем a, т. е. округленное до ближайшего целого с избытком.

2. Из описания алгоритма следует, что в применении метода ветвей и границ для полностью целочисленных и для частично-целочисленных задач нет никакой разницы.

Геометрически этот метод можно интерпретировать таким образом. Гиперплоскость, определяемая целевой функцией задачи, вдавливается внутрь многогранника планов соответствующей задачи ЛП до встречи с ближайшей целочисленной точкой этого многогранника.

3. Вводимые на каждой итерации новые ограничения вида $x_i \geqslant [x_{i_0}] + 1$ или $x_i \leqslant [x_{i_0}]$ играют роль отсечений.

4. При вводе нового ограничения нет необходимости заново решать всю задачу (4.4), а можно использовать результаты предыдущей итерации, непосредственно вводя в таблицу оптимального решения новое ограничение.

5. При решении ЛП-задачи максимизации методом ветвей и границ используют верхнюю границу:

$$\xi(G_k^{(i)}) = \max_{x \in G_k^{(i)}} f(x).$$

При этом признак оптимальности формулируется противоположным образом.

Пример 4.6. Рассмотрим пример 4.2, решенный методом Гомори, в § 2, гл. 4. В эквивалентной форме записи задача имеет вид:

$$\min -(x_1 + x_2)$$

$$\begin{cases} 2x_1 + 11x_2 \leqslant 38 \\ x_1 + x_2 \leqslant 7 \\ 4x_1 - 5x_2 \leqslant 5 \end{cases}$$

$$x_1,\ x_2 \geqslant 0,\ x_1,\ x_2 \text{ — целые числа.}$$

Нулевой шаг. Оптимальный план задачи ЛП $X_0 = \left(\frac{40}{9};\ \frac{23}{9}\right)$.

Тогда имеем $\xi(G^{(0)}) =]\, f(X_0)\, [= \;] -7\, [= -7$.

Так как план X_0 не удовлетворяет условию целочисленности, возьмем его нецелочисленную компоненту x_1 и разобьем множество $G^{(0)}$ на $G_1^{(1)}$ и $G_2^{(1)}$:

$$G_1^{(1)} = \{X : X \in G^{(0)} \cap x_1 \leqslant 4\},$$

$$G_2^{(1)} = \{X : X \in G^{(0)} \cap x_1 \geqslant 5\}.$$

Первый шаг. Решаем две задачи ЛП, состоящие в минимизации исходной задачи по множествам $G_1^{(1)}$ и $G_2^{(1)}$.

Покажем, как можно найти $\min f(x)$ на множествах $G_1^{(1)}$ и $G_2^{(1)}$, используя результаты предыдущей итерации.

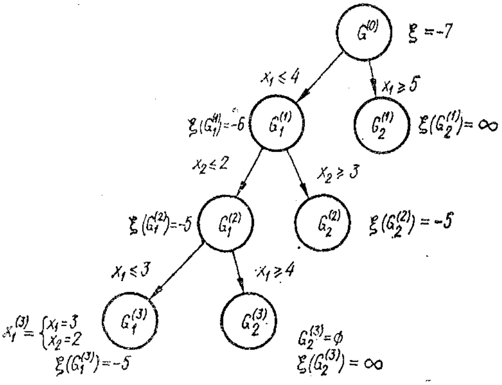


Рис. 4.5.

1. Итак, пусть требуется найти $\min f(x)$ при $X \in G_1^{(1)}$, где $G_1^{(1)} = G_0 \cap x_1 \leqslant 4$.

Воспользуемся таблицей оптимального плана X_0, которая приводится ниже (табл. 4.14).

Таблица 4.14

c_i			1	1	0	0	0
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
1	A_1	40/9	1	0	0	5/9	1/9
0	A_3	1	0	0	1	—6	1
1	A_2	23/9	0	1	0	4/9	—1/9
		7	0	0	0	1	0

Необходимо ввести в систему новое ограничение $x_1 \leqslant 4$. Запишем его в эквивалентном виде: $x_1 + x_6 = 4$. Допишем строку A_6 к табл. 4.14 и приходим к следующей таблице:

Таблица 4.15

c_i	B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	A_1	40/9	1	0	0	5/9	1/9	0
	A_3	1	0	0	1	—6	1	0
	A_3	23/9	0	1	0	4/9	—1/9	0
	A_6	4	1	0	0	0	0	1
	A_6^1	—4/9	0	0	0	—5/9	—1/9	1

Так как столбец A_1 должен быть базисным, то вычтя из строки A_6 строку A_1, получим A_6^1, которую и используем в дальнейших операциях, а A_6 вычеркнем. Обратим внимание, что строка A_6^1 представляет собой правильное отсечение Гомори, составленное по строке A_1. Следовательно, метод ветвей и границ для задачи ЛЦП представляет собой разновидность метода отсечений Гомори.

Так как $a_{60}^1 = -4/9 < 0$, то столбец A_0 табл. 4.15 представляет собой псевдоплан и потому применим двойственный симплекс-метод. Так как табл. 4.15 совпадает с табл. 4.1, то спустя две итерации мы найдем оптимальный план $X_1^{(1)}$, который приведен в табл. 4.3 (см. § 2).

Таким образом,

$$x_1 = 4;\quad x_2 = \frac{30}{11},\quad \xi(G_1^{(1)}) =]\,f(X_1^{(1)})\,[=]-\frac{74}{11}[= -6.$$

2. Найдем теперь $\min f(x)$ на множестве $G_2^{(1)} = G_0 \cap x_1 \geqslant 5$. Вводя свободную переменную x_7, ограничение $x_1 \geqslant 5$ запишем в виде $x_1 - x_7 = 5$ или $-x_1 + x_7 = -5$.

Запишем его в дополнительную строку A_7 в табл. 4.16, основная часть которой совпадает с табл. 4.14.

Складывая A_7 и A_1, чтобы снова привести A_1 к единичному виду, получим строку A_7 (табл. 4.16).

Таблица 4.16

c_i	B_x	A_0	A_1	A_2	A_4	A_4	A_5	A_7
	A_1	40/9	1	0	0	5/9	1/9	0
	A_3	1	0	0	1	—6	1	0
	A_2	23/9	0	1	0	4/9	—1/9	0
	A_7	—5	—1	0	0	0	0	1
	A_7^1	—5/9	0	0	0	5/9	1/9	1

Но так как $x^1_{70} = -5/9 < 0$, а $x^1_{7j} \geqslant 0$, для всех $j = \overline{1,7}$, то это есть признак неразрешимости данной задачи. Следовательно, $G^{(1)}_2 = \varnothing$ и поэтому $\xi\ (G^{(1)}_2) = \infty$.

Разбиваем $G^{(1)}_1$ на $G^{(1)}_{1,1}$ и $G^{(1)}_{1,2}$, где $G^{(1)}_{1,1} = \{X\colon\ X \in G^{(1)}_1 \cap x_2 \leqslant 2\}$, $G^{(1)}_{1,2} = X\colon\ X \in G^{(1)}_1 \cap x_2 \geqslant 3\}$.

Переобозначим $G^{(1)}_{1,1} = G^{(2)}_1$, $G^{(1)}_{1,2} = G^{(2)}_2$, $G^{(1)}_2 = G^{(2)}_3$.

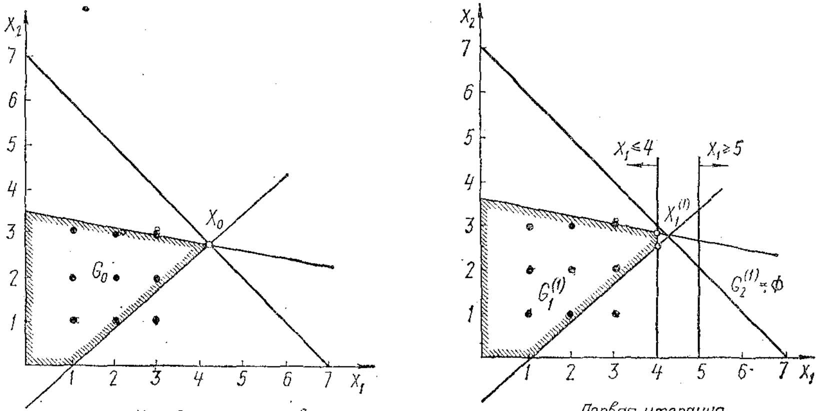
Второй шаг. Решаем две задачи ЛП, состоящие в минимизации исходной задачи при дополнительном ограничении $x_2 \leqslant 2$ и $x_2 \geqslant 3$.

Тогда

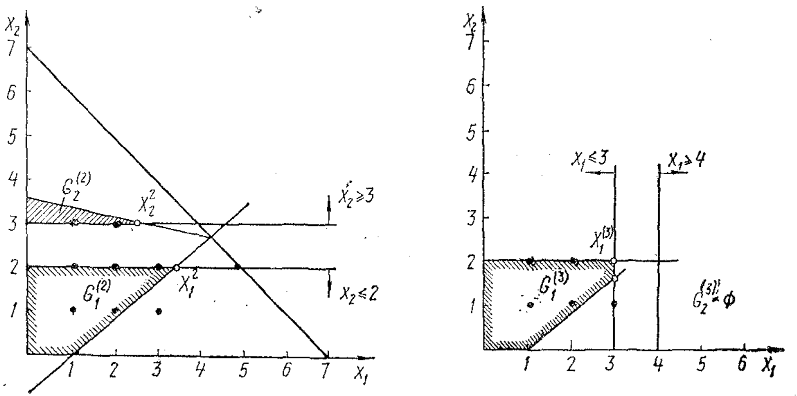
$$X^{(2)}_1 = \left\{3\frac{3}{4}\ ;\ 2\right\} \text{ и } \xi'\ (G^{(2)}_1) = \left]-5\frac{3}{4}\right[= -5;$$

$$X^{(2)}_2 = \left\{2\frac{1}{2}\ ;\ 3\right\} \text{ и } \xi'\ (G^{(2)}_2) = \left]-5\frac{1}{2}\right[= -5;$$

$$G^{(2)}_3 = \varnothing;\quad \xi'\ (G^{(2)}_3) = \infty.$$



Исходное множество G_0 Первая итерация



Вторая итерация Третья итерация

Рис. 4.6

Производим разветвление множества $G_1^{(2)}$:

$$G_1^{(2)} = G_{1,1}^{(2)} \cup G_{1,2}^{(2)},$$

где $G_{1,1}^{(2)} = \{X : X \in G_1^{(2)} \cap x_1 \leqslant 3\}$, $G_{1,2}^{(2)} = \{X : X \in G_1^{(2)} \cap x_1 \geqslant 4\}$.

Переобозначим $G_{1,1}^{(2)} = G_1^{(3)}$, $G_{1,2}^{(2)} = G_2^{(3)}$, $G_2^{(2)} = G_3^{(3)}$, $G_3^{(2)} = G_4^{(3)}$.

Третий шаг. Решаем две задачи ЛП, состоящие в минимизации исходной задачи по множествам $G_1^{(3)}$ и $G_2^{(3)}$. Находим $X_1^{(3)} = (3, 2)$ и $\xi'(G_1^{(3)}) =]-5[= -5$; $G_2^{(3)} = \varnothing$ и $\xi(G_2^{(3)}) = \infty$; $X_3^{(3)} = X_2^{(2)} = \left(2\frac{1}{2};\ 3\right)$ и $\xi(G_3^{(3)}) = -5$; $G_4^{(3)} = G_3^{(2)} = \varnothing$ и $\xi(G_4^{(3)}) = \infty$.

Дерево решений приведено на рис. 4.5.

На рис. 4.6 приведены графический способ решения данной задачи, а также графическая иллюстрация влияния вводимых ограничений на допустимое множество решений.

Итак, получен целочисленный план $X_1^{(3)} = (3;2)$, причем $\xi = \min\{\xi(G_1^{(3)}); \xi(G_2^{(3)}); \xi(G_3^{(3)}); \xi(G_4^{(3)})\} = \min\{-5; \infty; -5; \infty\} = -5$.

План $X_1^{(3)} = \{3, 2\}$ — оптимальный, так как $f(X) = -5 \leqslant \min\{\xi(G_1^{(3)}); \xi(G_2^{(3)}); \xi(G_3^{(3)})\}$.

Целевая функция $L_{\min} = -5$.

Метод ветвей и границ для задачи о коммивояжере

Постановка задачи о коммивояжере такова. Имеется n городов $A_1, A_2, \ldots, A_n$. Задана матрица расстояний между ними $C = \| c_{ij} \|$. Примем в общем случае, что $c_{ij} \neq c_{ji}$.

Необходимо отыскать такой кратчайший замкнутый маршрут или цикл $(i_1, i_2, \ldots, i_n)$, проходящий через каждый город один раз, при котором минимизируется суммарная длина пути:

$$l(i_1, i_2, \ldots, i_n) = c_{i_1 i_2} + c_{i_2 i_3} + \cdots + c_{i_n i_1}. \tag{4.12}$$

Описание алгоритма. 1. *Задание множества* $G^{(0)}$, которое состоит из всех циклов (замкнутых маршрутов).

2. *Задание множеств* $G_\nu^{(k)}$ $(\nu = 1, 2, \ldots, r_k;\ k = 1, 2)$, каждое из которых состоит из всех циклов, подчиненных одному из следующих дополнительных условий:

а) из пункта i следует идти непосредственно в пункт j для всех упорядоченных пар (i, j), входящих в некоторое множество $P_\nu^{(k)}$;

б) из пункта i запрещается идти непосредственно в пункт j для всех упорядоченных пар (i, j), входящих в некоторое множество $\overline{P}_\nu^{(k)}$.

3. *Вычисление оценки для* $G^{(0)}$. Рассмотрим некоторый цикл $i_1, i_2, \ldots, i_n$. Пройденное расстояние

$$l(i_1, i_2, \ldots, i_n) = c_{i_1 i_2} + c_{i_2 i_3} + \cdots + c_{i_n i_1}. \tag{4.13}$$

Пусть

$$\min_i c_{ij} = h_i.$$

Тогда

$$c'_{ij} = c_{ij} - h_i \geqslant 0$$

и

$$l(i_1, i_2, \ldots, i_n) = \sum_{i=1}^{n} h_i + c'_{i_1 i_2} + c'_{i_2 i_3} + \cdots + c'_{i_n i_1}.$$

Далее, пусть $\min c'_{ij} = H_j$.

Тогда $c''_{ij} = c'_{ij} - H_j \geqslant 0$

и

$$l(i_1, i_2, \ldots, i_n) = \sum_{j=1}^{n} h_i + \sum_{j=1}^{n} H_j + (c''_{i_1 i_2} + \cdots + c''_{i_n i_1}). \quad (4.14)$$

Описанное выше преобразование матрицы, позволяющее из исходной неотрицательной матрицы **C** получить матрицу **C″**, называется *приведением*, а величина

$$\sum_{i=1}^{n} h_i + \sum_{j=1}^{n} H_j = h_\Sigma \quad (4.15)$$

суммой приводящих констант.

Имеет место следующая теорема [28]. *Оптимальный план задачи о коммивояжере с матрицей* **C″** *является оптимальным и для задачи о коммивояжере с матрицей* **C**. Длина пути $l(i_1, i_2, \ldots, i_n)$, соответствующего матрице **C**, и длина пути l'', соответствующего матрице **C″**, связаны соотношением

$$l(i_1, i_2, \ldots, i_n) = l''(i_1, i_2, \ldots, i_n) + \xi(G^{(0)}), \quad (4.16)$$

где $\xi = h_\Sigma$.

Доказательство теоремы вытекает непосредственно из соотношения (4.14).

4. *Ветвление.* Каждой вершине множества $G_\nu^{(k)}$ дерева решений будет соответствовать своя оценка $\xi(G_\nu^{(k)})$ и своя приведенная матрица $C_\nu^{(k)}$.

Множество $G_\nu^{(k)}$ при ветвлении разбивается на два подмножества. По некоторому способу (указанному ниже) выбирают пару пунктов (r, m), не входящую в множества $P_\nu^{(k)}$ и $\overline{P}_\nu^{(k)}$. После этого производят ветвление:

$$G_\nu^{(k)} = G_{\nu,1}^{(k)} \cup G_{\nu,2}^{(k)}.$$

Здесь $G_{\nu,1}^{(k)}$ получается из $G_\nu^{(k)}$ при добавлении следующего условия: из пункта r следует идти непосредственно в пункт m; $G_{\nu,2}^{(k)}$ получается при добавлении условия: из пункта r запрещается непосредственный переход в пункт m.

Следовательно, $P_{\nu,1}^{(k)} = P_\nu^{(k)} \cup (r, m)$; $\overline{P}_{\nu,1}^{(k)} = \overline{P}_\nu^{(k)}$; $P_{\nu,2}^{(k)} = P_\nu^{(k)}$; $\overline{P}_{\nu,2}^{(k)} = \overline{P}_\nu^{(k)} \cup \overline{(r, m)}$.

Выбор пары (r, m) основан на следующих соображениях. Выбирают $G_{\nu,1}^{(k)}$ так, чтобы это множество с наибольшей вероятностью содержало оптимальный цикл, а $G_{\nu,2}^{(k)}$ — не содержало. Для этого выбирают пару (r, m) так, чтобы выполнялось условие

$$c_\nu^{(k)}(r, m) = 0, \tag{4.17}$$

где $c_\nu^{(k)}(r, m)$ — (r, m)-й элемент приведенной матрицы $\mathbf{C}_\nu^{(k)}$.

Затем необходимо выбрать пару (r, m) при соблюдении условия (4.17) таким образом, чтобы циклам, входящим в множество $G_{\nu,2}^{(k)}$, соответствовали как можно более длинные пути.

По определению множества $G_{\nu,2}^{(k)}$, путь по любому из этих циклов переходит из пункта r в некоторый промежуточный пункт j $(j \neq m)$, а в пункт m попадают из некоторого пункта i $(i \neq r)$. Ясно, что длина этого пути будет не меньше чем

$$\theta(r, m) = \min_{j \neq m} c_\nu^{(k)}(r, j) + \min_{i \neq r} c_\nu^{(k)}(i, m). \tag{4.18}$$

Остается выбрать пару (r, m) так, чтобы $\theta(r, m)$ было максимально, т. е.

$$\theta(r, m) = \max_{p,q} \theta(p, q) = \max_{p,q} \{\min_{j \neq q} c_\nu^{(k)}(p, j) + \min_{i \neq p} c_\nu^{(k)}(i, q)\}. \tag{4.19}$$

При этом необходимо выполнение условия

$$c_\nu^{(k)}(p, q) = 0. \tag{4.20}$$

5. *Преобразование матрицы расстояний при ветвлении. Пересчет оценок.* Рассмотрим ветвление $G_\nu^{(k)} = G_{\nu,1}^{(k)} \cup G_{\nu,2}^{(k)}$. Сначала рассмотрим множество $G_{\nu,2}^{(k)}$ и укажем правило перехода от $\mathbf{C}_\nu^{(k)}$ к $\mathbf{C}_{\nu,2}^{(k)}$.

Матрица $\mathbf{C}_{\nu,2}^{(k)}$ содержит те же строки и столбцы, что и матрица $\mathbf{C}_\nu^{(k)}$. Строим промежуточную матрицу $\tilde{\mathbf{C}}_{\nu,2}^{(k)}$ по правилу:

$$\tilde{c}_{\nu,2}^{(k)}(i, j) = \begin{cases} c_\nu^{(k)}(i, j), & \text{если } (i, j) \neq (r, m); \\ \infty, & \text{если } (i, j) = (r, m). \end{cases}$$

Применив к $\tilde{\mathbf{C}}_{\nu,2}^{(k)}$ процедуру проведения, получают $\mathbf{C}_{\nu,2}^{(k)}$.

При этом сумма приводящих констант равна $\theta(r, m)$. Таким образом,

$$\xi(G_{\nu,2}^{(k)}) = \xi(G_\nu^{(k)}) + \theta(r, m)\cdot \tag{4.21}$$

Теперь рассмотрим множество $G_{\nu,1}^{(k)}$ и определим правило перехода от $\mathbf{C}_\nu^{(k)}$ к $\mathbf{C}_{\nu,1}^{(k)}$.

Так как по определению множество $G_{\nu,1}^{(k)}$ заведомо содержит непосредственный переход из пункта r в пункт m, то при переходе от матрицы $\mathbf{C}_\nu^{(k)}$ к матрице $\mathbf{C}_{\nu,1}^{(k)}$ можно вычеркнуть строку r и столбец m. В результате получают $\tilde{\mathbf{C}}_{\nu,1}^{(k)}$.

Далее следует проверить, не состоит ли $\tilde{\mathbf{C}}_{\nu,1}^{(k)}$ точно из одного цикла. Если это так, то к матрице применяют процесс приведения. Получив сумму приводящих констант $h_\Sigma = \sum_{i=1}^{n} h_i + \sum_{j=1}^{n} H_j$, пересчитывают оценку

$$\xi(G_{\nu,1}^{(k)}) = \xi(G_\nu^{(k)}) + h_\Sigma. \tag{4.22}$$

При этом величина оценки $\xi(G_{v,1}^{(k)})$ равна длине этого единственного цикла.

Если $G_{v,1}^{(k)}$ не содержит точно один цикл, то следует запретить возможность образования подциклов (замкнутых маршрутов), проходящих через количество пунктов меньше n. Для этого необходимо найти все маршруты, которые включают переход (r, m). Обозначим $S_{v,1}^{(k)}$ — множество таких маршрутов, т. е. множество подциклов, содержащих (r, m).

Процесс запрещения образования подциклов выполняют следующим образом.

В множество $S_{v,1}^{(k)}$ входит подцикл (r, m, r). Его запрещают, полагая $\tilde{c}_{v,1}^{(k)}(m, r) = \infty$. Далее, если из элементов $P_v^{(k)}$ можно составить маршрут $(i_1, i_2, \ldots, i_p, r)$, то в множество $S_{v,1}^{(k)}$ входят циклы вида $(i_q, i_{q+1}, \ldots, i_p, r, m, i_q)$, где $1 \leqslant q \leqslant p$. Запрещают эти маршруты, полагая $\tilde{c}_{v,1}^{(k)}(m, i_q) = \infty$.

Если же из элементов множества $P_v^{(k)}$ можно составить маршрут $(m, i_1, i_2, \ldots, i_p)$, то в $S_{v,1}^{(k)}$ входят циклы вида $(r, m, i_1, \ldots, i_p, r)$. Поэтому запрещают эти маршруты, полагая $\tilde{c}_{v,1}^{(k)}(i_q, r) = \infty$, $1 \leqslant q \leqslant p$.

Все остальные элементы матрицы $\tilde{C}_{v,1}^{(k)}$ остаются без изменений.

Таблица 4.17

C =

i \ j	0	1	2	3	4	5
0	×	4	10	13	4	8
1	2	×	9	7	6	7
2	8	5	×	5	5	9
3	5	8	5	×	7	10
4	6	4	4	9	×	4
5	5	1	4	8	3	×

Затем применяют операцию приведения к матрице $\tilde{C}^{(h)}_{v,1}$ и, найдя сумму приводящих констант $h^{(h)}_{\Sigma,v1}$, получают матрицу $\mathbf{C}^{(h)}_{v,1}$. Оценку $\xi\,(G^{(h)}_{v,1})$ определяют по формуле

$$\xi\,(G^{(k)}_{v,1}) = \xi\,(G^{(k)}_{v}) + h^{(k)}_{\Sigma,v1}. \qquad (4.23)$$

Пример 4.7. Решить задачу о коммивояжере, определяемую матрицей **C** (табл. 4.17).

Применив процесс приведения, приходим к матрице $\mathbf{C}_0$ (табл. 4.18):

Таблица 4.18

$\mathbf{C}_0 =$

$i \backslash j$	0	1	2	3	4	5	h_i	α_i
0	×	0	6	9	0	4	4	0
0	0	×	7	5	4	5	2	4
2	3	0	×	$\boxed{0}$	0	4	5	0
3	0	3	0	×	2	5	5	0
4	2	0	0	5	×	0	4	0
5	4	0	3	7	2	×	1	2
H_j	0	0	0	0	0	0		
β_j	0	0	0	5	0	4		

Нулевой шаг. $\xi(G^{(0)}) = \Sigma h_i + \Sigma H_j = 21$.

Затем вычисляем величины α_{i_0} и β_{j_0}: $\alpha_{i_0} = c_{i_0 j_1}$, $\beta_{i_0} = c_{i_1 j_0}$, где $c_{i_0 j_1}$ — наименьший отличный от $c_{i_0 j_0} = 0$ элемент строки i_0, $c_{i_1 j_0}$ — наименьший отличный от $c_{i_0 j_0} = 0$ элемент столбца j_0.

Поскольку $\Theta_{23} = \alpha_2 + \beta_3 = 5$, $\Theta_{10} = \alpha_1 + \beta_0 = 4$, $\Theta_{45} = \alpha_4 + \beta_5 = 4$, $\max\limits_{p,q} \{\Theta_{p,q}\} = \max \{\Theta_{23}; \Theta_{10}; \Theta_{45}\} = 5 = \Theta\,(2,\ 3)$, то в качестве пары, входящей в искомый цикл, нужно выбрать пару (2, 3).

Первый шаг. Производим ветвление $G^{(0)} = G_1^{(1)} \cup G_2^{(1)}$, где $G_1^{(1)}$ такое, что $G_1^{(1)} = \{2, 3\}$, а $G_2^{(1)} = \overline{\{2, 3\}}$.

Вычисляем оценку $\xi(G_2^{(1)}) = \xi(G_0) + \Theta_{23} = 21 + 5 = 26$.

Для вычисления $\xi(G_1^{(1)})$ необходимо построить матрицу $C_1^{(1)}$. Вычеркнем в матрице C_0 вторую строку и третий столбец и, полагая элемент $C_{32} = \infty$, выполним процесс приведения. В результате получим матрицу $C_1^{(1)}$

Таблица 4.19

$C_1^{(1)} =$

i \ j	0	1	2	4	5	h_i	α_i
0	✕	0	6	0	4	0	0
1	0	✕	7	4	5	0	4
3	0	3	✕	2	5	0	2
4	2	0	0	✕	$\boxed{0}$	0	0
5	4	0	3	2	✕	0	2
H_j	0	0	0	0	0		
β_j	0	0	3	2	4		

Найдя $h_{\Sigma 1}^{(1)} = \Sigma h_i + \Sigma H_j = 0$, определим оценку для множества $G_1^{(1)}$: $\xi(G_1^{(1)}) = \xi(G_0) + h_{\Sigma 1}^{(1)} = 21$. Так как $\xi(G_1^{(1)}) < \xi(G_2^{(1)})$, то на следующем шаге разбиваем подмножество $G_1^{(1)}$.

Второй шаг. В качестве пары, по которой производим ветвление, используем пару (4, 5), так как $\Theta(4, 5) = 4 = \max\limits_{(r,m)} \Theta(r, m)$.

Производим ветвление $G_1^{(1)} = G_1^{(2)} \cup G_2^{(2)}$, где $G_1^{(2)} = \{(2, 3); (4, 5)\}$, $G_2^{(2)} = \{2, 3); \overline{(4, 5)}\}$. Вычисляем оценку $\xi(G_2^{(2)}) = \xi(G_{(1)}^{(1)}) + \Theta(4, 5) = 21 + 4 = 25$.

Затем строим матрицу $C_1^{(2)}$. Для этого вычеркиваем четвертую строку и пятый столбец и, полагая $c_{54} = \infty$, выполним процесс приведения. Получим матрицу $C_1^{(2)}$ (табл. 4.20).

Таблица 4.20

$C_1^{(2)} =$

$i \backslash j$	0	1	2	4	h_i	α_i
0	✕	0	3	0	0	0
1	$\boxed{0}$	✕	4	4	0	4
3	0	3	✕	2	0	2
5	4	0	0	✕	0	0
H_j	0	0	3	0		
β_j	0	0	3	0		

Таблица 4.21

$C_1^{(3)} =$

$i \backslash j$	1	2	4	h_i	α_i
0	✕	3	0	0	3
3	1	✕	0	2	1
5	0	0	✕	0	0
H_j	0	0	0		
β_j	1	3	0		

Находим $h_{\Sigma 1}^{(2)} = \Sigma h_{i,1}^{(2)} + \Sigma H_{j,1}^{(2)} = 3$.
Тогда $\xi(G_1^{(2)}) = \xi(G_1^{(1)}) + h_{\Sigma 1}^{(2)} = 21 + 3 = 24$.
Так как $\xi(G_1^{(2)}) < \xi(G_2^{(2)})$, то на очередном шаге производим ветвление множества $G_1^{(2)}$.

Третий шаг. На этом шаге выбираем пару (1,0), так как $\Theta(1, 0) = 4 = \max \Theta(p, q)$ (см. табл. 4.20).

Производим ветвление $G_1^{(2)} = G_1^{(3)} \cup G_2^{(3)}$, где $G_1^{(3)} = \{(2, 3); (4, 5); (1,0)\}$, $G_2^{(3)} = \{(2, 3); (4, 5); \overline{(1, 0)}\}$, $\xi(G_2^{(3)}) = \xi(G_1^{(2)}) + \Theta(1, 0) = 24 + 4 = 28$.

Строим матрицу $C_1^{(3)}$ (табл. 4.21):

$$h_{\Sigma 1}^{(3)} = 2; \quad \xi(G_1^{(3)}) = 24 + 2 = 26.$$

Так как $\xi(G_1^{(3)}) > \xi(G_2^{(3)})$, то производим ветвление множества $G_2^{(2)}$.

Четвертый шаг. Строим матрицу $C_2^{(2)}$, для этого в матрице $C_1^{(1)}$ полагаем $c_{45} = \infty$ и выполняем процесс приведения:

$$\xi(G_2^{(2)}) = \xi(G_1^{(1)}) + h_{\Sigma 2}^{(2)} = 21 + 4.$$

Таблица 4.22

$C_2^{(2)} =$

i	0	1	2	4	5	h_i	α
0	×	0	6	0	0	0	0
1	0	×	7	4	1	0	1
3	0	3	×	2	1	0	1
4	2	0	$\boxed{0}$	×	∞	0	0
5	4	0	3	2	×	0	2
H_j	0	0	0	0	4		
β_j	0	0	3	2	1		

Таблица 4.23

$C_1^{(4)} =$

i \ j	0	1	4	5	h_i	α_i
0	×	0	0	0	0	0
1	0	×	4	1	0	1
3	0	3	×	1	0	1
5	4	$\boxed{0}$	2	×	0	2
H_j	0	0	0	0		
β_j	0	0	2	1		

Таблица 4.24

$C_1^{(5)} =$

i \ j	0	4	5	h_i	α_i
0	×	0	0	0	0
1	$\boxed{0}$	4	×	0	4
3	0	×	1	0	1
H_j	0	0	0		
β_j	0	4	1		

Таблица 4.25

$C_1^{(6)} =$

i \ j	4	5	h_i	α_i
0	0	×	0	
3	×	0	0	
H_j	0	1		
β_j				

Выбираем пару (4, 2). Производим ветвление $G_2^{(2)} = G_1^{(4)} \cup G_2^{(4)}$, где $G_1^{(4)} = G_2^{(2)} \cup (4, 2)$, $G_2^{(4)} = G_2^{(2)} \cup \overline{(4,2)}$.

$$\xi(G_2^{(4)}) = 25 + 3 = 28.$$

Вычислим оценку $\xi(G_1^{(4)})$, для чего строим матрицу $C_1^{(4)}$ (табл. 4.23).

Так как $\Sigma h_i + \Sigma H_j = 0$, то $\xi(G_1^{(4)}) = \xi(G_2^{(2)}) = 25$.

Поскольку $\xi(G_1^{(4)}) < \xi(G_2^{(4)})$, то на следующем шаге производим ветвление $G_1^{(4)}$.

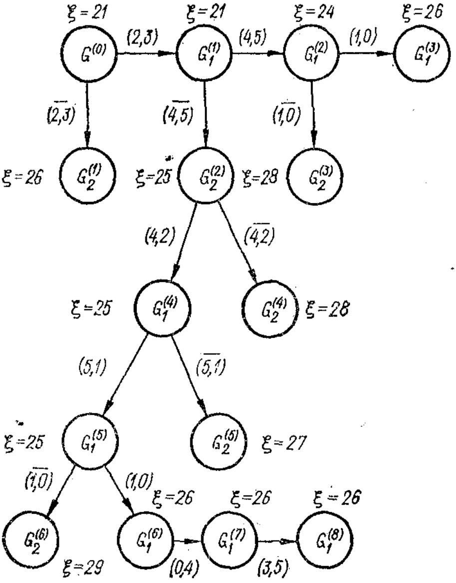


Рис. 4.7.

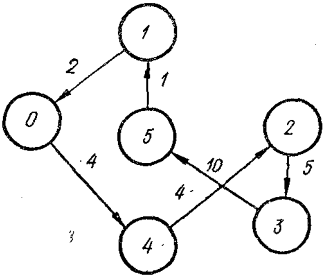


Рис. 4.8.

Пятый шаг. Выбираем пару (5, 1). Θ (5, 1) = 2. Производим ветвление $G_1^{(4)} = G_1^{(5)} \cup G_2^{(5)}$, где $G_1^{(5)} = G_1^{(4)} \cup (5, 1)$, $G_2^{(5)} = G_1^{(4)} \cup \overline{(5, 1)}$; $\xi(G_2^{(5)}) = \xi(G_1^{(4)}) + \Theta(5, 1) = 25 + 2 = 27$.

Строим матрицу $C_1^{(5)}$. Так как $\sum_i h_i + \sum_j H_j = 0$, то $\xi(G_1^{(5)}) = \xi(G_1^{(4)}) = 25$.

Шестой шаг. Выбираем очередную пару (1, 0), так как $\Theta(1, 0) = 4 = \max \Theta(p, q)$. Производим ветвление $G_1^{(5)} = G_1^{(6)} \cup G_2^{(6)}$, где $G_1^{(6)} = G_1^{(5)} \cup (1, 0)$; $G_2^{(6)} = G_1^{(5)} \cup \overline{(1, 0)}$.

Находим оценки для образованных множеств:

$$\xi(G_2^{(6)}) = 25 + 4 = 29.$$

Далее строим матрицу $C_1^{(6)}$ (табл. 4.25)

$$\xi(G_1^{(6)}) = 25 + 1 = 26.$$

Из матрицы $C_1^{(6)}$ выбираем две последние пары (0, 4) и (3, 5).

Последние два шага выполняем аналогичным образом.

Получаем цикл, отвечающий множеству $G_1^{(8)}$:(2, 3), (4, 2), (5, 1), (1, 0), (0, 4), (3, 5).

Длина цикла равна оценке для множества $G_1^{(8)} : l = \xi(G_1^{(8)}) = 26$. В этом можно убедиться непосредственной проверкой. Поскольку оценка $\xi(G_1^{(8)})$ для множества $G_1^{(8)}$ не превышает оценок для всех остальных вершин, которые отвечают отброшенным вариантам, то найденный цикл является искомым.

Процесс построения дерева решений приведен на рис. 4.7, а искомый цикл — на рис. 4.8.

Вопросы для самопроверки

1. Перечислите основные математические модели дискретного программирования.
2. Что называется правильным отсечением, как оно формируется в методе отсекающих плоскостей?
3. Какой геометрический смысл имеет правильное отсечение?
4. Могут ли величины γ_{ij} быть отрицательными, если, например, $x_{ij} < 0$?
5. Что представляет собой полное множество всех отсечений, формируемых по некоторой симплекс-таблице? Как найти мощность этого множества?
6. Почему в методе отсекающих плоскостей используется двойственный симплекс-метод?
7. Сформулируйте достаточные условия возможности перехода к асимптотической задаче при решении задачи ЛЦП методом Гомори и поясните их смысл.
8. Какова основная идея метода ветвей и границ? В чем состоят особенности его реализации при решении конкретных классов задач (моделей)?
9. Как производятся ветвления множества и вычисляются оценки при решении методом ветвей задачи линейного целочисленного программирования (ЛЦП) на минимум и максимум?
10. Какая аналогия существует между методом Гомори и методом ветвей и границ применительно к задачам ЛЦП?
11. Как производятся ветвления множества и вычисляются оценки при решении методом ветвей и границ задачи о коммивояжере?
12. Укажите основные достоинства и недостатки метода ветвей и границ при решении комбинаторных задач дискретного программирования.

Глава 5. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Задача нелинейного программирования (НП-задача) в общем виде формулируется следующим образом.

Найти

$$\max f(x_1, \ldots, x_n)$$

при условиях

$$\begin{cases} g_1(x_1, \ldots, x_n) \geqslant 0; \\ g_2(x_1, \ldots, x_n) \geqslant 0; \\ g_m(x_1, \ldots, x_n) \geqslant 0, \end{cases}$$

где функции $f_1(x_1, \ldots, x_n)$, $g_i(x_1, \ldots, x_n)$ $i = \overline{1, m}$ в общем случае нелинейны.

В отличие от задач линейного программирования (ЛП) для задач НП общего метода решения нет.

В задаче ЛП допустимое множество R всегда является выпуклым с конечным числом крайних точек. Используя аппарат симплекс-метода и перебрав только крайние точки, всегда за конечное число шагов возможно найти оптимальное решение. Напротив, в НП-задачах, если в ограничениях есть нелинейность, то выпуклость допустимого множества и конечность числа его крайних точек необязательны. Из-за этих особенностей и возникают основные трудности решения НП-задачи.

Рассмотрим следующие примеры:

Пример 5.1. Допустимая область R определяется ограничениями (рис. 5.1):

$$g_1(x_1, x_2) = 6 - x_1 \geqslant 0;$$
$$g_2(x_1, x_2) = 6 - x_2 \geqslant 0;$$
$$g_3(x_1, x_2) = 6 - x_1x_2 \geqslant 0;$$
$$x_1 \geqslant 0; \quad x_2 \geqslant 0.$$

Допустимое множество решений является невыпуклым.

Пример 5.2.

$$g_1(x_1, x_2) = 4 - x_1^2 - x_2^2 \geqslant 0,$$
$$x_1 \geqslant 0; \quad x_2 \geqslant 0.$$

Допустимое множество решений (рис. 5.2) хотя и выпукло, но имеет бесконечное число крайних точек.

Пример 5.3.

$$\max(10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2) = \max f(x_1, x_2).$$

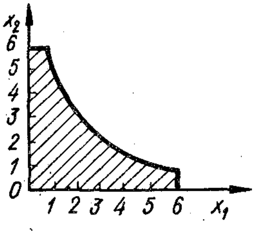


Рис. 5.1.

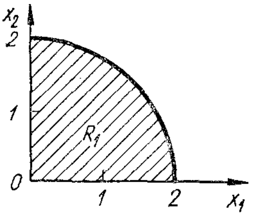


Рис. 5.2.

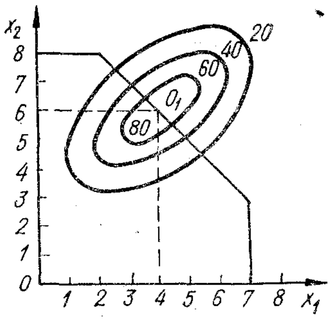


Рис. 5.3.

При ограничениях

$$g_1(x_1, x_2) = 7 - x_1 \geqslant 0;$$
$$g_2(x_1, x_2) = 8 - x_2 \geqslant 0;$$
$$g_3(x_1, x_2) = 10 - x_1 - x_2 \geqslant 0.$$

Построим допустимое множество решений (рис. 5.3).

Задаваясь $f(x_1, x_2) = c$, строим семейство эллипсов с общими осями. Из рисунка следует, что точка абсолютного максимума ($x_1^0 = 4$, $x_2^0 = 6$) попала на границу множества решений.

§ 1. КЛАССИЧЕСКИЙ МЕТОД ОПРЕДЕЛЕНИЯ УСЛОВНОГО ЭКСТРЕМУМА

Для определения условного экстремума (т. е. экстремума при ограничениях) могут быть использованы методы дифференциального исчисления, когда $f(x_1, ..., x_n)$ имеет не ниже второй производной. Рассмотрим некоторые важные понятия и теоремы классического математического анализа, которые лежат в основе методов поиска условного экстремума.

Теорема 5.1. *Если $f(x_1, ..., x_n)$ — непрерывная функция, определенная на замкнутом и ограниченном множестве R, то она достигает на этом множестве, по крайней мере один раз, максимального и минимального значения (теорема существования экстремума).*

Следующая теорема определяет возможные местоположения максимума.

Теорема 5.2. *Если $f(x_1, ..., x_n)$ является функцией нескольких переменных, определенной на допустимой области R, то максимальное значение f, если оно существует, достигается в одной или нескольких точках, которые принадлежат одному из следующих множеств:*

1) S_1 — множество стационарных точек;
2) S_2 — множество точек границы;
3) S_3 — множество точек, где $f(x_1, ..., x_n)$ недифференцируема.

Определение 1. *Множество точек S_1 (x) функции $f(x_1, x_2, ..., x_n)$ называется множеством стационарных точек, если они удовлетворяют условию*

$$\frac{\partial f(x)}{\partial x_j} = 0; \quad j = \overline{1, n}. \qquad (1.1)$$

Определение 2. *Функция f достигает относительного максимума в точке $(x_1^0, x_2^0, \ldots, x_n^0)$, если для всех точек R, лежащих в малой окрестности точки $(x_1^0, x_2^0, \ldots, x_n^0)$, имеет место неравенство*

$$f(x_1^0, x_2^0, \ldots, x_n^0) \geqslant f(x_1, x_2, \ldots, x_n). \qquad (1.2)$$

Определение 3. *Функция f достигает абсолютного максимума в точке $(x_1^0, x_2^0, \ldots, x_n^0)$, если для всех точек $\{x_1, \ldots, x_n\} \in R$ справедливо неравенство*

$$f(x_1^0, x_2^0, \ldots, x_n^0) \geqslant f(x_1, x_2, \ldots, x_n).$$

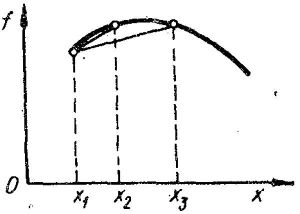


Рис. 5.4.

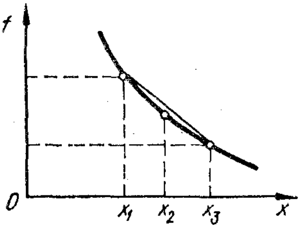


Рис. 5.5.

Для нахождения стационарных точек допустимой области R можно использовать теорему 5.3.

Теорема 5.3. *Пусть $f(x_1, x_2, \ldots, x_n)$ — дифференцируема в некоторой области R. Если в некоторой внутренней точке $(x_1^0, x_2^0, \ldots, x_n^0)$ области R функция f имеет относительный максимум, то*

$$\frac{\partial f(x_1^0, x_2^0, \ldots, x_n^0)}{\partial x_i} = 0, \quad i = 1, 2, \ldots, n. \qquad (1.3)$$

Пример 5.4. Пусть $f(x_1, x_2) = 10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2$ определена на множестве $R \times R$ (т. е. на всей плоскости x_1Ox_2).

Для определения относительного экстремума этой функции имеем два уравнения:

$$\frac{\partial f}{\partial x_1} = 10 + x_2 - 4x_1 = 0,$$

$$\frac{\partial f}{\partial x_2} = 20 + x_1 - 4x_2 = 0,$$

решив которые, находим $x_1^0 = 4$, $x_2^0 = 6$.

Для того чтобы определить, являются ли найденные стационарные точки точками максимума или минимума, необходимо исследовать $f(x_1, x_2, \ldots, x_n)$ в окрестности стационарных точек и определить, является она выпуклой или вогнутой.

Определение 4. *Пусть R — выпуклое множество точек n-мерного пространства. Функция f, определенная на R, называется вогнутой (выпуклой вверх), если для любой пары точек $\mathbf{x}_1, \mathbf{x}_2 \in R$*

и произвольного $0 \leqslant k \leqslant 1$ *выполняется неравенство* (рис. 5.4)

$$f\,[(k\mathbf{x}_1 + (1 - k)\,\mathbf{x}_2] \geqslant kf\,(\mathbf{x}_1) + (1 - k)\,f\,(\mathbf{x}_2). \tag{1.4}$$

Если

$$f\,[k\,(\mathbf{x}_1) + (1 - k)\,\mathbf{x}_2] \leqslant kf\,(\mathbf{x}_1) + (1 - k)\,f\,(\mathbf{x}_2), \tag{1.5}$$

то функция называется выпуклой (рис. 5.5).

Если в (1.4) или (1.5) есть строгие неравенства, то функция называется строго вогнутой или строго выпуклой соответственно.

Критерий выпуклости и вогнутости функции n переменных может быть сформулирован в виде следующей теоремы.

Теорема 5.4. *Дифференцируемая функция* $f\,(\mathbf{x})$ *строго вогнута в некоторой окрестности точки* $\mathbf{x}_0 = (x_1^0, x_2^0, \ldots, x_n^0)$, *если выполняются следующие условия:*

$$f_{11}(\mathbf{x}_0) < 0 \qquad \begin{vmatrix} f_{11}(\mathbf{x}_0) & f_{12}(\mathbf{x}_0) \\ f_{21}(\mathbf{x}_0) & f_{22}(\mathbf{x}_0) \end{vmatrix} > 0; \qquad \begin{vmatrix} f_{11}(\mathbf{x}_0) & f_{12}(\mathbf{x}_0) & f_{13}(\mathbf{x}_0) \\ f_{21}(\mathbf{x}_0) & f_{22}(\mathbf{x}_0) & f_{23}(\mathbf{x}_0) \\ f_{31}(\mathbf{x}_0) & f_{32}(\mathbf{x}_0) & f_{33}(\mathbf{x}_0) \end{vmatrix} < 0,$$

т. е. если знаки определителей чередуются, где

$$f_{ij}(\mathbf{x}_0) = \left.\frac{\partial^2 f\,(\mathbf{x})}{\partial x_i \partial x_j}\right|_{(\mathbf{x}=\mathbf{x}_0)}. \tag{1.6}$$

Функция $f\,(\mathbf{x})$ строго выпукла в окрестности точки $\mathbf{x}_0$, если все определители (выписанные выше) положительны. Имеет место следующая теорема.

Теорема 5.5. *Для того чтобы в точке* $\mathbf{x}_0$ *достигался внутренний относительный максимум, достаточно равенства нулю всех первых частных производных и строгой вогнутости функций в окрестности* $\mathbf{x}_0$.

Для того чтобы в $\mathbf{x}_0$ был относительный минимум, достаточно, чтобы все частные производные обращались в 0 в точке $\mathbf{x}_0$, а сама функция в ее окрестности была строго выпукла.

Пример 5.5. $f\,(x_1, x_2) = 10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2$. Стационарная точка $\mathbf{x}_0 = [4,6]$. Исследуем точку на относительный максимум или минимум:

$$f_{11}(x_1, x_2) = -4, \; f_{12} = 1, \; f_{21} = 1, \; f_{22}(x_1, x_2) = -4.$$

Так как

$$f_{11} = -4 < 0, \; \begin{vmatrix} f_{11}(4,6) & f_{12}(4,6) \\ f_{21}(4,6) & f_{22}(4,6) \end{vmatrix} = \begin{vmatrix} -4 & 1 \\ 1 & -4 \end{vmatrix} = 15 > 0,$$

то функция f достигает в точке $x_1^0 = 4$, $x_2^0 = 6$ относительного максимума.

Справедливо следующее утверждение. Если $f\,(\mathbf{x})$ строго выпуклая (вогнутая) функция на всем множестве R, то f обладает только одним относительным минимумом (максимумом), который является и абсолютным.

Теорема 5.6 (*о выпуклости допустимого множества решений*). *Пусть* $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, ..., $g_m(\mathbf{x}) \geqslant 0$ *и* $\mathbf{x} \geqslant 0$ — *ограничения задачи нелинейного программирования. Если функции* g_1, g_2, ..., g_m *вогнуты,* *то допустимое множество* $R = \{\mathbf{x} : \mathbf{x} \geqslant 0$ *и* $g_i(\mathbf{x}) \geqslant 0$ $(i = 1, 2, \dots, m)\}$ *является выпуклым.*

Доказательство. Для доказательства достаточно показать, что множество $R_i = \{\mathbf{x} : g_i(\mathbf{x}) \geqslant 0;\ \mathbf{x} \geqslant 0\}$ при каждом $i = 1, 2, \dots, m$ будет выпуклым. Тогда $R = R_1 \cap R_2 \cap \dots \cap R_m$ будет также выпуклым, так как пересечение конечного числа выпуклых множеств выпукло. Рассмотрим некоторую вогнутую функцию $g_i(\mathbf{x}) \geqslant 0$. Выберем две произвольные точки $\mathbf{x}_1$ и $\mathbf{x}_3 \geqslant 0$ (рис. 5.6). Тогда

$$\mathbf{x}_2 = \lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_3 \geqslant 0, \quad 0 < \lambda < 1.$$

Так как $\mathbf{x}_1 \in R_i$ и $\mathbf{x}_3 \in R_i$, то и точка $\mathbf{x}_2$ принадлежит R_i. Из условия вогнутости g_i следует, что

$$g_i[\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_3] \geqslant \lambda g_i(\mathbf{x}_1) + (1 - \lambda) g_i(\mathbf{x}_3) \geqslant 0.$$

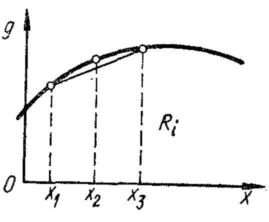


Рис. 5.6.

Следовательно, множество R_i содержит отрезок $\lambda g_i(x_1) + (1 - \lambda) \times g_i(x_3)$, а поэтому R_i — выпукло (рис. 5.6).

Справедливо следующее утверждение.

Если функции $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})$ *вогнуты (выпуклы) на множестве* R, *то функция* $g(\mathbf{x}) = \sum_{i=1}^{p} k_i f_i(\mathbf{x})$ *также вогнута (выпукла), при условии, что* $k_i \geqslant 0$, $i = 1, 2, \dots, p$. Его доказательство предоставляется читателю.

Рассмотрим классический метод поиска условного экстремума. Он состоит в следующем.

1) Отыскивают множество всех стационарных точек $S_1(\mathbf{x})$ функции $f(\mathbf{x})$ внутри допустимого множества R. Найденные точки далее исследуют на максимум (минимум) и определяют точку наибольшего максимума $\mathbf{x}_0 \in S_1(\mathbf{x})$.

2) Переходят к исследованию точек границы $S_2(\mathbf{x})$ и отысканию тех из них, где $f(\mathbf{x})$ достигает максимума. Этот процесс состоит в следующем. Выбирают произвольную границу, определяемую, например, условием $g_1(\mathbf{x}) = 0$.

Если функция

$$g_i(\mathbf{x}) = g_i(x_1, x_2, \dots, x_n) = 0 \qquad (1.7)$$

с разделяющимися переменными, то всегда можно, определив из (1.7) переменную

$$x_i = \varphi_i(\{x_j\}),\ j = \overline{1, n},\ j \neq i), \qquad (1.8)$$

подставить ее в выражение для $f(\mathbf{x})$. Тем самым задача сведется к поиску безусловного экстремума, для чего используется процедура, описанная в п. 1. Обозначим через $\mathbf{x}_i^*$ точку границы $g_i(\mathbf{x}) = 0$, $\mathbf{x}_i^* \in R$, в которой $f(\mathbf{x})$ достигает максимума. Повторив вышеописанную процедуру по всем остальным границам, найдем соответственно экстремальные точки всех границ $\mathbf{x}_i^*$, $i = 1, m$.

3) Непосредственным сравнением значений $f(\mathrm{x})$ для всех точек x_i^* и стационарной внутренней точки x_0 определяют точку абсолютного максимума $\mathrm{x}_{\mathrm{opt}}$ на множестве решений R. Такой прямолинейный подход требует больших вычислительных затрат и применим лишь в простейших случаях, при небольшом числе ограничений и если функции $g_i(\mathrm{x})$ — с разделяющимися переменными. Поэтому ниже рассматриваются более эффективные методы решения задач условной оптимизации.

§ 2. МЕТОД МНОЖИТЕЛЕЙ ЛАГРАНЖА

Метод множителей Лагранжа позволяет отыскивать максимум (или минимум) функции при ограничениях-равенствах. Основная идея метода заключается в переходе от задачи на условный экстремум к задаче отыскания безусловного экстремума некоторой специально построенной *функции Лагранжа*.

Пусть требуется найти

$$\min f(x_1, x_2, \ldots, x_n) \tag{2.1}$$

при ограничениях

$$\begin{aligned} &h_1(x_1, \ldots, x_n) = 0, \\ &h_2(x_1, \ldots, x_n) = 0, \\ &\ldots\ldots\ldots\ldots \\ &h_m(x_1, \ldots, x_n) = 0. \end{aligned} \tag{2.2}$$

Предположим, что функции f, h_1 h_2, ..., h_m дифференцируемы. Введем набор переменных $\lambda_1, \lambda_2, \ldots, \lambda_m$ (по числу ограничений), которые называются *множителями Лагранжа*, и составим так называемую функцию Лагранжа следующего вида:

$$L(x_1, x_2, \ldots, x_n, \lambda_1, \ldots, \lambda_m) =$$

$$= f(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{m} \lambda_i h_i(x_1, x_2, \ldots, x_n). \tag{2.3}$$

Тогда для того, чтобы вектор $\mathrm{x}^0 = \{x_1^0, \ldots, x_n^0\}$ являлся решением задачи (2.1) при ограничениях (2.2), необходимо существование такого $\lambda^0 = \{\lambda_1^0, \ldots, \lambda_m^0\}$, что пара векторов $\{\mathrm{x}^0, \lambda^0\}$ удовлетворяет системе уравнений:

$$\frac{\partial L(\mathrm{x}^0, \lambda^0)}{\partial x_j} = 0, \quad j = \overline{1, n}, \tag{2.4}$$

$$\frac{\partial L(\mathrm{x}^0, \lambda^0)}{\partial \lambda_i} = 0, \quad i = \overline{1, m}. \tag{2.5}$$

Покажем необходимость условий (2.5) и (2.4) для следующего простого примера.

Найти

$$\min_{x_1, x_2, x_3} f(x_1, x_2, x_3) \tag{2.6}$$

при условиях

$$h_1(x_1, x_2, x_3) = 0, \; h_2(x_1, x_2, x_3) = 0. \tag{2.7}$$

Ограничения (2.7) определяют собой допустимую область S, которая представляет собой кривую в $R^{(3)}$ и определяется пересечением $h_1(\mathrm{x})$ и $h_2(\mathrm{x})$.

Допустим, что данная задача имеет точку минимума в S_1: $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$, функции f, h_1, h_2 обладают непрерывными производными первого порядка на некотором открытом множестве и градиенты

$$\nabla h_1 = \left[\frac{\partial h_1}{\partial x_1} \frac{\partial h_1}{\partial x_2} \frac{\partial h_1}{\partial x_3}\right]^T, \quad \nabla h_2 = \left[\frac{\partial h_2}{\partial x_1} \frac{\partial h_2}{\partial x_2} \frac{\partial h_2}{\partial x_3}\right]^T$$

линейно-независимы.

Если две переменные в уравнениях (2.7) можно выразить через третью в виде $x_2 = u(x_1)$ и $x_3 = v(x_1)$, то, подставляя их в целевую функцию (2.6), преобразуем исходную задачу в следующую задачу без ограничений, которая содержит одну-единственную переменную x_1:

$$\min f(x_1, u(x_1), v(x_1)). \tag{2.8}$$

Так как градиенты $\nabla h_i(x_1, x_2, x_3)$, $i = 1, 2$ предполагаются непрерывными и линейно-независимыми, то можно применить известную из анализа теорему о неявной функции и найти стационарную точку x_1^*, а затем и $x_2^* = u(x_1^*)$ и $x_3^* = v(x_1^*)$.

Указанный подход можно в принципе распространить и на случай функций n переменных $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ при наличии m ограничений-равенств:

$$h_1(\mathbf{x}) = 0, \; h_2(\mathbf{x}) = 0, \; \ldots, \; h_m(\mathbf{x}) = 0. \tag{2.9}$$

Если функции $h_1, \ldots, h_m$ удовлетворяют условиям теоремы о неявной функции, то m из n переменных уравнений (2.9) можно выразить через остальные $(n - m)$ переменных, подставить их в $f(\mathbf{x})$ и таким образом преобразовать задачу минимизации с ограничениями в задачу безусловной минимизации с $(n - m)$ переменными. Однако такой подход очень трудно реализовать на практике, так как очень трудно разрешить уравнения (2.9) относительно некоторых m переменных.

Поэтому рассмотрим другой подход, использующий метод множителей Лагранжа.

Пусть x_1^* — точка минимума, определяемого уравнением (2.8). Согласно известной теореме анализа о неявной функции можно записать

$$\frac{df}{dx_1} = \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2}\frac{du}{dx_1} + \frac{\partial f}{\partial x_3}\frac{dv}{dx_1} = 0. \tag{2.10}$$

Аналогичные соотношения получаем для ограничивающих уравнений:

$$\frac{\partial h_i}{\partial x_1} + \frac{\partial h_i}{\partial x_2}\frac{du}{dx_1} + \frac{\partial h_i}{\partial x_3}\frac{dv}{dx_1} = 0, \quad i = 1, 2. \tag{2.11}$$

Запишем уравнения (2.10) и (2.11) совместно в виде

$$\mathbf{A} \times \begin{bmatrix} 1 \\ \frac{du}{dx} \\ \frac{dv}{dx_1} \end{bmatrix} = 0, \tag{2.12}$$

где

$$\mathbf{A} = [\nabla f(\mathbf{x}^*) \quad \nabla h_1(\mathbf{x}^*) \quad \nabla h_2(\mathbf{x}^*)].$$

Так как вектор $\left[1 \ \frac{du}{dx_1} \ \frac{dv}{dx_1}\right]$ не является нулевым, то из (2.12) следует, что $\det \mathbf{A} = 0$. Отсюда следует, что вектор-столбцы матрицы $\mathbf{A}$ линейно-зависимы. Следовательно, существуют три такие скаляра a, b и c, не все равные 0, что

$$a\nabla f(\mathbf{x}^*) + b\nabla h_1(\mathbf{x}^*) + c\nabla h_2(\mathbf{x}^*) = \mathbf{0}. \qquad (2.13)$$

Постоянная a не может равняться 0, так как согласно предположению ∇h_1 и ∇h_2 — линейно-независимы. Поэтому после деления (2.13) на a приходим к (2.14):

$$\nabla f(\mathbf{x}^*) + \lambda_1 \nabla h_1(\mathbf{x}^*) + \lambda_2 \nabla h_2(\mathbf{x}^*) = \mathbf{0}. \qquad (2.14)$$

Таким образом, для задачи минимизации с ограничениями (2.6) существуют такие λ_1 и λ_2, для которых справедливо (2.14) и которые одновременно не обращаются в 0. Итак, справедливость условий (2.4) для случая $n = 3$ показана.

Следовательно, для отыскания минимума (2.6) при условиях (2.7) необходимо отыскивать критическую точку функции Лагранжа $L(\mathbf{x}, \Lambda) = f(x) + \lambda_1 h_1(x) + \lambda_2 h_2(x)$. Для того чтобы найти искомые λ_1, λ_2 и $\mathbf{x}^*$, решается совместно система уравнений (2.5) и (2.14).

С геометрической точки зрения условие (2.14) означает, что $\nabla f(\mathbf{x}^*)$ лежит в плоскости, натянутой на векторы $\nabla h_i(\mathbf{x}^*)$.

Рассмотрим теперь общий случай произвольного n. Пусть задача нелинейного программирования задана в виде (2.1), (2.2), все функции $f(\mathbf{x})$, $h_i(\mathbf{x})$ $i = \overline{1, m}$ $(m < n)$ — вещественные функции на множестве $R^{(n)}$, имеющие непрерывные частные производные. Пусть S — подмножества множества $R^{(n)}$, на котором все функции $h_i(\mathbf{x}) = 0$, $i = \overline{1, m}$, т. е.

$$S = (\mathbf{x} : h_i(\mathbf{x}) = 0, \quad i = \overline{1, m}).$$

Тогда справедлива следующая теорема о множителях Лагранжа.

Теорема 5.7. *Допустим, что существует такая точка* $\mathbf{x}^*$, *в которой достигается относительный экстремум* (2.1) *при условиях* (2.2). *Если ранг матрицы* $\mathbf{I} = \left[\frac{\partial h_i}{\partial x_j}\right]_{j=\overline{1, n}}^{i=\overline{1, m}}$ *в точке* $\mathbf{x}^*$ *равен* m, *то существуют* m *вещественных чисел* $\lambda_1, \lambda_2, \ldots, \lambda_m$, *не все из которых равны нулю одновременно, при которых*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) = \mathbf{0}. \qquad (2.15)$$

Заметим, что если ввести функцию Лагранжа $L(\mathbf{x}, \Lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \times$ $\times h_i(\mathbf{x})$, то данная теорема определяет необходимые условия, при которых задача (2.1), (2.2) может быть сведена к нахождению решения уравнения $\nabla L(\mathbf{x}, \Lambda) = \mathbf{0}$.

На основании вышеизложенного метод множителей Лагранжа можно сформулировать следующим образом:

1. Составляют функцию Лагранжа $L(\mathbf{x}, \lambda)$.

2. Находят частные производные: $\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j}$, $\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda_i}$ $j = \overline{1, n}$; $i = \overline{1, m}$.

3. Решают систему

$$\begin{cases} \frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} = 0, \quad j = \overline{1, n} \\ \frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda_i} = h_i(x) = 0, \quad i = \overline{1, m} \end{cases} \qquad \textbf{(2.16)}$$

и отыскивают точки $\mathbf{x}^0 = \{x_1^0, \ldots, x_n^0\}$, удовлетворяющие системе (2.16).

Найденные точки исследуют далее на минимум (или максимум).

Пример 5.6. Имеется два способа производства некоторого продукта. Обозначим через y_1, y_2 количество продукта, произведенного первым или вторым способом соответственно. Издержки производства H при каждом способе зависят от произведенных y_1, y_2 следующим образом:

$$H_1(y_1) = a_0 + a_1 y_1 + a_2 y_1^2 \qquad a_0,\ a_1,\ a_2 > 0;$$

$$H_2(y_2) = b_0 + b_1 y_2 + b_2 y_2^2 \qquad b_0,\ b_1,\ b_2 > 0.$$

За некоторый промежуток времени необходимо произвести ровно c единиц продукции (т. е. $y_1 + y_2 = c$), распределив ее между двумя способами так, чтобы минимизировать общие издержки. Составим функцию Лагранжа для этой задачи

$$L(y_1, y_2, \lambda) = a_0 + a_1 y_1 + a_2 y_1^2 + b_0 + b_1 y_2 + b_2 y_2^2 + \lambda(c - y_1 - y_2),$$

откуда

$$L_{y_1} = a_1 + 2a_2 y_1 - \lambda = 0;$$

$$L_{y_2} = b_1 + 2b_2 y_2 - \lambda = 0;$$

$$L_\lambda = c - y_1 - y_2 = 0.$$

Решая эту систему, находим искомые количества y_1^0, y_2^0:

$$y_1^0 = \frac{b_2}{a_2 + b_2} c + \frac{b_1 - a_1}{2(a_2 + b_2)};$$

$$y_2^0 = \frac{a_2}{a_2 + b_2} c - \frac{b_1 - a_1}{2(a_2 + b_2)}.$$

§ 3. ЗАДАЧА НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ПРИ ОГРАНИЧЕНИЯХ-НЕРАВЕНСТВАХ

Рассмотрим теперь случай задачи с ограничениями типа неравенств:

$$\min f(\mathbf{x}) \qquad \textbf{(3.1)}$$

$$g_i(\mathbf{x}) \leqslant 0, \quad i = \overline{1, m}. \qquad \textbf{(3.2)}$$

В точке минимума $\mathbf{x}^*$ неравенства $g_i(\mathbf{x})$ могут выполняться как равенства либо как строгие неравенства.

Ограничение $g_i(\mathbf{x})$ называется *активным* в точке $\mathbf{x}^*$, если оно выполняется в ней как строгое равенство, т. е. если $g_i(\mathbf{x}^*) = 0$.

Используя геометрические свойства допустимой области, найдем необходимые условия экстремума для задач минимизации с ограничивающими неравенствами. Для этого сначала рассмотрим случай, когда все $g_i(\mathbf{x})$ линейны. Итак, пусть требуется найти при условиях

$$\min f(\mathbf{x})$$
$$g_i(\mathbf{x}) = -\eta^T\mathbf{x} + b_i \leqslant 0,$$
$$i = \overline{1, m}. \qquad (3.3)$$

Здесь каждое ограничивающее уравнение определяет полупространство в R^n. Допустимая область S задана пересечением m полупространств, определяемых m уравнениями (3.3) и, следовательно, является выпуклым многогранником. Вектор η_i является нормалью к гиперплоскости, определяемой уравнением $g_i(\mathbf{x}) = 0$, и направлен внутрь области S.

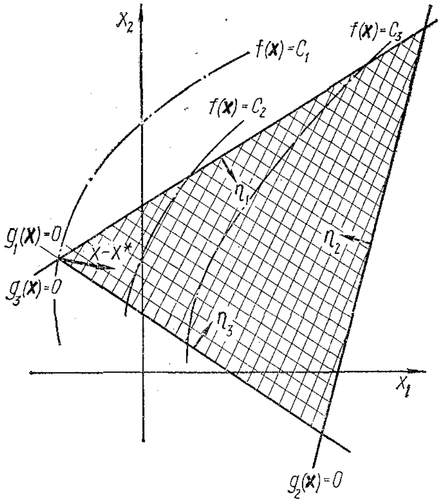


Рис. 5.7.

Пусть $\mathbf{x}^*$ является точкой минимума задачи (3.1) с ограничениями (3.3). Обозначим множество индексов активных ограничений

$$I = \{i : g_i(\mathbf{x}) = 0\}. \qquad (3.4)$$

Например, на рис. 5.7 приведен пример минимизации с линейными ограничениями при $n = 2$ и $m = 3$, $I = \{1, 3\}$.

Выберем любую допустимую точку $\mathbf{x}$ из S. Вектор $\mathbf{x}-\mathbf{x}^*$ направлен из $\mathbf{x}^*$ внутрь области S. Такой вектор будем называть входящим вектором. Для этого вектора с учетом того, что $\eta_i = -\nabla g_i(\mathbf{x}^*)$, можно записать следующее условие:

$$\eta_i^T(\mathbf{x}-\mathbf{x}^*) \geqslant 0 \text{ или } \nabla g_i^T(\mathbf{x}^*)(\mathbf{x}-\mathbf{x}^*) \leqslant 0 \qquad (3.5)$$

для всех $i \in I$ и $\mathbf{x} \in S$.

Таким образом, входящий вектор $\mathbf{x}$ определяет допустимое направление перемещения из точки $\mathbf{x}^*$. Но так как $f(\mathbf{x})$ минимальна в точке $\mathbf{x}^*$, то при любом $\mathbf{x}-\mathbf{x}^*$, удовлетворяющем (3.5), будем иметь

$$\nabla^T f(\mathbf{x}^*)(\mathbf{x}-\mathbf{x}^*) \geqslant 0. \qquad (3.6)$$

Применим теперь лемму Фаркаша (см. приложение 1).

Из условий (3.5) и (3.6) на основании леммы Фаркаша следует, что существует множество таких неотрицательных скаляров $\{\lambda_i\}$, для которых

$$\nabla f(\mathbf{x}^*) = \sum_{i \in I} \lambda_i \eta_i = -\sum_{i \in I} \lambda_i \nabla g_i(\mathbf{x}^*). \qquad (3.7)$$

Отметим, что уравнение (3.7) аналогично (2.15).

Если принять, что $\lambda_j = 0$ при $j \notin I$ (т. е. для неактивных ограничений), то (3.7) можно переписать в виде

$$\nabla f(\mathbf{x}^*) = \sum_{i=1}^{m} \lambda_i \eta_i = -\sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*). \qquad (3.8)$$

Кроме того, получим

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) = 0, \qquad (3.9)$$

поскольку при $i \ni I$, $g_i(\mathbf{x}^*) = 0$, а при $i \notin I$, $\lambda_i = 0$.

Итак, уравнения ограничений могут быть включены в целевую функцию следующим образом:

$$\varphi(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}), \ \lambda_i \geqslant 0. \qquad (3.10)$$

Следовательно, $\mathbf{x}^*$ удовлетворяет следующим условиям:

$$\nabla \varphi(\mathbf{x}^*) = \nabla f(\mathbf{x}^*) + \sum_{i} \lambda_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}, \qquad (3.11)$$

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) = 0, \ \lambda_i \geqslant 0, \ i = \overline{1, m}. \qquad (3.12)$$

При рассмотрении задачи минимизации $f(\mathbf{x})$ при условиях $g_i(\mathbf{x}) \leqslant 0$ может случиться так, что не существует таких λ_i^*, $i = \overline{1, m}$, при которых без дополнительных предположений о природе функций $g_i(\mathbf{x})$ были бы справедливы уравнения (3.11) и (3.12), где $\mathbf{x}^*$ — оптимальное решение. Эти дополнительные предположения называют *условиями регулярности ограничений*. (В частности, в случае ограничений-равенств в качестве таких условий мы использовали линейную независимость векторов-градиентов ограничений).

Теорема Куна — Таккера. Выше мы получили условия оптимальности (3.11) и (3.12) для НП-задачи с линейными ограничениями. Обобщим эти условия на случай задачи (3.1) — (3.2), когда все ограничения нелинейны.

Условия оптимальности решения задачи нелинейного программирования формулируются в следующей теореме Куна — Таккера, имеющей исключительно важное значение для теории нелинейного программирования.

Теорема 5.8. *Пусть f, $g_i(\mathbf{x})$, $i = \overline{1, m}$ обладают непрерывными частными производными на некотором открытом множестве R^n, содержащем $\mathbf{x}^*$. Если $\mathbf{x}^*$ является точкой минимума функции $f(\mathbf{x})$ при ограничениях $g_i(\mathbf{x}) \leqslant 0$, $i = \overline{1, m}$, удовлетворяющих условию регулярности в виде линейной независимости, то существуют такие* *неотрицательные множители Лагранжа* $\lambda_1, \lambda_2, \ldots, \lambda_m$, *что*

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g\ (\mathbf{x}^*) = \mathbf{0}, \tag{3.13}$$

$$\sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) = 0,\ \ \lambda_i \geqslant 0. \tag{3.14}$$

Определим функцию Лагранжа следующим образом:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}). \tag{3.15}$$

Тогда теорему Куна — Таккера можно записать:

$$\nabla_x \mathbf{x}, \lambda) = \mathbf{0}, \tag{3.16}$$

$$\nabla_\lambda L(\mathbf{x}, \lambda = L\, g(\mathbf{x}) \leqslant \mathbf{0}, \tag{3.17}$$

$$\lambda^T \nabla_\lambda L(\mathbf{x}, \lambda) = \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) = 0. \tag{3.18}$$

Заметим, что множители Лагранжа λ_i в НП-задаче с ограничениями-равенствами являются знаконеопределенными, тогда как в теореме Куна — Таккера они должны быть положительными.

Доказательство. При достаточно малых $t > 0$, разлагая $f(\mathbf{x}^* + t\mathbf{z})$ в ряд Тейлора, получим

$$f(\mathbf{x}^* + t\mathbf{z}) = f(\mathbf{x}^*) + t\nabla f^T(\mathbf{x}^*)\,\mathbf{z} + o(t), \tag{3.19}$$

где $o(t)$ — остаточный член 2-го порядка малости (t^2).

Пусть I — множество активных ограничений.

Тогда

$$g_j(\mathbf{x}^* + t\mathbf{z}) = g_j(\mathbf{x}^*) + t\nabla g_j^T(\mathbf{x}^*)\,\mathbf{z} + o(t) = t\nabla g_j^T(\mathbf{x}^*)\,\mathbf{z} + o(t),$$

так как $g_j(\mathbf{x}^*) = 0$ при $j \in I$.

Заметим, что система уравнений

$$\begin{cases} \nabla f^T(\mathbf{x}^*)\,\mathbf{z} < 0, & (3.20) \\ \nabla g_j^T(\mathbf{x}^*)\,\mathbf{z} \leqslant 0 & (3.21) \end{cases}$$

несовместна, так как в противном случае при достаточно малом $t > 0$ для некоторого $\mathbf{z}$ мы бы получили:

$$f(\mathbf{x}^* + t\mathbf{z}) < f(\mathbf{x}^*),$$

$$g_i(\mathbf{x}^* + t\mathbf{z}) \leqslant 0,\quad j = \overline{1, m},$$

что противоречит предположению об оптимальности точки $\mathbf{x}^*$. Для доказательства используем лемму, являющуюся следствием леммы Фаркаша.

Лемма. *При любой матрице* А *выполняется одно из двух условий:*

1. *Либо выполняется следующая система неравенств:*

$$\mathrm{A}\mathbf{x} < \mathbf{0}. \tag{3.22}$$

2. *Либо выполняется следующая система равенств*:

$$\lambda^T A = 0,$$
$$\lambda \geqslant 0, \quad \lambda \neq 0. \tag{3.23}$$

Одновременно условия (3.22) *и* (3.23) *выполняться не могут*. Справедливость леммы следует из леммы Фаркаша (см. приложение 1). Применим эту лемму к (3.20) — (3.21), приняв за матрицу A матрицу

$$\begin{bmatrix} \nabla f(x^*) \\ \nabla g_i(x^*) \end{bmatrix} \quad i \in I.$$

Поскольку система 3.20—3.21 не имеет решений, то существуют такие $\lambda_0, \lambda_1, \ldots, \lambda_m$, что

$$\lambda_0 \nabla f(x^*) + \sum_{i \in I} \lambda_i \nabla g_i(x^*) = 0, \tag{3.24}$$

где $\begin{bmatrix} \lambda_0 \\ \lambda_i \end{bmatrix} \neq 0.$

Если теперь положить, что $\lambda_i = 0$ для $i \notin I$, то получим $\sum_{i=1}^{m} \lambda_i g_i(x^*) = 0$. Условие $\sum_{i=1}^{m} \lambda_i g_i(x^*) = 0$ называют условием *дополняющей нежесткости*.

Покажем, что λ_0 в (3.24) не может быть равно 0. Действительно, если допустить, что $\lambda_0 = 0$, то получим

$$\sum_{i \in I} \lambda_i \nabla g_i(x^*) = 0. \tag{3.25}$$

Но (3.25) противоречит условию теоремы о линейной независимости векторов $\nabla g_i(x^*)$. Остается принять $\lambda_0 \neq 0$. Тогда, разделив обе части (3.24) на λ_0, получим

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i' \nabla g_i(x^*) = 0.$$

Теорема доказана полностью.

Понятие регулярности было введено впервые Куном и Таккером в 1951 году и имеет различные формы. В частном случае, когда $f(x)$ и все $g_i(x)$, $i = \overline{1, m}$ являются выпуклыми функциями, то условие регулярности имеет следующий вид: существует такой вектор x, что $g_i(x) < 0$ для всех $i = \overline{1, m}$. Это условие называют *условием регулярности Слейтера* [3, 24].

Седловая точка и задача нелинейного программирования

Рассмотрим функцию $L(x, \lambda)$ от векторов x и λ.

Определение. *Пара векторов* (x^*, λ^*) *называется седловой точкой функции* $L(x, \lambda)$, *если при всех* $\lambda \geqslant 0$ *и* $x \in R^n$ *выполняется* *условие*

$$L(\mathbf{x}^*, \lambda) \leqslant L(\mathbf{x}^*, \lambda^*) \leqslant L(\mathbf{x}, \lambda^*). \tag{3.26}$$

Неравенство (3.26) называют неравенством для седловой точки. Очевидно, в седловой точке $(\mathbf{x}^*, \lambda^*)$ выполняется условие

$$L(\mathbf{x}^*, \lambda^*) = \max_{\lambda \geqslant 0} \min_{\mathbf{x} \in R^n} L(\mathbf{x}, \lambda) = \min_{\mathbf{x} \in R^n} \max_{\lambda \geqslant 0} L(\mathbf{x}, \lambda). \tag{3.27}$$

Между понятием седловой точки функции Лагранжа $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$ и решением задачи НП имеется взаимосвязь, которая устанавливается в следующей теореме.

Теорема 5.9. *Пусть $f(\mathbf{x})$ и все $g_j(\mathbf{x})$ выпуклы и функции $g_j(\mathbf{x})$ удовлетворяют условию регулярности Слейтера.*

Вектор $\mathbf{x}^*$ *является решением НП-задачи* (3.1), (3.2) *тогда и только тогда, когда существует такой вектор* $\lambda^* \geqslant \mathbf{0}$, *что*

$$L(\mathbf{x}^*, \lambda) \leqslant L(\mathbf{x}^*, \lambda^*) \leqslant L(\mathbf{x}, \lambda^*) \tag{3.28}$$

и

$$\lambda^{*T} g(\mathbf{x}^*) = 0. \tag{3.29}$$

Доказательство. Докажем сначала достаточность условий теоремы. Пусть $(\mathbf{x}^*, \lambda^*)$ — седловая точка функции $L(\mathbf{x}, \lambda)$. Но тогда из правого неравенства (3.26) получим

$$L(\mathbf{x}^*, \lambda^*) = f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* g_i(\mathbf{x}^*) \leqslant L(\mathbf{x}, \lambda^*) = f(\mathbf{x}) + \sum_i \lambda_i^* g_i(\mathbf{x}), \tag{3.30}$$

так как $\lambda_i^* \geqslant 0$, а $g_i(\mathbf{x}) \leqslant 0$, то $\sum_i \lambda_i^* g_i(\mathbf{x}) \leqslant 0$.

Но, с другой стороны, $\sum_i \lambda_i^* g_i^*(x) = 0$ согласно (3.29).

Поэтому из (3.30) следует неравенство

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}) + \sum_i \lambda_i^* g_i(\mathbf{x}) \leqslant f(\mathbf{x}) \tag{3.31}$$

для всех $\mathbf{x}$, удовлетворяющих ограничениям НП-задачи. Таким образом, $\mathbf{x}^*$ — оптимальное решение НП-задачи.

Перейдем к доказательству необходимости.

Допустим, что $\mathbf{x}^*$ — оптимальное решение задачи нелинейного программирования. Заметим, что система

$$\begin{cases} f(\mathbf{x}) - f(\mathbf{x}^*) < 0, \\ g(\mathbf{x}) \leqslant 0 \end{cases} \tag{3.32}$$

не имеет решения, так как $\mathbf{x}^*$ — точка минимума НП-задачи.

Отсюда следует также, что не имеет решения и следующая система:

$$\begin{cases} f(\mathbf{x}) - f(\mathbf{x}^*) < 0, \\ g(\mathbf{x}) < 0. \end{cases} \tag{3.33}$$

Тогда согласно теореме Фана (см. приложение 4) существуют такие λ_0^*, $\lambda_i^* \geqslant 0$, что

$$\lambda_0^* [f(\mathbf{x}) - f(\mathbf{x}^*)] + \sum_i \lambda_i^* g_i(\mathbf{x}) \geqslant 0. \tag{3.34}$$

Так как

$$\lambda_i^* \geqslant 0 \quad \text{и} \quad g_i(\mathbf{x}^*) \leqslant 0,$$

то

$$\sum_i \lambda_i^{T*} g_i(\mathbf{x}) \leqslant 0 \text{ для всех } \mathbf{x}. \tag{3.35}$$

Если же в (3.34) положить $\mathbf{x} = \mathbf{x}^*$, то получим

$$\sum_i \lambda_i^* g_i(\mathbf{x}^*) \geqslant 0. \tag{3.36}$$

Сравнив (3.35) с (3.36), получим

$$\sum_i \lambda_i^* g_i(\mathbf{x}^*) = 0. \tag{3.37}$$

Но тогда из уравнений (3.34) и (3.37) получим, что

$$\lambda_0^* f(\mathbf{x}^*) + \sum_i \lambda_i^* g_i(\mathbf{x}^*) \leqslant \lambda_0^* f(\mathbf{x}) + \sum_i \lambda_i^* g_i(\mathbf{x}). \tag{3.38}$$

Таким образом, доказано правое неравенство для седловой точки. Так как $g(\mathbf{x}^*) \leqslant \mathbf{0}$, то $\lambda^T g(\mathbf{x}^*) \leqslant 0$ при любом $\lambda \geqslant \mathbf{0}$. Следовательно,

$$\lambda_0^* f(\mathbf{x}^*) + \lambda^T g(\mathbf{x}^*) \leqslant \lambda_0^* f(\mathbf{x}^*) = \lambda_0^* f(\mathbf{x}^*) + \lambda^{T*} g(\mathbf{x}^*). \tag{3.39}$$

Разделив обе части (3.39) на $\lambda_0^* > 0$, получим левое неравенство для седловой точки

$$f(\mathbf{x}^*) + \frac{\lambda^T}{\lambda_0^*} g(\mathbf{x}^*) = L(\mathbf{x}^*, \lambda) \leqslant f(\mathbf{x}^*) + \frac{\lambda^{T*}}{\lambda_0^*} g(\mathbf{x}^*) = L(\mathbf{x}^*, \lambda^*).$$

Таким образом, теорема доказана.

Чтобы обеспечить условие $\lambda_0^* > 0$, необходимо предположить существование условия регулярности Слейтера. Действительно, пусть $\lambda_0^* = 0$. Тогда $\lambda_i^* \geqslant 0$ и выражение (3.38) примет вид

$$\sum_i \lambda_i^* g_i(\mathbf{x}^*) = 0 \leqslant \sum_i \lambda_i^* g_i(\mathbf{x}). \tag{3.40}$$

В то же время условие регулярности Слейтера утверждает, что существует такой $\mathbf{x}$, что $g(\mathbf{x}) < \mathbf{0}$ и, следовательно, $\sum_i \lambda_i^* g_i(\mathbf{x}) < 0$. Так как это противоречит уравнению (3.40), то предположения теоремы вместе с условием регулярности Слейтера обеспечивают ее справедливость.

Таким образом, при *выполнении условий теоремы 5.9 задача нелинейного программирования оказывается эквивалентной задаче об отыскании седловой точки функции Лагранжа.*

Применение теоремы Куна — Таккера для задачи вогнутого программирования

Выше нами была рассмотрена НП-задача в виде (3.1), (3.2), когда на переменные $\{x_j\}$ не накладывались условия неотрицательности. Зачастую в задачах исследования операций приходится решать задачи, в которых переменные x_j по физическим условиям должны удовлетворять условию $x_j \geqslant 0$ для всех $j = \overline{1, n}$.

Основные положения теории могут быть легко распространены на этот случай. Действительно, пусть НП-задача записана в виде: найти

$$\min f(\mathbf{x}) \tag{3.41}$$

при условиях

$$g_i(\mathbf{x}) \leqslant 0, \quad i = \overline{1, m}, \tag{3.42}$$

$$x_j \geqslant 0, \quad j = \overline{1, n}. \tag{3.43}$$

Введем обозначения $x_j = -h_j(x_j)$.

Тогда ограничения (3.43) можно записать в общем виде

$$h_j(\mathbf{x}) \leqslant 0, \quad j = \overline{1, n}. \tag{3.44}$$

Задача теперь оказывается заданной в каноническом виде (3.1) — (3.2). Применим к ней теорему Куна — Таккера, для чего составим функцию Лагранжа

$$L(\mathbf{x}, \lambda, \mathbf{u}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} u_j h_j(\mathbf{x}), \tag{3.45}$$

где $u_j \geqslant 0$ — множители, связанные с ограничениями $h_j(\mathbf{x}) \leqslant 0$. Условия теоремы Куна — Таккера для (3.45) выглядят так:

$$\nabla L(\mathbf{x}, \lambda, \mathbf{u}) = \nabla f(\mathbf{x}) + \sum_i \lambda_i \nabla g_i(\mathbf{x}) + \sum_j u_j \nabla h_j(\mathbf{x}) = \mathbf{0} \tag{3.46}$$

или

$$\frac{\partial L(\mathbf{x}, \lambda, \mathbf{u})}{\partial x_j} = \frac{\partial f(\mathbf{x})}{\partial x_j} + \sum_i \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j} - u_j = 0, \quad j = 1, n \tag{3.47}$$

$$u_j x_j = 0, \quad u_j \geqslant 0, \tag{3.48}$$

$$\lambda_i g_i(x) = 0, \quad \lambda_i \geqslant 0. \tag{3.49}$$

Условия (3.47), (3.48) и (3.49) можно записать в следующей эквивалентной форме:

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial x_j} = u_j \geqslant 0, \tag{3.50}$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} x_j = 0, \quad j = \overline{1, n}. \tag{3.51}$$

Нетрудно увидеть, что условия (3.51) представляют собой условия дополняющей нежесткости для ограничений неотрицательности. Таким образом, мы получили необходимые условия для оптимального решения задачи НП вида (3.41) — (3.42), которые могут быть сформулированы в следующей теореме.

Теорема 5.10. *Пусть НП-задача задана в виде (3.41) — (3.43), а функции* f (x) *и* g_1 (x) ,..., g_m (x) *дифференцируемы и выпуклы по* x. *Вектор* $\mathbf{x}^0 \geqslant \mathbf{0}$ *является оптимальным решением задачи тогда и только тогда, когда существует такой вектор* $\lambda^0 \geqslant 0$, *что пара* $(\mathbf{x}^0, \lambda^0)$ *является седловой точкой функции Лагранжа* L (x, λ), *т. е. выполняются следующие условия:*

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_j} \geqslant 0, \quad j = \overline{1, n}, \tag{3.52}$$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_j} x_j^0 = 0, \tag{3.53}$$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial \lambda_i} = g_i(\mathbf{x}^0) \leqslant 0, \tag{3.54}$$

$$\lambda_i^0 g_i(\mathbf{x}^0) = 0, \quad i = \overline{1, m}. \tag{3.55}$$

Задача (3.41) — (3.43) при условии, что f (x) и все g_i (x) — выпуклые функции, является задачей *выпуклого программирования*. Ограничения $g_i(\mathbf{x}) \leqslant 0$, $i = \overline{1, m}$ определяют выпуклое множество, и требуется найти минимум выпуклой функции f (x) на выпуклом множестве решений $R(\mathbf{x}) = \{\mathbf{x} : \mathbf{x} \geqslant 0;\ g_i(\mathbf{x}) \leqslant 0,\ i = \overline{1, m}\}$.

Рассмотрим задачу так называемого *вогнутого* программирования: найти

$$\max_{\mathbf{x} \geqslant 0} f(\mathbf{x}) \tag{3.56}$$

при условиях

$$\left.\begin{aligned} g_1(\mathbf{x}) &\geqslant 0, \\ g_m(\mathbf{x}) &\geqslant 0, \end{aligned}\right\} \tag{3.57}$$

$$\mathbf{x} \geqslant \mathbf{0}, \tag{3.58}$$

где функции f (x) и все g_i (x) вогнуты по x.

Покажем ее эквивалентность задаче выпуклого программирования (3.41) — (3.43). Для этого обозначим $f'(\mathbf{x}) = -f(\mathbf{x})$, $g_i'(\mathbf{x}) = -g_i(\mathbf{x})$, и так как $\max f(\mathbf{x}) = \min -f(\mathbf{x})$, то мы приходим к задаче

$$\min f'(\mathbf{x}) \tag{3.59}$$

при условиях.

$$g_i'(\mathbf{x}) \leqslant 0, \tag{3.60}$$

$$\mathbf{x} \geqslant \mathbf{0}. \tag{3.61}$$

Заметим, что все функции $f'(\mathbf{x})$, $g_i'(\mathbf{x})$ будут выпуклы по x, а потому задача (3.59) — (3.61) — это задача выпуклого программирования. Итак, эквивалентность задач (3.56) — (3.58) и (3.41) — (3.43) установлена.

Нетрудно получить соответствующие условия оптимальности для НП-задачи (3.56) — (3.58), аналогичные условиям (3.52) — (3.55). Они формулируются следующим образом.

Теорема 5.11. *Пусть НП-задача задана в виде (3.56) — (3.58), а функции* f (x), g_i (x), $i = \overline{1, m}$ *дифференцируемы. Для того чтобы*

$\mathbf{x}^0$ *являлся оптимальным решением этой задачи, необходимо, чтобы существовал такой вектор* $\lambda^0 \geqslant 0$, для которого выполняются условия:

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_j} \leqslant 0, \quad j = \overline{1, n}; \tag{3.62}$$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_j} x_j^0 = 0, \quad j = \overline{1, n}; \tag{3.63}$$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial \lambda_i} = g_i(x) \geqslant 0, \quad i = \overline{1, m}; \tag{3.64}$$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial \lambda_i} \lambda_i^0 = 0, \quad i = \overline{1, m}. \tag{3.65}$$

Если функции $f(\mathbf{x})$ и $g_i(\mathbf{x})$ вогнуты, то эти условия (3.62) — (3.65) оказываются и достаточными.

§ 4. ДВОЙСТВЕННОСТЬ В ЗАДАЧАХ ОПТИМИЗАЦИИ

Задача линейного программирования как задача Лагранжа

Использование понятия седловой точки позволяет установить глубокие взаимосвязи между прямой и двойственной задачами в математическом программировании. Прежде чем рассмотреть общий случай нелинейного программирования, исследуем эти взаимосвязи для линейного программирования.

Пусть имеется пара двойственных ЛП-задач в виде:

Задача 1		Задача 2	
$\min \mathbf{b}^T\mathbf{x},$	(4.1)	$\max \mathbf{c}^T\mathbf{y},$	(4.3)
$\mathbf{x} \geqslant \mathbf{0};$		$\mathbf{y} \geqslant \mathbf{0};$	
$\mathbf{A}\mathbf{x} \geqslant \mathbf{c};$	(4.2)	$\mathbf{A}^T\mathbf{y} \leqslant \mathbf{b}.$	(4.4)

Для удобства запишем ограничения задачи 1 в виде

$$g(\mathbf{x}) = \mathbf{c} - \mathbf{A}\mathbf{x} \leqslant \mathbf{0}.$$

Для исследования связи прямой и двойственной задачи введем функции Лагранжа. Образуем функцию Лагранжа для задачи 1:

$$\varphi(\mathbf{x}, \lambda) = \mathbf{b}^T\mathbf{x} + \lambda^T(\mathbf{c} - \mathbf{A}\mathbf{x}), \quad \lambda \geqslant \mathbf{0}. \tag{4.5}$$

Так как ограничения линейны, то выполняются условия регулярности теоремы Куна — Таккера и, следовательно, имеем:

$$\lambda_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = \mathbf{0}, \quad \mathbf{x}_0 \geqslant \mathbf{0}, \quad \lambda_0 \geqslant \mathbf{0}. \tag{4.6}$$

Образуем функцию Лагранжа для задачи 2, предварительно преобразовав ее в эквивалентную задачу минимизации $(\min \mathbf{c}^T\mathbf{y})$, для чего введем множители Лагранжа μ:

$$\psi(\mathbf{y}, \mu) = -\mathbf{c}^T\mathbf{y} + \mu^T(\mathbf{A}^T\mathbf{y} - \mathbf{b}), \quad \mu \geqslant \mathbf{0}.$$

Условия регулярности вновь, как и в первом случае, выполняются, и потому мы имеем

$$\mu_0^T (\mathbf{A}^T \mathbf{y}_0 - \mathbf{b}) = 0, \quad \mu_0 \geqslant 0, \quad \mathbf{y}_0 \geqslant 0. \tag{4.7}$$

Заметим, что если положить $\lambda = \mathbf{y}$, а $\mu = \mathbf{x}$, то $\varphi(\mathbf{x}, \mathbf{y}) = -\psi(\mathbf{y}, \mathbf{x})$. Таким образом, если переменные $\mathbf{y}$ трактовать как множители Лагранжа в задаче 1, а $\mathbf{x}$ — как множители Лагранжа в задаче 2, то целевые функции обеих задач будут равны по величине и противоположны по знаку.

Используя функцию Лагранжа $\varphi(\mathbf{x}, \mathbf{y})$, можно легко показать, что задача 1 может быть записана в виде

$$\min_{\mathbf{x} \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{y}^T \nabla_y \varphi(\mathbf{x}, \mathbf{y})] \tag{4.8}$$

при условиях

$$\nabla_y \varphi(\mathbf{x}, \mathbf{y}) \leqslant 0, \quad \mathbf{y} \geqslant 0.$$

Аналогичным образом задачу 2 можно записать в виде

$$\min_{\mathbf{y} \geqslant 0} [\psi(\mathbf{x}, \mathbf{y}) - \mathbf{x}^T \nabla_x \psi(\mathbf{y}, \mathbf{x})] \tag{4.9}$$

при условии

$$\nabla_x \psi(\mathbf{y}, \mathbf{x}) \leqslant 0. \tag{4.10}$$

Используя соотношения $\varphi(\mathbf{x}, \mathbf{y}) = -\psi(\mathbf{y}, \mathbf{x})$, получим для задачи 2

$$\min_{\mathbf{y} \geqslant 0} [-\varphi(\mathbf{x}, \mathbf{y}) + \mathbf{x}^T \nabla_x \varphi(\mathbf{x}, \mathbf{y})] \sim \max_{\mathbf{y} \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{x}^T \nabla_x \varphi(\mathbf{x}, \mathbf{y})] \tag{4.11}$$

при условии

$$\nabla_x \varphi(\mathbf{x}, \mathbf{y}) \geqslant 0, \quad \mathbf{x} \geqslant 0. \tag{4.12}$$

Используя соотношения (4.8) — (4.12), мы получаем следующее представление пары двойственных ЛП-задач с помощью функций Лагранжа $\varphi(\mathbf{x}, \mathbf{y})$:

Задача 1	Задача 2
$\min\limits_{\mathbf{x} \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{y}^T \nabla_y \varphi(\mathbf{x}, \mathbf{y})]$ (4.13)	$\max\limits_{\mathbf{y} \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{x}^T \nabla_x \varphi(\mathbf{x}, \mathbf{y})]$ (4.15)
при условии	при условии
$\nabla_y \varphi(\mathbf{x}, \mathbf{y}) \leqslant 0$ (4.14) $\mathbf{y} \geqslant 0$	$\nabla_x \varphi(\mathbf{x}, \mathbf{y}) \geqslant 0$ (4.16) $\mathbf{x} \geqslant 0$

При этом для оптимальных решений $(\mathbf{x}_0, \mathbf{y}_0)$ выполняются следующие условия, аналогичные (4.6), (4.7):

$$\mathbf{y}_0^T \nabla_y \varphi(\mathbf{x}_0, \mathbf{y}_0) = 0, \quad \mathbf{x}_0^T \nabla_x \varphi(\mathbf{x}_0, \mathbf{y}_0) = 0. \tag{4.17}$$

Справедлива следующая теорема, устанавливающая связь между оптимальными решениями двойственных задач ЛП и седловой точкой.

Теорема 5.12. *Вектор* $\mathbf{x}_0$ *является оптимальным решением ЛП-задачи 1 тогда и только тогда, когда существует такой вектор* $\mathbf{y}_0$, *что* $(\mathbf{x}_0, \mathbf{y}_0)$ *является седловой точкой функции Лагранжа* $\varphi(\mathbf{x}, \mathbf{y})$, *т. е. для всех* $\mathbf{x} \geqslant 0$ и $\mathbf{y} \geqslant 0$

$$\varphi(\mathbf{x}_0, \mathbf{y}) \leqslant \varphi(\mathbf{x}_0, \mathbf{y}_0) \leqslant \varphi(\mathbf{x}, \mathbf{y}_0). \tag{4.18}$$

Доказательство. Докажем сначала достаточность условий теоремы. Пусть $(\mathbf{x}_0, \mathbf{y}_0)$ — седловая точка, тогда левое неравенство (4.18) выполняется при всех $\mathbf{y} \geqslant 0$ и, в частности, при $\mathbf{y} = 0$. Но поскольку $\mathbf{c} - \mathbf{A}\mathbf{x}_0 \leqslant 0$, то левое неравенство

$$\varphi(\mathbf{x}_0, \mathbf{y} = 0) = \mathbf{b}^T\mathbf{x}_0 \leqslant \varphi(\mathbf{x}_0, \mathbf{y}_0) = \mathbf{b}^T\mathbf{x}_0 + \mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) \tag{4.19}$$

будет возможно только в случае, если

$$\mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = 0. \tag{4.20}$$

Учитывая это, из правого неравенства получим

$$\mathbf{b}^T\mathbf{x}_0 + \mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = \mathbf{b}^T\mathbf{x}_0 \leqslant \mathbf{b}^T\mathbf{x} + \mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}) \leqslant \mathbf{b}^T\mathbf{x}.$$

Таким образом, для всех допустимых $\mathbf{x} \geqslant 0$ $\mathbf{b}^T\mathbf{x}_0 \leqslant \mathbf{b}^T\mathbf{x}$, что устанавливает оптимальность вектора $\mathbf{x}_0$.

Докажем теперь необходимость условий теоремы, т. е. что из оптимальности вектора $\mathbf{x}_0$ следует существование такого вектора $\mathbf{y}_0$, что пара $(\mathbf{x}_0, \mathbf{y}_0)$ будет седловой точкой функции $\varphi(\mathbf{x}, \mathbf{y})$.

Согласно теореме 2.5 главы 2 из оптимальности $\mathbf{x}_0$ следует существование такого $\mathbf{y}_0$, что $\mathbf{b}^T\mathbf{x}_0 = \mathbf{c}^T\mathbf{y}_0$, где $\mathbf{y}_0$ — оптимальное решение двойственной задачи. Тогда согласно теореме 2.6 этой же главы должно выполняться равенство $\mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = 0$. Но так как $\mathbf{x}_0$ — допустимое решение, то $\mathbf{c} - \mathbf{A}\mathbf{x}_0 \leqslant 0$, откуда для любого $\mathbf{y} \geqslant 0$

$$\mathbf{y}^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) \leqslant 0.$$

Таким образом, справедливо неравенство

$$\mathbf{b}^T\mathbf{x}_0 + \mathbf{y}^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = \varphi(\mathbf{x}_0, \mathbf{y}) \leqslant \mathbf{b}^T\mathbf{x}_0 + \mathbf{y}_0^T(\mathbf{c} - \mathbf{A}\mathbf{x}_0) = \varphi(\mathbf{x}_0, \mathbf{y}_0).$$

Левое неравенство для седловой точки установлено. Аналогично может быть получено и правое неравенство, и теорема доказана.

Итак, приходим к выводу, что отыскание решения для пары двойственных задач ЛП эквивалентно нахождению седловой точки соответствующей функции Лагранжа.

Этот вывод справедлив и для некоторых классов задач нелинейного программирования.

Задача нелинейного программирования как задача о седловой точке

Выше нами были получены соотношения (4.13), (4.14) и (4.15), (4.16), связывающие пару двойственных ЛП-задач.

Рассмотрим теперь НП-задачу в виде:
найти

$$\min f(\mathbf{x}) \tag{4.21}$$

при условиях

$$g_i(\mathbf{x}) \leqslant 0 \quad (i = \overline{1, m}). \tag{4.22}$$

Введя функцию Лагранжа $\varphi(\mathbf{x},\ \mathbf{y}) = f(\mathbf{x}) + \sum_i y_i g_i(\mathbf{x})$, можно записать эту задачу в эквивалентном виде:

$$\min_{x} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{y}^T \nabla_y \varphi(\mathbf{x}, \mathbf{y})] \tag{4.23}$$

при условиях

$$\nabla_y \varphi(\mathbf{x}, \mathbf{y}) \leqslant 0, \quad \mathbf{y} \geqslant 0. \tag{4.24}$$

По аналогии с линейным программированием назовем следующую задачу:

$$\max_{\mathbf{y} \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{x}^T \nabla_x \varphi(\mathbf{x}, \mathbf{y})] \tag{4.25}$$

при условии

$$\nabla_x \varphi(\mathbf{x}, \mathbf{y}) \geqslant 0, \tag{4.26}$$

двойственной к задаче (4.23) — (4.24).

Справедлива следующая теорема, аналогичная теореме двойственности линейного программирования, впервые полученная Вулфом [3].

Теорема 5.13. *Пусть функции $f(\mathbf{x})$, $g_i(\mathbf{x})$, $i = \overline{1, m}$ выпуклы и дифференцируемы в R^n и выполняются условия регулярности. Если прямая задача (4.21), (4.22) имеет решение $\mathbf{x}_0$, то в двойственной задаче (4.25) существует такой вектор $\mathbf{y}_0$, который является ее оптимальным решением, и при этом экстремумы пары двойственных задач равны между собой.*

Для иллюстрации этой теоремы рассмотрим задачу квадратичного программирования:
найти

$$\max_{\mathbf{x} \geqslant 0} (\mathbf{b}^T \mathbf{x} + \mathbf{x}^T \mathbf{C} \mathbf{x}) \tag{4.27}$$

при условии

$$\mathbf{A}\mathbf{x} \leqslant \mathbf{A}_0, \tag{4.28}$$

где $\mathbf{C}$ — симметричная, отрицательно определенная матрица.

Преобразуем (4.27), (4.28) в эквивалентную задачу минимизации и построим функцию Лагранжа

$$\varphi(\mathbf{x}, \mathbf{y}) = -(\mathbf{b}^T \mathbf{x} + \mathbf{x}^T \mathbf{C} \mathbf{x}) + \mathbf{y}^T (\mathbf{A}\mathbf{x} - \mathbf{A}_0). \tag{4.29}$$

Исходную задачу можно сформулировать в виде

$$\min_{x \geqslant 0} [\varphi(\mathbf{x}, \mathbf{y}) - \mathbf{y}^T \nabla_y \varphi(\mathbf{x}, \mathbf{y})] \tag{4.30}$$

при условиях

$$\nabla_y \varphi(\mathbf{x}, \mathbf{y}) \leqslant 0, \quad \mathbf{y} \geqslant 0. \tag{4.31}$$

Легко видеть, что задача (4.30), (4.31) эквивалентна задаче (4.27), (4.28).

По аналогии с линейным программированием запишем двойственную задачу к задаче квадратичного программирования:

$$\max_{y\geqslant 0}\,[\varphi(x, y) - x^T\nabla_x\varphi(x, y)] \tag{4.32}$$

при условии

$$\nabla_x\varphi(x, y) \geqslant 0, \quad x \geqslant 0. \tag{4.33}$$

Подставляя выражение для $\varphi(x, y)$ из (4.29) в (4.32) и (4.33), получим следующую запись двойственной задачи:

$$\max_{y\geqslant 0}\,(-A_0^Ty + x^TCx) \to \min_{y\geqslant 0}\,(A_0^Ty - x^TCx) \tag{4.34}$$

при условии

$$-2Cx + A^Ty - b \geqslant 0. \tag{4.35}$$

$$x\geqslant 0$$

Оптимальные решения x_0 и y_0 связаны между собой соотношениями:

$$x_0^T\nabla_x\varphi(x_0, y_0) = 0 \tag{4.36}$$

и

$$y_0^T\nabla_y\varphi(x_0, y_0) = 0. \tag{4.37}$$

Используя выражение (4.29) и подставляя его в (4.36) и (4.37), получим соответственно:

$$x_0^T(b + 2Cx_0 - A^Ty_0) = 0, \tag{4.38}$$

$$y_0^T(Ax_0 - A_0) = 0. \tag{4.39}$$

Покажем, что действительно задачи (4.27) — (4.28) и (4.34) — (4.35) являются двойственными и выполняется утверждение теоремы о том, что

$$\max_{x\geqslant 0}\,(b^Tx + x^TCx) = \min_{y\geqslant 0}\,(A_0^Ty - x^TCx). \tag{4.40}$$

Поскольку x_0 и y_0 связаны между собой соотношениями (4.38) и (4.39), то имеем

$$A_0^Ty_0 = y_0^TAx_0 = x_0^Tb + 2x_0^TCx_0$$

или

$$A_0^Ty_0 - x_0^TCx_0 = b^Tx_0 + x_0^TCx_0.$$

Тем самым устанавливаем справедливость соотношения (4.40) и теоремы 5.12.

§ 5. КВАДРАТИЧНОЕ ПРОГРАММИРОВАНИЕ

К задачам квадратичного программирования относят специальный класс задач НП, для которых целевая функция $f(x)$ — квадратичная, вогнутая, а все ограничения линейны.

Применив к этой задаче теорему Куна—Таккера, получим условия для оптимального решения в виде системы линейных уравнений, решить которые возможно симплекс-методом.

В матричном виде задача квадратичного программирования записывается так.

Найти

$$\max f(\mathbf{x}) = \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} = \sum_{j=1}^{n} b_j x_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_j x_i \tag{5.1}$$

при ограничениях

$$\mathbf{A}\mathbf{x} \leqslant \mathbf{A}_0$$

и

$$\mathbf{x} \geqslant 0, \tag{5.2}$$

где $\mathbf{C} = \| c_{ij} \|_{n,n}$ — симметричная, отрицательно определенная матрица;

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_n \end{bmatrix}; \quad \mathbf{A}^{m\times n} = \| a_{ij} \|; \quad \mathbf{A}_0 = \begin{bmatrix} a_{10} \\ a_{20} \\ a_{m0} \end{bmatrix}.$$

Заметим, что если $\mathbf{C}$ — отрицательно определенная матрица, то квадратичная форма $\mathbf{x}^T\mathbf{C}\mathbf{x}$ вогнута (выпукла вверх). Следовательно, задача (5.1) — (5.2) является задачей вогнутого программирования.

Примеры квадратичных функций:

$$f(x_1, x_2) = 5x_1 + 2x_2 - 4x_1^2 - x_2^2,$$

$$f(x_1, x_2, x_3) = x_1x_2 + 2x_2x_3 - 3x_1^2 - 3x_2^2 - 4x_3^2$$

(это вогнутая функция).

Будем предполагать, что $f(\mathbf{x})$ — строго вогнута.

Применив к задаче (5.1) и (5.2) теорему Куна—Таккера, получим необходимые и достаточные условия оптимальности решения в виде следующей теоремы.

Теорема 5.14. *Вектор* $\mathbf{x}_0 \geqslant 0$ *является оптимальным решением задачи квадратичного программирования тогда и только тогда, когда существуют такие m-мерные векторы* $\boldsymbol{\lambda} > 0$, $\mathbf{w} \geqslant 0$ *и n-мерный* $\mathbf{v} \geqslant 0$, *что выполняются следующие условия:*

1) $\mathbf{b} + \mathbf{C}\mathbf{x} - \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{v} = 0;$

2) $\mathbf{A}_0 - \mathbf{A}\mathbf{x} - \mathbf{w} = 0;$

3) $\mathbf{v}^T\mathbf{x} = 0;$

4) $\mathbf{w}^T\boldsymbol{\lambda} = 0.$

Заметим, что условия 1 и 2 образуют относительно переменных $\mathbf{x}$, $\boldsymbol{\lambda}$, $\mathbf{v}$ и $\mathbf{w}$ систему из $n + m$ уравнений с $2(m + n)$ неизвестными.

Доказательство. Составим функцию Лагранжа для задачи (5.1)

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{A}_0 - \mathbf{A}\mathbf{x}). \tag{5.3}$$

Заметим, что

$$\frac{\partial(\mathbf{b}^T\mathbf{x})}{\partial\mathbf{x}} = \mathbf{b}, \tag{5.4}$$

$$\frac{\partial(\mathbf{x}^T\mathbf{C}\mathbf{x})}{\partial\mathbf{x}} = 2\mathbf{C}\mathbf{x}, \tag{5.5}$$

в чем можно убедиться непосредственной проверкой.

Применив теорему Куна—Таккера к функции (5.3) и используя (5.4) и (5.5), получим:

а) $\frac{\partial \mathbf{L}(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = \mathbf{b} + \mathbf{C}\mathbf{x} - \mathbf{A}^T\lambda \leqslant \mathbf{0}$, причем, если $\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} < 0$,

$$\text{то } x_j^0 = 0; \quad j = 1, \ldots, n;$$

б) $\frac{\partial \mathbf{L}(\mathbf{x}, \lambda)}{\partial \lambda} = \mathbf{A}_0 - \mathbf{A}\mathbf{x} \geqslant \mathbf{0}$, причем, если $\frac{\partial L}{\partial \lambda_i} > 0$,

$$\text{то } \lambda_i^0 = 0.$$

Введем два вспомогательных вектора $\mathbf{v}^T = (v_1, v_2, \ldots, v_n) \geqslant 0$ и $\mathbf{w}^T = (w_1, w_2, \ldots, w_m) \geqslant 0$, причем выберем $v_i > 0$, если $\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} < 0$ и $v_j = 0$, если $\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_j} = 0$. Аналогично выберем $w_i > 0$, если $\frac{\partial L}{\partial \lambda_i} > 0$ и $w_j = 0$, если $\frac{\partial L}{\partial \lambda_i} = 0$ $(i = 1, 2, \ldots, m)$.

Прибавив вектор $\mathbf{v}$ к условию а) и вычтя $\mathbf{w}$ из б), получаем равенства:

$$\begin{aligned} &1)\ \mathbf{b} + \mathbf{C}\mathbf{x} - \mathbf{A}^T\lambda + \mathbf{v} = \mathbf{0}, \\ &2)\ \mathbf{A}_0 - \mathbf{A}\mathbf{x} - \mathbf{w} = \mathbf{0}. \end{aligned} \tag{5.6}$$

Сравнив компоненты векторов $\mathbf{x}$ и $\mathbf{v}$, а также λ и $\mathbf{w}$, получим два условия *дополняющей нежесткости*:

$$3)\ \mathbf{x}^T\mathbf{v} = 0; \quad 4)\ \lambda^T\mathbf{w} = 0.$$

Теорема доказана.

Из условий 3) и 4) следует, что по меньшей мере n переменных из $\mathbf{x}$, $\mathbf{v}$, а также m переменных из набора λ, $\mathbf{w}$ обращаются в нуль.

Как уже отмечалось, система (5.6) состоит из уравнений с $2(m + n)$ переменных.

Таким образом, если существует оптимальное решение задачи (5.1), то оно должно быть одним из базисных решений (5.6). Поскольку для нахождения допустимого базисного решения может быть применен симплекс-метод Данцига, то этот метод (как, впрочем, и другие методы ЛП) вполне пригоден для решения задач квадратичного программирования.

Перепишем (5.6) в виде:

$$\begin{aligned} &\mathbf{C}\mathbf{x} - \mathbf{A}^T\lambda + \mathbf{v} = -\mathbf{b}, \\ &\mathbf{A}\mathbf{x} + \mathbf{w} = \mathbf{A}_0. \end{aligned} \tag{5.7}$$

Для нахождения начального базиса (5.7) можно применить метод искусственных переменных. Введем искусственные переменные $\mathbf{z} = \{z_1, \ldots, z_n\}$ и $\mathbf{y} = \{y_1, \ldots, y_m\}$. В результате приходим к следующей системе:

$$\begin{aligned} &\mathbf{C}\mathbf{x} - \mathbf{A}^T\lambda + \mathbf{v} + \mathbf{z} = -\mathbf{b}, \\ &\mathbf{A}\mathbf{x} + \mathbf{w} + \mathbf{y} = A_0. \end{aligned} \tag{5.8}$$

Выбрав компоненты векторов z и y одинакового знака со знаками соответствующих свободных членов — b, A_0, находим начальное базисное решение.

Составив псевдоцелевую функцию $z = \sum_{i=1}^{m} My_i + \sum_{j=1}^{n} Mz_j \ (M \gg c)$, выводим из базиса искусственные переменные $\{y_i\}$ и $\{z_j\}$ и вводим x, λ, v и w. При этом следует учитывать условия дополняющей нежесткости $x^T v = 0$; $\lambda^T w = 0$.

Если удается вывести все искусственные переменные и при этом удовлетворяются условия 3) и 4) теоремы 5.13, то найденное базисное решение будет оптимальным.

Пример 5.7.

Найти

$$\max f(x_1, x_2) = \max (10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2)$$

при ограничениях

$$\left.\begin{aligned} 8 - x_2 \geqslant 0, \\ 10 - x_1 - x_2 \geqslant 0, \end{aligned}\right\}$$
$$x_1 \geqslant 0, \quad x_2 \geqslant 0.$$

Так как в данном случае квадратичная форма $f(x_1, x_2)$ вогнута и ограничения линейны, то мы имеем задачу квадратичного программирования. Составим функцию Лагранжа $L(x_1, x_2, \lambda_1, \lambda_2) = (10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2) + \lambda_1(8 - x_2) + \lambda_2(10 - x_1 - x_2)$. Применив теорему Куна—Таккера, получим следующие условия для седловой точки:

$$\left.\begin{aligned} &\frac{\partial L}{\partial x_1} = 10 + x_2 - 4x_1 - \lambda_2 \leqslant 0; \\ &\frac{\partial L}{\partial x_2} = 20 + x_1 - 4x_2 - \lambda_1 - \lambda_2 \leqslant 0; \\ &8 - x_2 \geqslant 0; \\ &10 - x_1 - x_2 \geqslant 0; \end{aligned}\right\} \text{I}$$

$$\left.\begin{aligned} &\frac{\partial L}{\partial x_1} x_1 = (10 + x_2 - 4x_1 - \lambda_2) x_1 = 0; \\ &\frac{\partial L}{\partial x_2} x_2 = (20 + x_1 - 4x_2 - \lambda_1 - \lambda_2) x_2 = 0; \\ &\frac{\partial L}{\partial \lambda_1} \lambda_1 = (8 - x_2) \lambda_1 = 0; \\ &\frac{\partial L}{\partial \lambda_2} \lambda_2 = (10 - x_1 - x_2) \lambda_2 = 0. \end{aligned}\right\} \text{II}$$

Требуется найти такое решение системы неравенств I, которое одновременно удовлетворяет и системе II. Для этого можно было бы, например, применить метод перебора и перебрать все возможные варианты вида:

1) $x_1 \neq 0, \quad x_2 \neq 0, \quad \lambda_1 \neq 0, \quad \lambda_2 \neq 0;$

2) $x_1 = 0, \quad x_2 \neq 0, \quad \lambda_1 \neq 0, \quad \lambda_2 \neq 0;$

. . . .

16) $x_1 = 0, \quad x_2 = 0, \quad \lambda_1 = 0, \quad \lambda_2 = 0.$

Однако в данном случае этот подход не рационален, поскольку можно применить стандартный алгоритм линейного программирования.

Перепишем систему I в виде

$$\left.\begin{aligned} 4x_1 - x_2 + \lambda_2 &\geqslant 10, \\ -x_1 + 4x_2 + \lambda_1 + \lambda_2 &\geqslant 20, \\ x_2 &\leqslant 8, \\ x_1 + x_2 &\leqslant 10. \end{aligned}\right\} \text{I, } a$$

Таблица 5.1

				0	0	0	0	0	0	M	M	0	0
		B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
←	M	A_7	10	$\boxed{4}$	−1	0	1	−1		1			
	M	A_8	20	−1	4	1	1		−1		1		
	0	A_9	8	0	1	0	0					1	
	0	A_{10}	10	1	1	0	0						1
			30M	3M	3M	M	2M	−M	−M	0	0	0	0

Таблица 5.2

				0	0	0	0	0	0	M	M	0	0
		B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
		A_1	$2\frac{1}{2}$	1	$-\frac{1}{4}$	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0
←	M	A_8	$22\frac{1}{2}$	0	$\boxed{3\frac{3}{4}}$	1	$\frac{5}{4}$	$-\frac{1}{4}$	−1	$\frac{1}{4}$	1	0	0
		A_9	8	0	1	0	0		0	0		1	0
		A_{10}	$7\frac{1}{2}$	0	$\frac{5}{4}$	0	$-\frac{1}{4}$	$+\frac{1}{4}$	0	$-\frac{1}{4}$	0	0	1
			$\frac{45M}{2}$	0	$\frac{15M}{4}$	M	$\frac{5M}{4}$	$-\frac{M}{4}$	−M	$-\frac{3M}{4}$	0	0	0

Вводим теперь свободные переменные v_1, v_2, w_1, w_2, обращающие неравенства (I, a) в равенства. Эти переменные удовлетворяют дополнительным условиям $v_1 x_1 = 0$; $v_2 \lambda_2 = 0$; $w_1 \lambda_1 = 0$; $w_2 \lambda_2 = 0$.

$$\text{III} \left\{ \begin{array}{rr} 4x_1 - x_2 + \lambda_2 - v_1 = 10 \quad, & (1) \\ -x_1 + 4x_2 + \lambda_1 + \lambda_2 - v_2 = 20\,, & (2) \\ x_2 + w_1 = 8, & (3) \\ x_1 + x_2 + w_2 = 10. & (4) \end{array} \right.$$

Для нахождения допустимого базисного решения системы III применим метод искусственных переменных. Вводим переменные y_1 и y_2 в ограничения 1) и 2) соответственно и образуем псевдоцелевую функцию $z = \min My_1 + My_2$. Составим первую симплекс-таблицу (табл. 5.1).

Выполнив первый шаг симплекс-преобразования, выводим из базиса A_7 *и переменную* y_1. Приходим к табл. 5.2.

Выполнив очередной шаг симплекс-метода, получаем табл. 5.3.

Таблица 5.3

									M	M		
	B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
0	A_1	4	1	0	$\frac{1}{15}$	$\frac{1}{3}$	$-\frac{16}{60}$	$-\frac{1}{15}$	$\frac{16}{60}$	$\frac{1}{15}$	0	0
0	A_2	6	0	1	$\frac{4}{15}$	$\frac{1}{3}$	$-\frac{1}{15}$	$-\frac{4}{15}$	$\frac{1}{3}$	$\frac{4}{15}$	0	0
0	A_9	2	0	0	$-\frac{4}{15}$	$-\frac{1}{3}$	$+\frac{1}{15}$	$+\frac{4}{15}$	$-\frac{1}{3}$	$-\frac{4}{15}$	1	0
0	A_{10}	0	0	0	$-\frac{1}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$	$+\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$	0	1
		0	0	0	0	0	0	0	$-M$	$-M$	0	0

Нами получено следующее допустимое базисное решение:

$$x_1^* = 4; \quad x_2^* = 6; \quad w_1^* = 2; \quad w_2^* = 0; \quad \lambda_1 = \lambda_2 = v_1 = v_2 = 0.$$

Так как $x_1 v_1 = 0$; $x_2 v_2 = 0$; $\lambda_1 w_1 = 0$; $\lambda_2 w_2 = 0$, то это решение является оптимальным.

Итак,

$$x_1^{\mathrm{opt}} = 4; \quad x_2^{\mathrm{opt}} = 6.$$

§ 6. ГЕОМЕТРИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Постановка задачи геометрического программирования и ее основные свойства

Достаточно широкий и интересный с практической точки зрения класс задач НП составляют задачи геометрического программирования.

Примером такой задачи служит, например, следующая.

Пусть требуется переправить через реку 400 куб. м гравия. Допустим, что гравий грузится в открытый ящик длиной t_1, шириной t_2 и высотой t_3. Боковые стороны и дно ящика изготовлены из материала, 1 кв. м которого стоит 10 руб., а передняя и задняя стенка из материала, кв. м которого стоит 20 руб. Каждая перевозка ящика любых размеров с одного берега на другой и обратно стоит 0,1 руб., причем после его использования ящик не будет иметь остаточной стоимости.

Чему равняется минимальная суммарная стоимость транспортировки 400 куб. м гравия?

Очевидно, при линейных размерах ящика t_1, t_2 и t_3 число рейсов, которые нужно выполнить для перевозки 400 куб. м гравия, составляет $\frac{400}{t_1t_2t_3}$, а стоимость перевозок $0{,}1\,\frac{400}{t_1t_2t_3}=\frac{40}{t_1t_2t_3}$.

Общая стоимость материала составляет $40t_2t_3+20t_1t_3+10t_1t_2$. Следовательно, суммарная стоимость перевозок с учетом стоимости материала составит

$$g(\mathrm{t})=\frac{40}{t_1t_2t_3}+40t_2t_3+20t_1t_3+10t_1t_2.$$

Заметим, что функция $g(\mathrm{t})$ состоит из слагаемых $u_i(\mathrm{t})$ вида $u_i(\mathrm{t})=c_it_1^{\alpha_{i1}}t_2^{\alpha_{i2}}\dots t_m^{\alpha_{im}}$, где $c_i>0$; $t_j\geqslant 0$; $j=\overline{1,m}$.
Функции вида $u_i(\mathrm{t})$ носят название позиномов.

Для исследования задачи минимизации позиномов мы используем известное неравенство о среднем арифметическом и геометрическом, которое называют *геометрическим* *.

Для n переменных $u_i>0$ оно выглядит так:

$$\frac{1}{n}\sum_{i=1}^{n}u_i\geqslant u_1^{\frac{1}{n}}u_2^{\frac{1}{n}}\dots u_n^{\frac{1}{n}}. \qquad (6.1)$$

Причем равенство в (6.1) будет в случае, когда $u_1=u_2=\dots=u_i=u$. Более общим случаем является геометрическое неравенство для средневзвешенных арифметического и геометрического:

$$\sum_{i=1}^{n}\delta_iu_i\geqslant\prod_{i=1}^{n}u_i^{\delta_i}, \qquad (6.2)$$

где все веса $\delta_i\geqslant 0$ и выполняется условие нормализации $\sum_{i=1}^{n}\delta_i=1$. В частном случае при $n=2$, $\delta_1=\frac{1}{4}$; $\delta_2=\frac{3}{4}$ получим $\frac{1}{4}u_1+\frac{3}{4}u_2\geqslant u_1^{\frac{1}{4}}u_2^{\frac{3}{4}}$. Доказательство неравенства (6.2) дано в приложении 5.

Это неравенство может быть использовано для нахождения минимума позинома $g(t)$.

---

* Отсюда и проистекает название соответствующего класса методов.

Пусть, например, $q(t_1, t_2) = 4t_1 + \frac{t_1}{t_2^2} + \frac{4t_2}{t_1}$.

Чтобы найти оценку снизу для $q(t)$ используем геометрическое неравенство $\left(\text{при } \delta_1 = \frac{1}{4}, \delta_2 = \frac{1}{4}, \delta_3 = \frac{1}{2}\right)$:

$$\frac{1}{4} u_1 + \frac{1}{4} u_2 + \frac{1}{2} u_3 \geqslant u_1^{\frac{1}{4}} \cdot u_2^{\frac{1}{4}} \; u_3^{\frac{1}{2}}.$$

Тогда

$$g(\mathrm{t}) \geqslant (16t_1)^{\frac{1}{4}} \cdot \left(\frac{4t_1}{t_2^2}\right)^{\frac{1}{4}} \cdot \left(\frac{8t_2}{t_1}\right)^{\frac{1}{2}} = 8.$$

Таким образом, 8 является оценкой снизу для $q(\mathrm{t})$ при $t_1 > 0$ и $t_2 > 0$. Более того, 8 является точной нижней гранью для $q(\mathrm{t})$, так как $q\left(t_1 = \frac{1}{2};\ t_2 = \frac{1}{4}\right) = 8$.

Двойственная функция. Рассмотрим теперь общий случай геометрического неравенства (6.2).

$$\delta_1 U_1 + \delta_2 U_2 + \cdots + \delta_n U_n \geqslant U_1^{\delta_1} U_2^{\delta_2} \ldots U_n^{\delta_n},$$

где все $U_i > 0$, а веса $\delta_i \geqslant 0$ удовлетворяют условию нормализации $\delta_1 + \delta_2 + \ldots + \delta_n = 1$.

Для удобства произведем замену переменных в (6.2), полагая $u_1 = \delta_1 U_1$, $u_2 = \delta_2 U_2$, ..., и т. д. Тогда геометрическое неравенство (6.2) примет вид:

$$u_1 + u_2 + \cdots + u_n \geqslant \left(\frac{u_1}{\delta_1}\right)^{\delta_1} \cdot \left(\frac{u_2}{\delta_2}\right)^{\delta_2} \cdot \ldots \cdot \left(\frac{u_n}{\delta_n}\right)^{\delta_n}. \tag{6.3}$$

Левая часть (6.3) представляет собой функцию-позином $g(\mathrm{t})$. Назовем ее *прямой функцией* задачи геометрического программирования (ГП).

Правая часть (6.3) называется преддвойственной функцией. Обозначим ее через V. Тогда неравенство (6.3) можно будет записать

$$g \geqslant V. \tag{6.4}$$

Если исходная функция $g = u_1 + u_2 + \cdots + u_n$ — позином, причем $u_i = c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}$, то подставляя выражения для u_i в правую часть (6.4), получим выражение для V:

$$V(\delta, \mathrm{t}) = \left(\frac{c_1}{\delta_1}\right)^{\delta_1} \left(\frac{c_2}{\delta_2}\right)^{\delta_2} \cdot \ldots \cdot \left(\frac{c_n}{\delta_n}\right)^{\delta_n} t_1^{D_1} t_2^{D_2} \ldots t_m^{D_m}, \tag{6.5}$$

где

$$D_j = \sum_{i=1}^{n} \delta_i a_{ij}; \quad j = \overline{1, m}. \tag{6.6}$$

Допустим, что можно выбрать веса δ_i так, чтобы все показатели D_j обратились в нули. Тогда преддвойственная функция $V(\delta, \mathrm{t})$ не будет зависеть от переменных t_j и назовем ее *двойственной функцией*

$$v(\delta) = \left(\frac{c_1}{\delta_1}\right)^{\delta_1} \left(\frac{c_2}{\delta_2}\right)^{\delta_2} \cdot \ldots \cdot \left(\frac{c_n}{\delta_n}\right)^{\delta_n}. \tag{6.7}$$

Из неравенства (6.3) следует, что g (t) имеет положительную точную нижнюю грань M. Тогда мы можем записать, что

$$g(\mathrm{t}) \geqslant M \geqslant v(\delta). \tag{6.8}$$

Из (6.8) следует, что M является оценкой сверху для двойственной функции для любого выбора весов δ_i, при котором все показатели D_j обращаются в 0.

Используем полученные выше результаты для решения рассмотренной выше задачи.

Для функции

$$g(\mathrm{t}) = \frac{40}{t_1 t_2 t_3} + 40 t_2 t_3 + 20 t_1 t_3 + 10 t_1 t_2$$

запишем двойственную функцию:

$$v(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1} \left(\frac{40}{\delta_2}\right)^{\delta_2} \left(\frac{20}{\delta_3}\right)^{\delta_3} \left(\frac{10}{\delta_4}\right)^{\delta_4}, \tag{6.9}$$

где веса δ_i, $i = \overline{1, 4}$ удовлетворяют условиям:

$$\begin{aligned} D_1 &= -\delta_1 + \delta_3 + \delta_4 = 0; \\ D_2 &= -\delta_1 + \delta_2 + \delta_4 = 0; \\ D_3 &= -\delta_1 + \delta_2 + \delta_3 = 0. \end{aligned} \tag{6.10}$$

Условия (6.10) называют *условиями ортогональности*. Кроме того, веса δ_i должны удовлетворять *условию нормализации*:

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1. \tag{6.11}$$

Решая систему уравнений (6.10), находим единственное решение:

$$\delta_1^* = \frac{2}{5}; \quad \delta_2^* = \frac{1}{5}; \quad \delta_3^* = \frac{1}{5}; \quad \delta_4^* = \frac{1}{5}.$$

Подставляя эти значения в (6.9), получим

$$v(\delta^*) = \left(\frac{40}{{}^2/_5}\right)^{2/5} \left(\frac{40}{{}^1/_5}\right)^{1/5} \left(\frac{20}{{}^1/_5}\right)^{1/5} \left(\frac{10}{{}^1/_5}\right)^{1/5} = 100.$$

Итак,

$$\min g(\mathrm{t}) = v(\delta^*) = 100.$$

Таким образом, минимальная суммарная стоимость равна 100 руб.

Максимум двойственной функции

Докажем теперь, что существуют положительные веса δ_i, удовлетворяющие условиям ортогональности, и *максимум двойственной функции v (δ) равен минимуму прямой функции g (t).*

Предполагаем, что позином g (t) имеет минимальное значение в некоторой точке $(t_1^*, t_2^*, \ldots, t_m^*)$, все координаты которой положительны.

В минимизирующей точке t^* производные g (t) по каждой переменной обращаются в 0 и мы получаем m уравнений вида:

$$0 = \frac{\partial g(\mathrm{t}^*)}{\partial t_j} t_j^* = \sum_{i=1}^{n} t_j^* \frac{\partial u_i(\mathrm{t}^*)}{\partial t_j} = \sum_{i=1}^{n} u_i(\mathrm{t}^*)\, a_{ij}. \tag{6.12}$$

Разделив эти уравнения (6.12) на g (t) $\neq 0$ и полагая

$$\frac{u_i(\mathbf{t}^*)}{g(\mathbf{t}^*)} = \delta_i^*, \tag{6.13}$$

получим, что

$$\sum_{i=1}^{n} \delta_i a_{ij} = 0, \quad j = \overline{1, m}. \tag{6.14}$$

Таким образом, вектор δ^*, задаваемый (6.13), удовлетворяет условиям ортогональности, а кроме того, очевидно, он удовлетворяет и условиям нормализации.

Поэтому

$$g(\mathbf{t}^*) = (g^*)^{\delta_1^*}(g^*)^{\delta_2^*} \ldots (g^*)^{\delta_n^*} = \prod_{i=1}^{n} (g^*)^{\delta_i^*}.$$

Но с другой стороны, согласно (6.13) $g^* = \dfrac{u_i(\mathbf{t}^*)}{\delta_i^*}$,

и поэтому

$$\prod_{i=1}^{n} (g^*)^{\delta_i^*} = \prod_{i=1}^{n} \left(\frac{u_i(t^*)}{\delta_i^*}\right)^{\delta_i^*} = v(\delta^*). \tag{6.15}$$

Тогда из (6.15) следует, что

$$g(\mathbf{t}^*) = v(\delta^*). \tag{6.16}$$

Это соотношение вместе с (6.8) доказывает, что минимум прямой функции g равен максимуму двойственной функции v.

Определение минимизирующей точки

Как следует из (6.13) и (6.16), метод геометрического программирования позволяет найти минимальное значение позинома g (t) без предварительного определения искомой точки $\mathbf{t}^*$. В этом методе первоначально находится точка $(\delta_1^*, \delta_2^*, \ldots, \delta_n^*)$, которая максимизирует двойственную функцию v при условиях ортогональности и нормализации.

Остается теперь задача определения точки минимума $\mathbf{t}^* = (t_1^*, t_2^*, \ldots, t_m^*)$. Покажем, как найти искомую точку $\mathbf{t}^*$ при известных δ_i^*, $v(\delta^*)$.

Для этого воспользуемся соотношениями (6.13). Так как

$$g(\mathbf{t}^*) = v(\delta^*), \text{ то}$$

$$u_i(\mathbf{t}^*) = v(\delta^*)\,\delta_i^*, \quad i = \overline{1, n}. \tag{6.17}$$

Решая систему (6.17), найдем искомые значения для неизвестных $\mathbf{t}^* = (t_1^*, t_2^*, \ldots, t_m^*)$.

В качестве примера рассмотрим задачу 1, сформулированную выше, и покажем, как найти переменные t_1^*, t_2^*, t_3^*.

Мы нашли, что

$$\delta_1^* = \frac{2}{5}; \quad \delta_2^* = \frac{1}{5}; \quad \delta_3^* = \frac{1}{5}; \quad \delta_4^* = \frac{1}{5}; \quad M = v(\delta^*) = 100,$$

а из (6.13) следует, что t^* удовлетворяет системе

$$\left(\frac{2}{5}\right)\cdot 100 = u_1 = 40t_1^{-1}t_2^{-1}t_3^{-1}; \tag{6.18}$$

$$\left(\frac{1}{5}\right)\cdot 100 = u_2 = 40t_2t_3;$$

$$\left(\frac{1}{5}\right)\cdot 100 = u_3 = 20t_1t_3;$$

$$\left(\frac{1}{5}\right)\cdot 100 = u_4 = 10t_1t_2.$$

Решая уравнения системы (6.18), находим искомые значения переменных $t_1^* = 2$; $t_2^* = 1$; $t_3^* = \frac{1}{2}$.

Непосредственной проверкой убеждаемся, что при оптимальном значении вектора $\mathbf{t}^*$ выполняется равенство $g(\mathbf{t}^*) = 100 = v(\delta^*)$.

Прямая и двойственная задача геометрического программирования (общая формулировка) и их свойства

Выше были рассмотрены частные постановки задачи геометрического программирования (ГП) и исследованы некоторые свойства и взаимосвязь прямой и двойственной задач ГП. В настоящем разделе рассматриваются постановки и элементы теории геометрического программирования, а также соответствие между прямой и двойственной задачами геометрического программирования. В основе теории геометрического программирования лежит обобщенное геометрическое неравенство, а также теорема Куна—Таккера для задач нелинейного программирования.

В наиболее общей постановке прямая задача ГП формулируется следующим образом [19].

Прямая задача А. Найти минимальное значение функции $g_0(\mathbf{t})$ при ограничениях:

$$t_1 > 0,\ t_2 > 0,\ \ldots,\ t_m > 0 \tag{6.19}$$

и

$$g_1(\mathbf{t}) \leqslant 1,\ g_2(\mathbf{t}) \leqslant 1,\ \ldots,\ g_p(\mathbf{t}) \leqslant 1, \tag{6.20}$$

где

$$g_k(\mathbf{t}) = \sum_{i \in I[k]} c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}},\quad k = 0, 1, \ldots, p, \tag{6.21}$$

$$I[k] = \{m_k;\ m_{k+1},\ \ldots,\ n_k\};\quad k = 0, 1, \ldots, p, \tag{6.22}$$

$$m_0 = 1;\ m_1 = n_0 + 1;\ m_k = n_{k-1} + 1;\ \ldots,\ n_p = n. \tag{6.23}$$

Все $c_i > 0$; $i = \overline{1, n}$, а показатели степени a_{ij} — вещественные числа. Следовательно, все функции $g_k(\mathbf{t})$ — позиномы. Минимизируемая функция $g_0(\mathbf{t})$ называется прямой функцией, ограничения (6.19) — условиями неотрицательности, а ограничения (6.20) — вынужденными ограничениями. Матрица $\mathbf{A} = \| a_{ij} \|;\ \begin{matrix} i = n \\ j = m \end{matrix}$ — называется матрицей экспонент.

Двойственная задача, соответствующая прямой задаче А, формулируется следующим образом.

Двойственная задача В. Найти максимальное значение функции — произведения

$$v(\delta) = \left[\prod_{i=1}^{n}\left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right]\prod_{k=1}^{p}\lambda_k(\delta)^{\lambda_k(\delta)}, \tag{6.24}$$

где

$$\lambda_k(\delta) = \sum_{i \in I[k]} \delta_i, \quad k = \overline{1, p}, \tag{6.25}$$

а множества индексов $I[k]$ определяются согласно (6.22) и (6.23). Множители c_i, как и выше, предполагаются положительными, а на переменные $\delta = (\delta\delta_2, \dots, \delta_n)$ налагаются следующие ограничения:

$$\delta_1 \geqslant 0,\ \delta_2 \geqslant 0,\ \dots,\ \delta_n \geqslant 0, \tag{6.26}$$

$$\sum_{i \in I[0]} \delta_i = 1 \tag{6.27}$$

и

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0, \quad j = 1, 2, \dots, m. \tag{6.28}$$

Функция $v(\delta)$ называется *двойственной функцией*, а переменные $\delta_1, \delta_2, \dots, \delta_n$ — *двойственными переменными*. Условие (6.27) — условие нормализации, а соотношения (6.28) — условия ортогональности.

Путем сопоставления форм записи прямой задачи (6.19) — (6.23) и двойственной (6.24)—(6.28) можно установить следующие соответствия между ними.

1. Каждому члену позиномов прямой задачи вида $u_i = c_i t_1^{a_{i1}} \dots t_m^{a_{im}}$ соответствует одна двойственная переменная δ_i и наоборот.

2. Каждый множитель $\lambda_k(\delta)^{\lambda_k(\delta)}$ функции $v(\delta)$ определяется вынужденным ограничением $g_k(\mathbf{t}) \leqslant 1$.

Заметим, что целевая функция $g_0(t)$ не влечет появление такого множителя, так как по условию нормализации $\lambda_0(\delta) = 1$.

3. Условие нормализации (6.27) — это единственное условие, по которому различаются целевая функция $g_0(\mathbf{t})$ и позиномы — ограничения $g_k(\mathbf{t})$.

Связь между оптимальными решениями прямой и двойственной задач геометрического программирования устанавливается в следующей основной теореме [19].

Теорема 5.15. *Пусть прямая задача А совместна и существует решение* $\mathbf{t}'$ *такое, что* $g_k(\mathbf{t}') < 1$, $k = \overline{1, p}$, а *также оптимальное решение. Тогда:*

1. Соответствующая двойственная задача совместна и существует точка, удовлетворяющая двойственным ограничениям, в которой достигается условный максимум двойственной функции $v(\delta)$.

2. Максимальное значение целевой функции двойственной задачи равно минимальному значению целевой функции прямой задачи

$$\min g_0(\mathbf{t}) = \max v(\delta).$$

3. Если $\mathbf{t}^*$ — минимизирующая точка прямой задачи А, то существуют неотрицательные множители Лагранжа $\overset{*}{\mu}_k$, $k = \overline{1, p}$, такие, что функция Лагранжа

$$L(\mathbf{t}, \boldsymbol{\mu}) = g_0(\mathbf{t}) + \sum_{k=1}^{p} \mu_k [g_k(\mathbf{t}) - 1] \qquad (6.29)$$

обладает тем свойством, что

$$L(\mathbf{t}^*, \boldsymbol{\mu}) \leqslant L(\mathbf{t}^*, \boldsymbol{\mu}^*) = g_0(\mathbf{t}^*) \leqslant L(\mathbf{t}, \boldsymbol{\mu}^*) \qquad (6.30)$$

для произвольных $t_j > 0$ и произвольных $\mu_k \geqslant 0$.
Кроме того, существует максимизирующий вектор $\boldsymbol{\delta}^*$ двойственной задачи с компонентами, определяемыми из условий

$$\delta_i^* = \begin{cases} \dfrac{c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}}{g_0(\mathbf{t})}; & i \in I[0], \\ \dfrac{\mu_k c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}}{g_0(\mathbf{t})}, & i \in I[k], \quad k = 1, p, \end{cases} \qquad (6.31)$$

где

$$\mathbf{t} = \mathbf{t}^* \text{ и } \boldsymbol{\mu} = \boldsymbol{\mu}^*.$$

Далее

$$\lambda_k(\boldsymbol{\delta}^*) = \frac{\overset{*}{\mu}_k}{g_0(\mathbf{t}^*)}. \qquad (6.32)$$

4. Если $\boldsymbol{\delta}^*$ — максимизирующая точка двойственной задачи В, то любая минимизирующая точка $\mathbf{t}^*$ прямой задачи А удовлетворяет системе уравнений

$$c_i t_1^{a_{i1}} \ldots t_2^{a_{im}} = \begin{cases} \delta_i^* v(\boldsymbol{\delta}^*); & i \in I[0], \\ \dfrac{\delta_i^*}{\lambda_k(\boldsymbol{\delta}^*)}; & i \in I[k], \end{cases} \qquad (6.33)$$

где k пробегает все положительные целочисленные значения, для которых $\lambda_k(\delta^*) > 0$.

Доказательство теоремы 5.15 приводится в приложении 5.

Пользуясь результатами этой теоремы, зная решение прямой задачи, можно на основании соотношений (6.31) определить максимизирующий вектор задачи δ^*. Наоборот, если известно решение двойственной задачи δ^*, то используя формулы (6.33), можно определить оптимальное решение прямой задачи $\mathbf{t}^*$. Заметим, что система (6.33) путем логарифмирования обеих частей каждого уравнения оказывается линейной относительно $\ln t_1$, $\ln t_2$, ..., $\ln t_m$. Наконец, из (6.32) следует, что величины $\lambda_k(\delta^*)$ с точностью до постоянного множителя оказываются множителями Лагранжа для прямой задачи.

Одним из условий теоремы 5.15 является предположение о существовании точки, удовлетворяющей ограничениям прямой задачи, где достигается минимум $g_0(\mathbf{t})$.

Следующая теорема двойственности определяет достаточные условия, при которых это предположение справедливо.

Теорема 5.16. *Если прямая задача* А *совместна и существует точка* δ^* *с положительными компонентами, удовлетворяющая ограничениям двойственной задачи* В, *то существует точка* $\mathbf{t}^*$, *удовлетворяющая ограничениям прямой задачи, в которой функция* $g_0(\mathbf{t})$ *достигает своего минимума.*

Исследуем некоторые свойства задачи геометрического программирования. С этой целью рассмотрим позином

$$g(\mathbf{t}) = \sum_i c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}} \tag{6.34}$$

и произведем в нем замену переменных, полагая

$$t_j = e^{z_j}, \quad j = 1, 2, \ldots, m. \tag{6.35}$$

Преобразованная функция, имеющая вид

$$g(\mathbf{z}) = \sum_i c_i e^{\sum\limits_{j=1}^{n} a_{ij} z_j}, \tag{6.36}$$

называется *положительной показательной* функцией. Отметим, что переменные z_j преобразованной прямой задачи меняются на множестве всех вещественных чисел, тогда как переменные исходные прямой задачи t_j должны удовлетворять условию неотрицательности.

Итак, мы преобразовали прямую задачу А к следующему виду.

Прямая задача A_z. Найти минимум положительной показательной функции $g_0(\mathbf{z})$ при ограничениях

$$g_1(\mathbf{z}) \leqslant 1,\ g_2(\mathbf{z}) \leqslant 1 \ \ldots \ g_p(\mathbf{z}) \leqslant 1, \tag{6.37}$$

где

$$g_k(\mathbf{z}) = \sum_{i \in I[k]} c_i e^{\sum\limits_j a_{ij} z_j}, \quad k = 0, 1, \ldots, p, \tag{6.38}$$

$$I[k] = \{m_k, m_{k+1}, \ldots, n_k\} \text{ и } m_0 = 1,\ m_1 = n_0 + 1,\ n_p = n.$$

Очевидно, все функции $g_k(\mathbf{z})$ — положительные показательные функции.

Анализ преобразованной задачи A_z позволяет выявить многие важные свойства прямой задачи А. Например, из рассмотрения задачи A_z сразу следует, что любая прямая задача может быть сформулирована так, что ее матрица экспонент $\mathbf{A} = \| a_{ij} \|$ будет иметь ранг m. Действительно, задача A_z может быть представлена в бескоординатной форме:

найти min функции $\sum\limits_{i \in I[0]} c_i e^{x_i}$ при ограничениях

$$\sum_{i \in I[k]} c_i e^{x_i} \leqslant 1, \quad k = \overline{1, p},$$

где вектор $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ принадлежит пространству, натянутому на столбцы матрицы $\mathbf{A} = \| a_{ij} \|$, $x_i = \sum\limits_{j=1}^{m} a_{ij} z_j$. Если ранг $\mathbf{A}$ меньше

m, то векторы-столбцы матрицы А линейно зависимы, и из них можно выделить базисные векторы и небазисные, линейно зависящие от базисных.

Такое выделение будет, естественно, не единственным. Однако при данном выделенном базисе переменным, не входящим в базис (небазисным), можно приписать произвольные значения, и потому их в сущности можно не считать переменными, так как это не оказывает влияния на минимальное значение задачи A_z.

Таким образом, без потери общности можно считать, что ранг матрицы А всегда равен m. Если такая матрица квадратная (т. е. $n = m$), то ее векторы-столбцы образуют базис для пространства E_n. В таком случае всегда существует последовательность векторов $\{z^q\}_{q=1}^{\infty}$, для которой

$$\sum_j a_{ij} z_j^{(q)} = -q, \quad i = 1, 2, \ldots, n; \quad q = 1, 2, \ldots. \tag{6.39}$$

Тогда из соотношения (6.39) следует, что

$$\lim_{q \to \infty} g_k(\mathbf{z}^{(q)}) = 0, \quad k = 0, 1, \ldots, p. \tag{6.40}$$

Но это означает, что задача A_z совместна и $g_0(\mathbf{z})$ имеет минимум, равный 0. Кроме того, ясно, что задачи этого типа вообще не имеют условного минимума. Соответствующая двойственная задача в этом случае всегда несовместна, поскольку нулевой вектор будет единственным решением, удовлетворяющим условиям ортогональности, но он не удовлетворяет условию нормализации.

Итак, мы приходим к выводу, что любая нетривиальная прямая задача может быть сформулирована так, что ее матрица-экспонент будет иметь ранг m, равный числу переменных прямой задачи, где m — строго меньше общего числа членов n.

Целое число $n - m - 1$ называется *степенью трудности* такой задачи.

Степень трудности совпадает с числом независимых переменных, по которым максимизируется двойственная целевая функция. Так, решение задачи с нулевой степенью трудности легко получается из решения двойственной задачи, поскольку ограничения для нее будут иметь единственное решение. Более того, нахождение этого решения δ^* очень несложно, так как все двойственные ограничения линейны. Вследствие того, что вектор δ^* является единственным решением двойственных ограничений, то он является также и максимизирующим вектором для двойственной задачи. Если же степень трудности больше нуля, то для нахождения δ^* требуются дополнительные усилия. Однако, как мы покажем, даже не определяя δ^*, из двойственной задачи можно извлечь много полезной информации.

Двойственная задача B имеет линейные ограничения двух видов: в форме неравенств $\delta_1 \geqslant 0, \ldots, \delta_i \geqslant 0, \ldots, \delta_n \geqslant 0$ и в форме равенств. Ограничения-равенства выражают условия ортогональности и нормализации. Все решения условий ортогональности образуют подпространство пространства E_n, которое называется *двойственным* *пространством* и является ортогональным дополнением прямого пространства, представляющего собой пространство векторов-столбцов матрицы A. Условие нормализации определяет гиперплоскость в E_n, которая называется гиперплоскостью нормализации, условия неотрицательности определяют первый ортант пространства E_n; пересечение двойственного пространства с гиперплоскостью нормализации и первым ортантом и определяет допустимое множество решений, называемое двойственной областью.

Для задачи геометрического программирования с нулевой степенью трудности двойственная область содержит не более одной точки. Для задачи с положительной степенью трудности d можно построить базисные векторы $\mathbf{b}^{(j)}$, $j = 0, 1, \ldots, d$ так, что общее решение двойственных ограничений будет иметь вид

$$\delta = \mathbf{b}^{(0)} + \sum_{j=1}^{d} r_j \mathbf{b}^{(j)}, \qquad (6.41)$$

где r_j — произвольные вещественные числа, удовлетворяющие условиям неотрицательности вида

$$\delta_i = b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)} \geqslant 0. \qquad (6.42)$$

Вектор $\mathbf{b}^{(0)}$, удовлетворяющий условиям нормализации

$$\sum_{i=1}^{n} b_i^{(0)} = 1 \qquad (6.43)$$

и условиям ортогональности

$$\sum_{i=0}^{n} a_{ij} b_i^{(0)} = 0, \quad j = \overline{1, m}, \qquad (6.44)$$

называется *вектором нормализации*.

Векторы $\mathbf{b}^{(j)}$, $j = 1, \ldots, d$ образуют базис пространства решений линейной однородной системы уравнений:

$$\sum_{i \in I[0]} y_i = 0; \qquad (6.45)$$

$$\sum_{i=1}^{n} a_{ij} y_i = 0, \quad j = 1, 2, \ldots, m.$$

Их называют *векторами невязки*, а переменные r_j, связанные с векторами $\mathbf{b}^{(j)}$,— базисными.

Выразив двойственную функцию v через базисные переменные, получим

$$v = \left\{\prod_{i=1}^{n} c_i^{\left[b_i^{(0)} + \sum\limits_{j=1}^{d} r_j b_i^{(j)}\right]}\right\} \left(\prod_{i=1}^{n} \delta_i^{-\delta_i}\right) \prod_{k=1}^{p} \lambda_k (\delta)^{\lambda_k(\delta)}. \qquad (6.46)$$

Пусть

$$K_j = \prod_{i=1}^{n} c_i^{b_i^{(j)}}, \quad j = 0, 1, \ldots, d. \qquad (6.47)$$

Тогда

$$v = K_0 \Bigg(\prod_{j=1}^{d} K_j^{r_j} \Bigg(\Bigg(\prod_{i=1}^{n} \delta_i^{-\delta_i} \Bigg) \prod_{k=1}^{p} \lambda_k (\delta)^{\lambda_k(\delta)}. \tag{6.48}$$

Отметим, что постоянная K_0 имеет размерность двойственной функции, а постоянные K_j, $j = \overline{1, d}$ — безразмерные величины. Эти постоянные называют базисными, так как они зависят от базиса $\{\mathbf{b}_j\}$, $j = 0, 1, \ldots, d$.

Таким образом, мы привели двойственную задачу B к следующему виду.

Преобразованная двойственная задача B_r.

Найти максимум функции

$$v(\mathbf{r}) = K_0 \left(\prod_{j=1}^{d} K_j^{r_j} \right) \left(\prod_{i=1}^{n} \delta_i (\mathbf{r})^{-\delta_i(\mathbf{r})} \right) \prod_{k=1}^{p} \lambda_k (\mathbf{r})^{\lambda_k(\mathbf{r})}, \tag{6.49}$$

где

$$\delta_i (\mathbf{r}) = b_i^{(0)} + \sum_{1}^{d} r_j b_i^{(j)}, \tag{6.50}$$

и

$$\lambda_k (\mathbf{r}) = \lambda_k (0) + \sum_{j=1}^{d} r_j \lambda_k^{(j)}; \quad k = \overline{1, p}, \tag{6.51}$$

$$K_j = \prod_{i=1}^{n} c_i^{b_i^{(j)}}; \quad j = 0, 1, 2, \ldots, d, \tag{6.52}$$

и

$$\lambda_k^{(j)} = \sum_{i \in I[k]} b_i^{(j)}, \quad \lambda_k^{(0)} = \sum_{i \in I[k]} b_i^{(0)}.$$

На вектор $\mathbf{r}$ наложены условия неотрицательности вида

$$\delta_i (\mathbf{r}) = b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)} \geqslant 0; \quad i = 1, 2, \ldots, n. \tag{6.53}$$

Заметим, что единственными ограничениями преобразованной двойственной задачи B_r являются ограничения, вытекающие из условий неотрицательности (6.53).

Все векторы, удовлетворяющие ограничениям преобразованной двойственной задачи B_r, образуют выпуклое множество, так как они являются линейными. Если бы функция $v(\mathbf{r})$ была вогнутой, то для отыскания максимума $v(\mathbf{r})$ можно было бы применить методы вогнутого программирования. Однако это условие для функции $v(\mathbf{r})$ в общем случае не справедливо. Но в то же время функция $\ln v(\mathbf{r})$ всегда вогнута, даже если $v(\mathbf{r})$ не обладает этим свойством. (Доказательство этого факта приводится в [19]). А так как логарифмическая функция является монотонно возрастающей, то $v(\mathbf{r})$ и $\ln v(\mathbf{r})$ имеют одно и то же множество максимизирующих точек. Таким образом, решение преобразованной двойственной задачи B_r сводится к оптимизации вогнутой функции, полученной заменой $v(\mathbf{r})$ на $\ln v(\mathbf{r})$ на выпуклом множестве допустимых решений.

Воспользуемся этими свойствами для нахождения условий оптимальности для искомого решения.

Допустим, что $\mathbf{r}$ — точка, в которой $\delta_i(\mathbf{r}) > 0$, $i = 1, 2, \ldots, n$. Логарифмируя $v(\mathbf{r})$ и дифференцируя по r_j в точке $\mathbf{r}$, получаем

$$\frac{\partial v/\partial r_j}{v(\mathbf{r})} = \ln K_j - \sum_{i=1}^{n} b_i^{(j)} \ln \delta_i(\mathbf{r}) + \sum_{k=1}^{p} \lambda_k^{(j)} \ln \lambda_k(\mathbf{r}). \tag{6.54}$$

Значит, точка $\mathbf{r}$ является стационарной точкой для $\ln v(\mathbf{r})$ тогда и только тогда, когда

$$\ln K_j = \sum_{i=1}^{n} b_i^{(j)} \ln \delta_i(\mathbf{r}) - \sum_{k=1}^{p} \lambda_k^{(j)} \ln \lambda_k(\mathbf{r}), \quad j = 1, 2, \ldots, d,$$

что эквивалентно

$$K_j = \prod_{i=1}^{n} \delta_i(\mathbf{r})^{b_{i(j}} \prod_{k=1}^{p} \lambda_k(\mathbf{r})^{-\lambda_{k(j)}}. \tag{6.55}$$

Если $\mathbf{r}$ удовлетворяет этим уравнениям, то после несложных преобразований выражение для $v(\mathbf{r})$ может быть приведено к виду

$$v(\mathbf{r}) = K_0 \prod_{i=1}^{n} \delta_i(\mathbf{r})^{-b_i^{(0)}} \prod_{k=1}^{p} \lambda_k(\mathbf{r})^{\lambda_{k^{(0)}}}. \tag{6.56}$$

Использование соотношений (6.55) и (6.56), а также свойства, что для вогнутой функции всякая стационарная точка является максимизирующей, доказывают следующую теорему [19].

Теорема 5.17. *Если δ удовлетворяет ограничениям двойственной задачи B и все компоненты δ положительны, то δ является максимизирующей точкой двойственной задачи тогда и только тогда, когда*

$$K_j = \left(\prod_{i=1}^{n} \delta_i^{b_{i^{(j)}}}(\mathbf{r})\right) \prod_{k=1}^{p} \lambda_k(\delta(\mathbf{r}))^{-\lambda_{k^{(j)}}}, \quad j = 1, 2, \ldots, d, \tag{6.57}$$

где

$$K_j = \prod_{i=1}^{n} c_i^{b_{i^{(j)}}}, \quad j = 1, 2, \ldots, d. \tag{6.58}$$

При этом, если δ удовлетворяет соотношениям (6.57), то

$$v(\delta) = K_0 \left(\prod_{i=1}^{n} \delta_i^{-b_{i^{(0)}}}(\mathbf{r})\right) \prod_{k=1}^{p} \lambda_k(\delta)^{\lambda_{k^{(0)}}}. \tag{6.59}$$

Уравнения (6.57) называются максимизирующими уравнениями. Они образуют систему, состоящую из d нелинейных уравнений относительно d базисных переменных r_j ($j = \overline{1, d}$). Решив эту систему, можно определить максимизирующую точку двойственной программы B. Формула (6.56) задает максимальное значение двойственной задачи B для такой максимизирующей точки. Заметим, что базисные постоянные K_j и двойственные переменные $\delta_i(\mathbf{r})$ входят в разные части максимизирующих уравнений, такое разделение переменных и постоянных облегчает исследование зависимости оптимальных точек двойственной задачи B и прямой задачи A от коэффициентов c_i.

Описание алгоритма

На основании вышеизложенного можно сформулировать следующий алгоритм геометрического программирования.

Пусть задана прямая задача в виде (6.19)—(6.23).

1. Составляем задачу, двойственную к ней, (6.24)—(6.28).

2. Находим решение ограничений ортогональности (6.28) и нормализации (6.27) — δ^*. Если степень трудности задачи $d = n - m - 1 = 0$, то это решение единственно.

3. Используя значение $\delta^* = \{\delta_i^*\}$ $i = \overline{1,\ n}$, составляем систему уравнений (6.33), которую решаем относительно неизвестных $\mathbf{t}^* = \{t_j^*\}_j = \overline{1,\ m}$.

При этом для искомого решения $v\,(\delta^*) = g_0\,(\mathbf{t}^*)$.

4. Если же степень трудности задачи $d > 0$, то общее решение системы уравнений (6.27) — (6.28) будет иметь вид:

$$\delta_i\,(\mathbf{r}) = b_i^{(0)} + \sum_{j=1}^{d} r_j b_i^{(j)} \geqslant 0; \quad i = \overline{1,\ n};$$

где $\mathbf{r}$ — вектор произвольных параметров.

5. Используя условия оптимальности теоремы 5.17, составляем и решаем систему нелинейных уравнений (6.57) относительно неизвестных r_j, $j = \overline{1,\ d}$. Найдя ее решение $\mathbf{r}^*$ и подставляя в (6.53), определяем оптимальные значения двойственных переменных $\delta_i = \delta_i\,(\mathbf{r}^*)$, а подставляя δ^* в (6.59), определим оптимальное значение двойственной целевой функции

$$v\,(\delta^*) = \max v.$$

6. Переходим к прямой задаче и, используя соотношения (6.33), решаем систему уравнений относительно искомых значений t_1^*, t_2^*, ..., t_m^* при уже найденных значениях переменных

$$\{\delta_i^*\} \quad i = \overline{1,\ n}.$$

Пример 5.9. Пусть требуется минимизировать позином

$$g_0\,(\mathbf{t}) = 40t_1t_2 + 20t_2t_3, \tag{1}$$

при ограничении

$$g\,(\mathbf{t}) = \frac{1}{5}\,t_1^{-1}t_2^{-1/2} + \frac{3}{5}\,t_2^{-1}t_3^{-2/3} \leqslant 1. \tag{2}$$

Заметим, что в данной задаче число членов $n = 4$, а число переменных $m = 3$. Следовательно, степень трудности задачи $d = n - m - 1 = 0$.

Таким образом, решение двойственных ограничений единственно.

Двойственная задача к задаче (1) запишется в виде:

$$\max v\,(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1} \cdot \left(\frac{20}{\delta_2}\right)^{\delta_2} \cdot \left(\frac{1/5}{\delta_3}\right)^{\delta_3} \cdot \left(\frac{3/5}{\delta_4}\right)^{\delta_4} \cdot (\delta_3 + \delta_4)^{\delta_3 + \delta_4}, \tag{3}$$

при условиях

$$\delta_1 + \delta_2 = 1;$$

$$\delta_1 - \delta_3 = 0; \tag{4}$$

$$\delta_1 + \delta_2 - \frac{1}{2}\,\delta_3 - \delta_4 = 0; \tag{5}$$

$$\delta_2 - \frac{2}{3}\delta_4 = 0;$$

$$\delta_1 \geqslant 0, \quad \delta_2 \geqslant 0, \quad \delta_3 \geqslant 0, \quad \delta_4 \geqslant 0. \tag{6}$$

Условия (4), (5) образуют систему из четырех уравнений с четырьмя неизвестными и имеют единственное решение:

$$\delta_1^* = \frac{1}{2}; \quad \delta_2^* = {}^1/_2; \quad \delta_3^* = {}^1/_2; \quad \delta_4^* = {}^3/_4.$$

Из выражения для g (t) видно, что выполнены условия теорем 5.15 и 5.16 и прямая задача совместна. Применим эти теоремы для нахождения искомого решения t*. Находим значение двойственной функции

$$v(\delta^*) = v({}^1/_2, {}^1/_2, {}^1/_2, {}^3/_4) = 40.$$

Следовательно, по теореме двойственности 5.14

$$g_0(t_1^*, t_2^*, t_3^*) = 40.$$

Используя четвертое утверждение теоремы двойственности и соотношение (6.33), составим следующую систему уравнений:

$$40t_1t_2 = \delta_1^* \, v(\delta^*) = \frac{1}{2}\cdot 40, \tag{7.а}$$

$$20t_2t_3 = \delta_3^* \, v(\delta^*) = \frac{1}{2}\cdot 40, \tag{7.б}$$

$${}^1/_5 t_1^{-1} t_2^{-1/_2} = \frac{\delta_3}{\lambda} = \frac{\delta_3}{\delta_3+\delta_4} = \frac{2}{5}, \tag{7.в}$$

$$\frac{3}{5} t_2^{-1} t_3^{-2/_3} = \frac{\delta_4}{\delta_3+\delta_4} = \frac{3}{5}. \tag{7.г}$$

Чтобы решить систему (7), прологарифмируем обе части соотношений (7, *а*) — (7, *г*). Получим систему линейных уравнений относительно $\ln t_1$, $\ln t_2$, $\ln t_3$. Она имеет единственное решение $\ln t_1 = -\ln 2$; $\ln t_2 = 0$; $\ln t_3 = 0$, откуда искомое решение имеет вид

$$t_1^* = \frac{1}{2}; \quad t_2^* = 1; \quad t_3^* = 1.$$

Нетрудно убедиться непосредственно, что g (1/2, 1, 1) = 40.

Пример 5.10. Пусть требуется минимизировать позином

$$g_0(\mathrm{t}) = 40t_1^{-1}t_2^{-1/_2}t_3^{-1} + 20t_1t_3 + 20t_1t_2t_3, \tag{1}$$

при условии

$$g(\mathrm{t}) = \frac{1}{3}t_1^{-2}t_2^{-2} + \frac{4}{3}t_2^{1/_2}t_3^{-1} \leqslant 1. \tag{2}$$

Соответствующая этой задаче двойственная программа состоит в максимизации

$$\max v(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1}\cdot\left(\frac{20}{\delta_2}\right)^{\delta_2}\cdot\left(\frac{20}{\delta_3}\right)^{\delta_3}\cdot\left(\frac{{}^1/_3}{\delta_4}\right)^{\delta_4}\cdot\left(\frac{{}^4/_3}{\delta_5}\right)^{\delta_5}\cdot(\delta_4+\delta_5)^{\delta_4+\delta_5}, \tag{3}$$

при ограничениях

$$\begin{aligned} &\delta_1 + \delta_2 + \delta_3 = 1; \\ &-\delta_1 + \delta_2 + \delta_3 - 2\delta_4 = 0; \end{aligned} \tag{4}$$

$$-{}^1/_2\delta_1 + \delta_3 - 2\delta_4 + {}^1/_2\delta_5 = 0; \tag{5}$$

$$-\delta_1 + \delta_2 + \delta_3 - \delta_5 = 0;$$

$$\delta_1 \geqslant 0, \quad \delta_2 \geqslant 0, \quad \delta_3 \geqslant 0, \quad \delta_4 \geqslant 0, \quad \delta_5 \geqslant 0. \tag{6}$$

Степень трудности программы равна $n - m - 1 = d = 5 - 3 - 1 = 1$.

1. Решаем систему (4) — (6) методом Гаусса; результаты последовательных итераций приведены в табл. 5.4 — 5.8.

Отсюда получаем следующее базисное решение:

$$\delta_1^* = {}^1/_2, \quad \delta_2^* = {}^1/_4, \quad \delta_3^* = {}^1/_4, \quad \delta_4^* = 0, \quad \delta_5^* = 0.$$

Полное множество решений системы равно при $\delta_4 = r$

$$\delta_1 = {}^1/_2 - r,$$
$$\delta_2 = {}^1/_4 + {}^1/_2 r;$$
$$\delta_3 = {}^1/_4 + {}^1/_2 r;$$
$$\delta_4 = r;$$
$$\delta_5 = 2r.$$

Из этих уравнений, используя условия неотрицательности, получаем, что r должно удовлетворять неравенству

$$0 \leqslant r \leqslant {}^1/_2. \qquad (7)$$

2. Записав общее решение системы в виде: $\delta = \mathrm{b}^{(0)} + r\mathrm{b}^{(1)}$, определяем, что

$$\mathrm{b}^{(0)} = \left[\frac{1}{2} \;\; \frac{1}{4} \;\; \frac{1}{4} \;\; 0 \;\; 0\right]; \qquad (8)$$

$$\mathrm{b}^{(1)} = [-1 \;\; {}^1/_2 \;\; {}^1/_2 \;\; 1 \;\; 2]. \qquad (9)$$

3. Используя условия оптимальности (6.57) теоремы 5.17, определяем искомое значение r^*, при котором

$$v(\delta(r^*)) = \max_r v(r).$$

Вычисляем $K_1 = \prod\limits_{i=1}^{n} c_i^{b_i^{(1)}} = 40^{-1} \cdot 20^{1/2} \cdot 20^{1/2} \cdot ({}^1/_3)^1 \cdot ({}^4/_3)^2 = 20/40 \cdot 1/3 \cdot 16/9 =$ $= 8/27$.

Но в силу (6.57) с учетом $d = 1$ и $p = 1$ получим

$$K_1 = \prod_{i=1}^{5} \delta_i^{b_i^{(1)}}(r)\, \lambda_1^{-\lambda_1^{(1)}}(r), \qquad (10)$$

где $\delta_i(r)$ определяется из (6), $b_i^{(1)}$ определяется из (9),

$$\lambda_1(r) = \delta_4(r) + \delta_5(r) = r + 2r;$$
$$\lambda_1^{(1)} = b_4^{(1)} + b_5^{(1)} = 1 + 2 = 3.$$

Подставляя эти значения в (10), получим

$$K_1 = ({}^1/_2 - r)^{-1} \cdot \left(\frac{1}{4} + \frac{1}{2} r\right)^{1/2} \cdot \left(\frac{1}{4} + \frac{1}{2} r\right)^{1/2} \cdot (r)^1 (2r)^2 (r + 2r)^{-3} = \frac{8}{27}. \qquad (11)$$

Откуда

$$\frac{\left(\frac{1}{4} + \frac{1}{2} r\right) r 4r^2}{\left(\frac{1}{2} - r\right) 3^3 \cdot r^3} = \frac{8}{27}$$

или

$${}^1/_4 + {}^1/_2 r = 2\,({}^1/_2 - r). \qquad (12)$$

Из (12) определяем искомое значение $r^* = 0{,}3$.

Таблица 5.4

δ_1	δ_2	δ_3	δ_4	δ_5	A_0
$\boxed{1}$	1	1			1
—1	1	1	—2		0
—1/2		1	—2	1/2	0
—1	1	1		—1	0

Таблица 5.5

δ_1	δ_2	δ_3	δ_4	δ_5	A_0
1	1	1			1
0	$\boxed{2}$	2	—2		1
0	1/2	$1^1/_2$	—2	1/2	1/2
0	2	2		—1	1

Таблица 5.6

δ_1	δ_2	δ_3	δ_4	δ_5	A_0
1	0	0	1		1/2
0	1	1	—1		1/2
0	0	1	$—1^1/_2$	$\boxed{1/2}$	1/4
0	0	0	2	—1	0

Таблица 5.7

δ_1	δ_2	δ_3	δ_4	δ_5	A_0
1		0	1		1/2
	1	1	—1		1/2
		2	$—3^1/_2$	1	1/2
		$\boxed{2}$	—1	0	1/2

Таблица 5.8

δ_1	δ_2	δ_3	δ_4	δ_5	A_0
1		0	1		1/2
	1	0	—1/2		1/4
		0	—2	1	0
		1	—1/2		1/4

Заметим, что это значение удовлетворяет неравенству (7). Подставляя $r^* = 0{,}3$ в (6), находим оптимальные значения двойственных переменных: $\delta_1^* = 1/2 - r^* = 0{,}2$; $\delta_2^* = {}^1/_4 + {}^1/_2 r = 0{,}4$; $\delta_3^* = \delta_2^* = 0{,}4$; $\delta_4^* = 0{,}3$; $\delta_5^* = 0{,}6$.

Тогда максимум двойственной функции будет равен:

$$v(\delta^*) = \left(\frac{40}{0{,}2}\right)^{0{,}2} \cdot \left(\frac{20}{0{,}4}\right)^{0{,}4} \cdot \left(\frac{20}{0{,}4}\right)^{0{,}4} \cdot \left(\frac{{}^1/_3}{0{,}3}\right)^{0{,}3} \cdot \left(\frac{{}^4/_3}{0{,}6}\right)^{0{,}6} \cdot (0{,}9)^{0{,}9} =$$

$$= 200^{0{,}2} \cdot (50)^{0{,}4} \cdot (50)^{0{,}4} \cdot \left(\frac{10}{9}\right)^{0{,}3} \cdot \left(\frac{10}{9}\right)^{0{,}6} \cdot (0{,}9)^{0{,}9} \cdot 2^{0{,}6} = 4^{0{,}2} \cdot 50 \cdot 4^{0{,}3} = 100.$$

Итак, $v = (\delta^*) = 100$.

Теперь, используя соотношения (6.33) из теоремы двойственности 5.15, нетрудно составить уравнения относительно неизвестных прямой задачи t_1, t_2, t_3:

$$40 t_1^{-1} t_2^{-1/2} t_3^{-1} = \delta_1^* v(\delta^*) = 0{,}2 \cdot 100 = 20; \tag{14}$$

$$20 t_1 t_3 = \delta_2^* v(\delta^*) = 0{,}4 \cdot 100 = 40; \tag{15}$$

$$20 t_1 t_2 t_3 = \delta_3^* v(\delta^*) = 0{,}4 \cdot 100 = 40; \tag{16}$$

$$\frac{1}{3} t_1^{-2} t_2^{-2} = \frac{\delta_4^*}{\delta_4^* + \delta_5^*} = \frac{0{,}3}{0{,}3 + 0{,}6} = \frac{1}{3}. \tag{17}$$

Разделив уравнение (16) на (15), получим $t_2^* = 40/40 = 1$.

Подставляя $t_2^* = 1$ в (17), получим $1/3\ t_1^{-2} = 1/3$, откуда $t_1^* = 1$. И наконец, подставив $t_1^* = 1$ в (15), находим $t_3^* = 2$. Итак, мы нашли следующее оптимальное решение прямой задачи $t_1^* = 1$; $t_2^* = 1$; $t_3^* = 2$;

$$g(t^*) = v(\delta^*) = 100.$$

Вопросы для самопроверки

1. Укажите основные трудности, возникающие при решении задачи нелинейного программирования.
2. При каких ограничениях можно применять метод множителей Лагранжа?
3. Какой смысл имеют множители Лагранжа?
4. Сформулируйте теорему Куна — Таккера. В чем состоит ее значение для решения НП-задач?
5. Поясните смысл условий регулярности. Какой вид они могут принимать?
6. Поясните смысл условий дополняющей нежесткости.
7. Запишите условия теоремы Куна—Таккера для НП-задачи при дополнительных ограничениях неотрицательности.
8. Как связаны между собой оптимальные решения пары двойственных ЛП-задач и седловая точка соответствующей функции Лагранжа?
9. Какая связь существует между оптимальным решением задачи нелинейного программирования и седловой точкой функции Лагранжа для этой задачи?
10. Какое свойство геометрического неравенства лежит в основе методов геометрического программирования?
11. Сформулируйте основную теорему геометрического программирования и покажите ее связь с теорией двойственности.
12. Какой смысл имеет условие «существует такое решение t, что $g_i(t) < 0$, $i = \overline{1, p}$», используемое в этой теореме?
13. Что такое степень трудности задачи геометрического программирования?

Глава 6. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Динамическое программирование — это вычислительный метод для решения задач определенной структуры. Динамическое программирование возникло и сформировалось в 1950—1953 гг. благодаря работам Р. Беллмана и его сотрудников. Первые задачи, которые привели к появлению вычислительного метода динамического программирования, являлись динамическими задачами управления запасами.

§ 1. СУЩНОСТЬ ВЫЧИСЛИТЕЛЬНОГО МЕТОДА

Сущность вычислительного метода рассмотрим на следующем примере.

Найти

$$\max_{x_1, \dots, x_n} z = \sum_{j=1}^{n} f_j(x_j) \tag{1.1}$$

при условиях

$$\sum_{j=1}^{n} a_j x_j \leqslant b; \quad a_j > 0, \quad x_j \geqslant 0. \tag{1.2}$$

Целевая функция задачи является суммой функций от одной переменной. Такая функция называется *аддитивной*. Если все $f_i(x_i)$, $i = 1, 2, \dots, n$ — выпуклые (вогнутые), то для решения может быть применен метод множителей Лагранжа. Однако, если имеется много локальных максимумов, то этот метод дает лишь одно из таких решений. В случае, если требуется найти глобальный максимум, метод множителей Лагранжа неприменим.

Рассмотрим метод, обеспечивающий решение задачи. Считаем все $\{a_j\}$, $j = 1, \dots, n$ и b целыми числами. Предположим также, что в задаче все переменные $\{x_j\}$ могут принимать только целочисленные значения.

Введем следующие значения. Через z^* обозначим абсолютный максимум z, при условии $\sum_{j=1}^{n} a_j x_j \leqslant b$. Выбираем значение x_n и, зафиксировав его, максимизируем z по всем остальным переменным $x_1, x_2, \dots, x_{n-1}$. Предположим, что такая максимизация проведена для всех возможных значений x_n. Тогда z^* будет наибольшим из всех возможных значений z. Формально этот процесс записывается так:

$$\max_{x_1, x_2, \dots, x_{n-1}} \left\{ \sum_{j=1}^{n} f_j(x_j) \right\} = f_n(x_n) + \max_{x_1, x_2, \dots, x_{n-1}} \left\{ \sum_{j=1}^{n-1} f_j(x_j) \right\}, \tag{1.3}$$

причем

$$\sum_{j=1}^{n-1} a_j x_j \leqslant b - a_n x_n.$$

Так как $\max \sum_{j=1}^{n-1} f_j(x_j)$ для неотрицательных целых чисел, удовлетворяющих условию $\sum_{j=1}^{n-1} a_j x_j \leqslant b - a_n x_n$, зависит от $b - a_n x_n$, то обозначим

$$\max_{x_1, \ldots, x_{n-1}} \sum_{j=1}^{n-1} f_j(x_j) = \Lambda_{n-1}(b - a_n x_n). \tag{1.4}$$

Допустим, что мы вычислили $\Lambda_{n-1}(b - a_n x_n)$ для всех допустимых целых значений $x_n = \left\{0, 1, \ldots, \left[\frac{b}{a_n}\right]\right\}$, где $\left[\frac{b}{a_n}\right]$ обозначает целую часть $\frac{b}{a_n}$.

Очевидно, что

$$z^* = \max_{x_n \geqslant 0} [f_n(x_n) + \Lambda_{n-1}(b - a_n x_n)]. \tag{1.5}$$

Для вычисления (1.5) определяем значения $f_n(x_n)$ и $\Lambda_{n-1}(b - a_n x_n)$ для всех допустимых значений x_n и выбираем максимальное. Одновременно находим и x_n^*. Таким образом, если бы была известна функция $\Lambda_{n-1}(b - a_n x_n)$, то вся задача свелась бы к задаче с одной переменной.

Покажем, как можно вычислять $\Lambda_{n-1}(b - a_n x_n)$.

Очевидно,

$$\Lambda_{n-1}(\xi) = \max_{x_1, \ldots, x_{n-1}} \sum_{j=1}^{n-1} f_j(x_j); \quad \sum_{j=1}^{n-1} a_j x_j \leqslant \xi.$$

Рассуждая, как выше, получаем

$$\Lambda_{n-1}(\xi) = \max_{x_{n-1}} [f_{n-1}(x_{n-1}) + \Lambda_{n-2}(\xi - a_{n-1} x_{n-1})], \tag{1.6}$$

где

$$\Lambda_{n-2}(\xi - a_{n-1} x_{n-1}) = \max_{x_1, \ldots, x_{n-1}} \sum_{j=1}^{n-2} f_j(x_j), \tag{1.7}$$

причем максимум берем по всем неотрицательным целым $x_1, \ldots, x_{n-2}$, удовлетворяющим условию $\sum_{j=1}^{n-2} a_j x_j \leqslant \xi - a_{n-1} x_{n-1}$.

Далее вычисляем $\Lambda_{n-2}(\xi)$, $\Lambda_{n-3}(\xi)$ и т. д., пока на последнем шаге не придем к

$$\Lambda_1(\xi) = \max_{0 \leqslant x_1 \leqslant \left[\frac{\xi}{a_1}\right]} f_1(x_1). \tag{1.8}$$

Чтобы решить задачу, процесс вычислений необходимо вести в обратном порядке, начиная с $\Lambda_1(\xi)$. Зафиксировав начало интервала и изменяя верхний его конец ξ, вычисляем

$$\Lambda_1(\xi) = \max f_1(x_1), \quad 0 \leqslant x_1 \leqslant \left[\frac{\xi}{a_1}\right]$$

для всех значений $\xi = 0, 1, \ldots, b$.

Оптимальное решение первого шага обозначим через $\hat{x}_1(\xi)$. Строим таблицу динамического программирования следующего вида (табл. 6.1).

Таблица 6.1

ξ	$\Lambda_1(\xi)$	$\hat{x}_1(\xi)$
0		
1		
. . .	. . .	. . .
b		

Вычислив $\Lambda_1(\xi)$, найдем $\Lambda_2(\xi)$, используя соотношение

$$\Lambda_2(\xi) = \max_{0 \leqslant x_2 \leqslant \left[\frac{\xi}{a_2}\right]} [f_2(x_2) + \Lambda_1(\xi - a_2x_2)].$$

Вычисляем последовательно $\Lambda_2(\xi)$ для всех $\xi = 0, 1, \ldots, b$, используя при этом результаты табл. 6.1.

Обозначим $\varphi_2(0;\ \xi) = f_2(0) + \Lambda_1(\xi)$.

Тогда

$$\varphi_2(1;\ \xi) = f_2(1) + \Lambda_1(\xi - a_2),\ \ldots,$$

$$\varphi_2\left(\frac{\xi}{a_2};\ \xi\right) = f_2\left(\frac{\xi}{a_2}\right) + \Lambda_1\left(\xi - a_2\left[\frac{\xi}{a_2}\right]\right).$$

Наибольшее из этих чисел и есть $\Lambda_2(\xi)$. Одновременно находим и $\hat{x}_2(\xi)$. Затем строим таблицу, аналогичную табл. 5.1, для $\Lambda_2(\xi)$ и $\hat{x}_2(\xi)$ $(\xi = 0, 1, \ldots, b)$. Так продолжается до вычисления $\Lambda_{n-1}(\xi)$ для $\xi = 0, 1, \ldots, b$. Функцию $\Lambda_n(\xi)$ табулировать не нужно, так как достаточно определить лишь $\Lambda_n(b) = z^*$. Одновременно находим и оптимальное значение для переменной $\overset{*}{x}_n(b)$.

Для нахождения значений всех остальных переменных $\overset{*}{x}_{n-1}, \overset{*}{x}_{n-2}, \ldots$ $\ldots, \overset{*}{x}_1$ следует использовать уже вычисленные таблицы $(n-1)$-го; $(n-2)$-го и т. д. шагов.

Из предыдущей таблицы $n - 1$-го шага находим $\overset{*}{x}_{n-1} = \hat{x}_{n-1}(b - a_n\overset{*}{x}_n)$.

Для этого берем $\xi = b - a_n x_n^*$. Аналогично определяем

$$x_{n-2}^* = x_{n-2}(b - a_n x_n^* - a_{n-1} x_{n-1}^*).$$

Как видим, динамическое программирование представляет собой направленный последовательный перебор вариантов, который обязательно приводит к *глобальному максимуму*. Для применения метода динамического программирования необходимо табулировать функции $\Lambda_1(\xi), \ldots, \Lambda_{n-1}(\xi)$ для всех допустимых значений ξ.

Сравним по числу необходимых операций динамическое программирование с простым перебором. Для упрощения расчета примем, что все $a_1 = a_2 = \ldots = a_n = 1$.

При простом переборе число возможных вариантов решений (при условии целочисленности всех переменных) равно числу способов, которыми можно разместить b одинаковых шаров в n урн.

Оно составляет

$$C_{n+b-1}^{b} = \frac{(n+b-1)!}{b!\,(n-1)!}.$$

При $n = 5$, $b = 20$ имеем $C_{24}^{4} = 10\,626$.

Оценим число операций, требуемых для решения этой задачи методом динамического программирования.

Для вычисления $\Lambda_k(\xi)$ при фиксированном ξ необходимо провести $(\xi + 1)$ вычислений. Поэтому для заполнения одной таблицы $\Lambda_k(\xi)$ необходимо проделать

$$\sum_{\xi=0}^{b} (\xi + 1) = \frac{(b+1)(b+2)}{2!} \text{ операций.}$$

Следовательно, для вычисления всех функций $\Lambda_1(\xi), \Lambda_2(\xi), \ldots$ $\ldots, \Lambda_{n-1}(\xi)$ необходимо $(n-1)\dfrac{(b+1)(b+2)}{2!}$ операций.

С учетом вычисления функции $\Lambda_n(b)$ общее число операций

$$N_\Sigma = \frac{(n-1)(b+1)(b+2)}{2!} + (b+1) = \frac{(b+1)[(n-1)(b+2)+2]}{2!}.$$

Подведем итоги. Рассмотренную задачу можно трактовать как задачу распределения с одним ограниченным источником сырья

$$\sum_{j=1}^{n} a_j x_j \leqslant b,$$

где x_j — количество сырья, используемое в j-м способе производства. Тогда $f_j(x_j)$ — доход от переработки j-м способом x_j единиц сырья. Поэтому $\Lambda_k(\xi)$ можно рассматривать как максимальный доход от первых k способов производства, когда общее количество сырья равно ξ единиц. Потому данная задача представляет собой n-шаговый процесс принятия решений, где на j-м шаге принимается решение, какое количество сырья из общего его объема следует направить на переработку по j-му способу. Как видим, структура задачи не изменяется от числа шагов, т. е. *задача инвариантна относительно n*. Решение для

k-шаговой задачи получается из решения для (k — 1)-шаговой задачи путем добавления k-го шага и использования результата, полученного для всех предыдущих шагов.

Следовательно, сущность динамического подхода состоит в замене решения данной n-шаговой задачи последовательностью задач: одношаговой, двухшаговой и т. д.

Отметим основные необходимые свойства задач, к которым возможно применить этот подход.

1. Задача должна допускать интерпретацию как n-шаговый процесс принятия решений.

2. Задача должна быть определена для любого числа шагов и иметь структуру, не зависящую от их числа.

3. При рассмотрении k-шаговой задачи должно быть задано некоторое множество параметров, описывающих состояние системы, от которых зависят оптимальные значения переменных. Причем это множество не должно изменяться при увеличении числа шагов (в рассмотренной задаче таким параметром было ξ — количество сырья).

4. Выбор решения (управления) на k-ом шаге не должен оказывать влияния на предыдущие решения, кроме необходимого пересчета переменных.

Пусть ξ — вектор параметров, описывающих состояние процесса (вектор состояния). Тогда $\Lambda_k(\xi)$ — оптимальное значение целевой функции для k-шагового процесса при условии ξ. $\Lambda_k(\xi)$ будем называть *функцией состояния*.

Пусть X — вектор переменных управления, подлежащих выбору на k-м шаге. Тогда для задач, к которым можно применить метод динамического программирования, должно выполняться следующее основное рекуррентное соотношение:

$$\Lambda_k(\xi) = \max_{\bar{X}_k} \{f(X_k;\ \xi) + \Lambda_{k-1}[T(\xi;\ X_k)]\}, \qquad (1.9)$$

где $T(\xi;\ X_k)$ — вектор состояния предыдущего (k — 1)-го шага, при условиях ξ и X_k.

Сформулируем теперь *принцип оптимальности* Р. Беллмана, который обосновывает это рекуррентное соотношение. Оптимальная стратегия обладает следующим свойством: *каковы бы ни были начальное состояние* ξ *и начальная стратегия* X_1, *последующие стратегии* (решения) *должны быть оптимальны по отношению к состоянию* $T(\xi;\ X_1)$, *получающемуся в результате предыдущего решения*, т. е. по отношению к текущему состоянию системы.

Таким образом, принцип оптимальности утверждает, что оптимальное управление системой на каждом шаге не зависит от предыстории процесса, т. е. как система пришла в текущее состояние, а определяется только самим этим состоянием. Системы (процессы), для которых справедливо это свойство, называются марковскими.

Задача о выборе траектории

Основной принцип динамического программирования можно продемонстрировать на следующем примере [11].

Пример 6.1. Самолет (или другой летательный аппарат), находящийся в точке S_0 и имеющий скорость V_0 и высоту H_0, должен быть поднят на заданную высоту $H_{\text{кон}}$, а скорость его доведена до значения $V_{\text{кон}}$. Известен расход горючего, необходимый для подъема с любой высоты H_1 на высоту H_2 при постоянной скорости V, и расход горючего, необходимый для изменения скорости от V_1 до V_2 ($V_2 > V_1$) при неизменной высоте H. *Найти оптимальный режим набора высоты и скорости, при котором общий расход горючего будет минимальным.*

Предположим, что весь процесс набора высоты и скорости можно разделить на ряд последовательных шагов (этапов) и за каждый шаг самолет увеличивает только высоту или только скорость. Изобразим состояние самолета точкой на плоскости VOH, где абсцисса отвечает скорости V, а ордината — высоте H. Процесс перемещения точки S из начального состояния S_0 в конечное $S_{\text{кон}}$ изобразится тогда на плоскости VOH некоторой ступенчатой ломаной линией (рис. 6.1). Эта траектория и будет характеризовать управление набором высоты и скорости.

Из всех траекторий нужно выбрать такую, на которой выбранный критерий — расход горючего Q — будет минимальным.

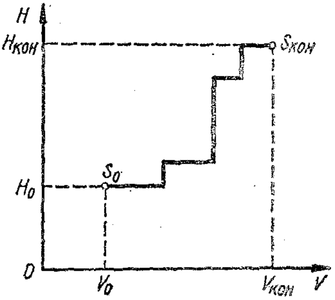


Рис. 6.1.

Для решения задачи методом динамического программирования разделим высоту $H_{\text{кон}} - H_0$ на n_1 равных частей, а скорость $V_{\text{кон}} - V_0$ на n_2 равных частей.

Таким образом, за один шаг процесса происходит либо изменение высоты на величину $\Delta H = \dfrac{H_{\text{кон}} - H_0}{n_1}$, либо изменение скорости — $\Delta V = \dfrac{V_{\text{кон}} - V_0}{n_2}$. Суммарное число шагов процесса по переводу самолета из S_0 в $S_{\text{кон}}$ равно $m = n_1 + n_2$.

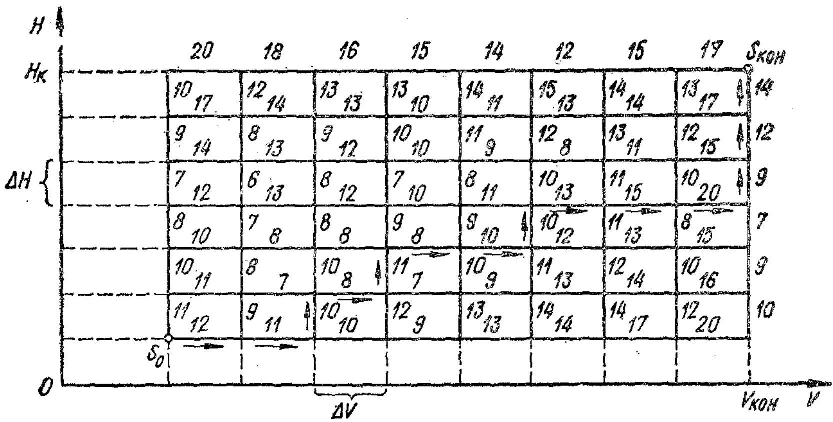


Рис. 6.2.

С каждым шагом связан определенный расход горючего. Запишем на каждом из отрезков соответствующую величину расхода горючего (в условных единицах). Тогда условия задачи представлены на рис. 6.2, а траектории, отмеченной стрелками, соответствует расход горючего

$$Q = 12 + 11 + 10 + 8 + 11 + 8 + 10 + 10 + 13 + 15 + 20 + 9 + 12 + 14 =$$

$$= 163 \text{ у. е. т.}$$

Число всех возможных траекторий достаточно велико, а потому простой перебор неприемлем. Поскольку конечное состояние $S_{\text{кон}}$ известно, то построение оптимальной траектории начнем с конца.

В концевую точку $S_{\text{кон}}$ можно переместиться только из двух соседних (B_1 и B_2), причем из каждой только одним способом (рис. 6.3), а поэтому выбора оптимального управления на последнем шаге нет,— оно единственно. Если на предпоследнем шаге мы находились в B_1, то перемещение в $S_{\text{кон}}$ происходит по горизонтали и тратится 17 у. е. т., если же находились в B_2, то движемся по вертикали и тратим 14 единиц. Эти минимальные величины расходов обводим кружками и ставим у точек B_1 и B_2.

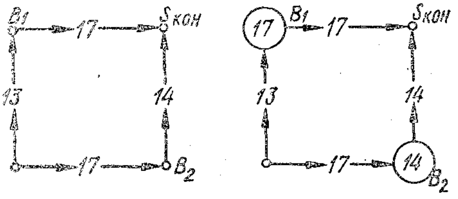


Рис. 6.3.

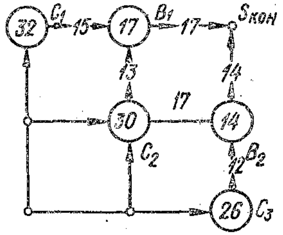


Рис. 6.4.

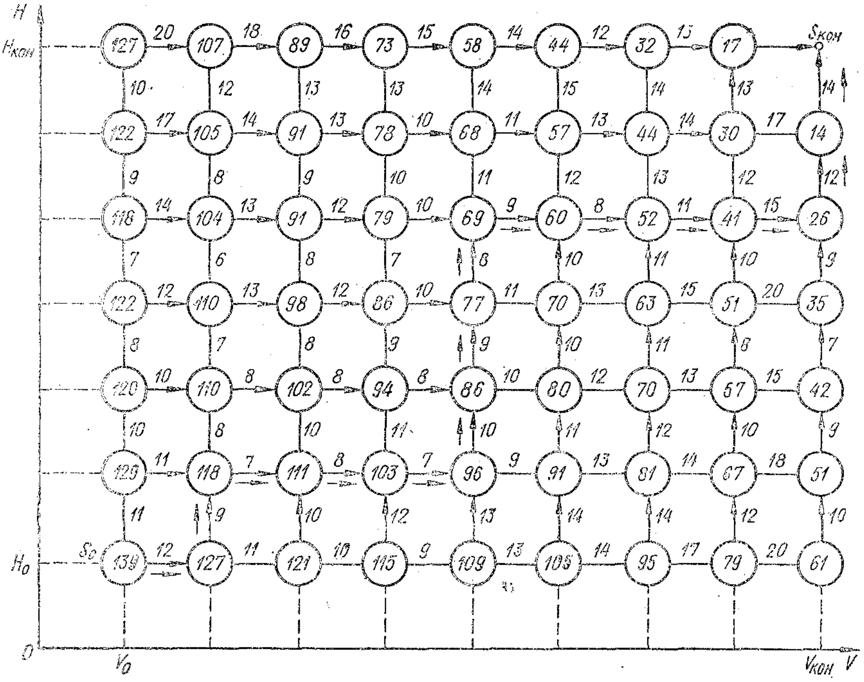


Рис. 6.5.

Таким образом, число в кружке у данной точки всегда означает минимальный расход горючего для перемещения самолета из данной точки в конечную, а оптимальная траектория указывается стрелками.

Переходим к планированию предпоследнего шага. Для этого рассмотрим пункты, из которых можно попасть в B_1 и B_2 за один шаг. Такими точками являются C_1, C_2

и C_3. Для каждой из них необходимо найти оптимальный путь в $S_{\text{кон}}$ и соответствующий ему расход горючего. В частности, для C_1 выбора нет, нужно двигаться по горизонтали и тратить $15 + 17 = 32$ у. е. т. Этот оптимальный путь пометим стрелкой. Для точки C_2 выбор есть: из нее можно идти в $S_{\text{кон}}$ через B_1 или через B_2. В первом случае израсходуется $13 + 17 = 30$ единиц, а во втором — $17 + 14 = 31$ единица. Итак, оптимальный путь из C_2 в $S_{\text{кон}}$ лежит через B_1.

Для точки C_3 путь в $S_{\text{кон}}$ единственный — по вертикали, ему соответствует расход в $12 + 14 = 26$ единиц горючего; эту величину обводим кружком при C_3, а стрелкой укажем оптимальное движение (рис. 6.4).

Таким образом, переходя от точки к точке справа налево и сверху вниз, для каждой узловой точки находим оптимальную траекторию, ведущую в $S_{\text{кон}}$, и соответствующую ей величину минимальных расходов. Эту величину будем обводить кружком. В конце концов этот процесс заканчивается, приведя к начальной точке S_0. Двигаясь теперь от нее в направлении стрелок, мы последовательно проходим искомую траекторию от начала в конец.

На рис. 6.5 приведен окончательный результат. Оптимальная траектория отмечена дополнительными стрелками. Число 139 означает минимальный расход горючего, отвечающий этой траектории.

Поскольку в данной задаче фиксированы и начальная, и конечная точки, то процесс построения оптимальной траектории можно было бы провести и в противоположном направлении, от начала в конец. Тогда числа в кружках означали бы минимальный расход горючего при перемещении самолета из начала — точки S_0 в точку, соответствующую числу в кружке. Очевидно, что окончательный результат — оптимальная траектория — при этом не изменится.

Задачи последовательного принятия решений

Структура рассмотренной задачи распределения ресурсов такова, что по мере построения процесса переменные x_i $(i = 1, ..., n)$ могли определяться в произвольном порядке. Однако существует много задач, где решения должны определяться в строгой временной последовательности, и потому переменные x_i и x_j нельзя менять местами. Подобные задачи называют задачами последовательного принятия решений. Для этих задач имеется выбор между двумя разными схемами решения:

1) решение задачи в прямом направлении, когда первый шаг схемы динамического программирования соответствует первому по времени принимаемому решению;

2) решение в обратном направлении, когда последний шаг схемы отвечает первому фактически принимаемому решению.

Для объяснения этих двух схем решения рассмотрим такую задачу.

Задача об использовании рабочей силы. Производителю работ нужно определить оптимальное число работников в каждый из n месяцев. Производственные задания для каждого месяца известны. Допустим, что в j-й месяц идеальное число рабочих — m_j. Если бы производитель работ мог увольнять и принимать новых рабочих без дополнительных затрат, то он мог бы в j-й месяц принять ровно m_j рабочих $(j = 1, 2, ..., n)$.

Предположим, что работа j-го месяца может быть выполнена и меньшим числом рабочих при сверхурочной работе. Пусть x_j фактическое число рабочих в j-й месяц. Затраты по изменению численности рабочих при переходе от $(j - 1)$-го месяца к j-му определяются функцией $f_j(x_j - x_{j-1})$.

В зависимости от знака величины $x_j - x_{j-1}$ функция $f_j(x_j - x_{j-1})$ определяет затраты по найму или увольнению. Очевидно, $f_j(0) \equiv 0$. Отклонение численности рабочих от m_j приводит к расходам $g_j(x_j - m_j)$, причем $g_j(0) \equiv 0$; $j = 1, 2, \ldots, n$. Считаем, что в начальный момент число рабочих составило m_0. Целевая функция задачи z определяется соотношением

$$z = \sum_{j=1}^{n} [f_j(x_j - x_{j-1}) + g_j(x_j - m_j)], \tag{1.10}$$

где $x_0 = m_0$. Очевидно, это задача с фиксированным началом x_0.

Выведем основное рекуррентное соотношение:

$$z = \min_{x_1,\ldots,x_n} \Big\{ f_1(x_1 - m_0) + g_1(x_1 - m_1) +$$

$$+ \sum_{j=2}^{n} \{f_j(x_j - x_{j-1}) + g_j(x_j - m_j)\};$$

$$\min_{x_2,\ldots,x_n} \sum_{j=2}^{n} \{f_j(x_j - x_{j-1}) + g(x_j - m_j)\} =$$

$$= \min_{x_2} \{[f_2(x_2 - x_1) + g_2(x_2 - m_2)] +$$

$$+ \min_{x_3,\ldots,x_n} \sum_{j=3}^{n} f_j(x_j - x_{j-1}) + g_j(x_j - m_j)\}. \tag{1.11}$$

Данную задачу удобно решать в обратном направлении. Обозначим

$$\Lambda_n(\xi) = \min_{x_n} [f_n(x_n - \xi) + g_n(x_n - m_n)], \tag{1.12}$$

где

$$\xi = x_{n-1}.$$

Основное рекуррентное соотношение имеет вид

$$\Lambda_k(\xi) = \min_{x_k} [f_k(x_k - \xi) + g_k(x_k - m_k) + \Lambda_{k+1}(x_k)], \tag{1.13}$$

где $\Lambda_k(\xi)$ — минимальные затраты за месяцы от k-го до n-го включительно, если количество работников в $(k - 1)$-й месяц равно ξ.

На последнем шаге определяется оптимальное число рабочих в первый месяц при условии, что на начало месяца их численность составляла m_0. Определив $\overset{*}{x}_1$, последовательно находим $\overset{*}{x}_2 = \hat{x}_2(\overset{*}{x}_1)$, $\overset{*}{x}_3 = \hat{x}_3(\overset{*}{x}_2)$ и т. д. Здесь решение в обратном направлении удобно, так как ничего не известно о числе рабочих на $(n + 1)$-й месяц, тогда как в начале процесса задано m_0.

Рассмотрим теперь случай, когда кроме m_0 задано m_{n+1}. На этот раз будем искать целые числа, обращающие в минимум выражение

$$z = \sum_{j=1}^{n} [f_j(x_j - x_{j-1}) + g_j(x_j - m_j) + f_{n+1}(m_{n+1} - x_n)]. \tag{1.14}$$

Здесь можно записать

$$z = \min_{x_n} \left\{ f_{n+1}(m_{n+1} - x_n) + \min_{x_1,\dots,x_{n-1}} \sum_{j=1}^{n} [f_j(x_j - x_{j-1}) + g_j(m_j - x_j)] \right\}.$$

Поскольку теперь задано конечное состояние системы m_{n+1}, то будем решать задачу в прямом направлении.

Определим последовательность функций состояния

$$\Lambda_k(\xi) = \min_{x_1,\dots,x_k} \left\{ f_{k+1}(\xi - x_k) + g_{k+1}(\xi - m_{k+1}) + \right.$$
$$\left. + \sum_{j=1}^{k} f_j(x_j - x_{j-1}) + g_j(m_j - x_j) \right\}, \quad k = 1, 2, \dots, n,$$

где минимум находят по целым и неотрицательным $x_1, \dots, x_k$. Тогда

$$\Lambda_1(\xi) = \min_{x_1} \{ f_2(\xi - x_1) + g_2(\xi - m_2) + f_1(x_1 - m_0) + g_1(m_1 - x_1) \}.$$

Основное рекуррентное соотношение ДП (динамического программирования)

$$\Lambda_k(\xi) = \min_{x_k} \{ f_{k+1}(\xi - x_k) + g_{k+1}(\xi - m_{k+1}) + \Lambda_{k-1}(x_k) \}. \qquad (1.15)$$

Функция $\Lambda_k(\xi)$ есть минимальные затраты в течение первых k месяцев при условии, что численность рабочих в $(k+1)$-й месяц равна ξ.

На последнем шаге при $k = n$, положив $\xi = m_{n+1}$, приходим к соотношению

$$\Lambda_n(m_{n+1}) = \min_{x_n} \{ f_{n+1}(m_{n+1} - x_n) + \Lambda_{n-1}(x_n) \}. \qquad (1.16)$$

Определив из (1.16) оптимальное значение $\overset{*}{x}_n$, по таблице предыдущего $(n - 1)$-го шага найдем искомые значения всех остальных переменных $\overset{*}{x}_{n-1}$, $\overset{*}{x}_{n-2}$, ..., $\overset{*}{x}_1$ (для этого необходимо подставить значение $\xi = \overset{*}{x}_n$). Подведем некоторые итоги. Пусть имеется задача последовательного принятия решений на n периодов. Если задано начальное состояние системы, то задача решается методом динамического программирования в обратном направлении. Если задано конечное состояние системы — то в прямом направлении. Наконец, если заданы как начальное, так и конечное состояние, то можно решать как в прямом, так и в обратном направлении. Результаты решения по обеим схемам совпадут.

Пример 6.2. Рассматривается задача об использовании рабочей силы на четыре периода:

$$j = 1, 2, 3, 4;$$

$$m_1 = 2; \quad m_2 = 5; \quad m_3 = 3; \quad m_4 = 1; \quad m_0 = 2.$$

Пусть функции $f_j(x_j)$ и $g_j(x_j)$ имеют следующий вид:

$$f_j(x_j) = \begin{cases} 10(x_j - x_{j-1}), & \text{если } x_j > x_{j-1} \\ 7(x_{j-1} - x_j), & \text{если } x_{j-1} > x_j \end{cases} \qquad f_j(0) = 0;$$

$$g_j(x_j) = \begin{cases} 8(x_j - m_j), & \text{если } x_j > m_j \\ 11(m_j - x_j), & \text{если } m_j > x_j \end{cases} \qquad g_j(0) = 0.$$

Таблица 6.2

ξ	x_4	$\Omega(\xi, x_4)$
0	0 1 2	11 10 28
1	0 1	18 0
2	0 1 2	25 7 8
3	0 1 2	32 14 15
4	0 1 2	39 21 22
5	0 1 2	46 28 29

Таблица 6.3

ξ	$\Lambda_4(\xi)$	x_4
0	10	1
1	0	1
2	7	1
3	15	1
4	21	1
5	28	1

Так как зафиксировано начало ($x_0 = m_0 = 2$), то задачу будем решать в обратном направлении.

Основное рекуррентное соотношение

$$\Lambda_k(\xi) = \min_{x_k} \{f_k(x_k - \xi) + g_k(x_k - m_k) + \Lambda_{k+1}(x_k)\},$$

$$\xi = x_{k-1};$$

$$\Lambda_n(\xi) = \min_{x_n} \{f_n(x_n - \xi) + g_n(x_n - m_n)\}, \quad n = 4.$$

Первый шаг

$$\Lambda_4(\xi) = \min_{x_4} \{f_4(x_4 - \xi) + g_4(x_4 - m_4)\} = \min_{x_4} \Omega_4(x_4, \xi).$$

Для отыскания $\Lambda_1(\xi)$ составляем таблицу значений $\Omega_4(x_4, \xi)$ (табл. 6.2). Значения $\min \Omega(x_4, \xi) = \Lambda_4(\xi)$ для всех ξ выбираем из табл. 6.2 и записываем в табл. 6.3.

1. $\xi = 0;\quad x_4 = 0;$

$\Omega = 0 + 11(1 - 0) = 11;$

$\xi = 0;\quad x_4 = 1;$

$\Omega = 10(1 - 0) + 0 = 10;$

$\xi = 0;\quad x_4 = 2;$

2. $\xi = 1;\quad x_4 = 0;$

$\Omega = 7(1 - 0) + 11(1 = 0) =$

$= 7 + 11 = 18;$

$\xi = 1;\quad x_4 = 1;$

$\Omega = 0 + 0 = 0.$

Таблица 6.4

$\xi = x_2$	x_3	Ω_3
0	0	40
	1	32
	2	38
1	0	50
	1	22
	2	28
2	0	57
	1	29
	2	18
	3	24
3	0	64
	1	36
	2	25
	3	14
4	2	32
	3	21
	4	29
5	2	39
	3	28
	4	36

Таблица 6.5

$\xi = x_2$	Λ_3	x_3	x_4
0	32	1	1
1	22	1	1
2	18	2	1
3	14	3	1
4	21	3	1
5	28	3	1

$\Omega = 10\,(2 - 0) + 8\,(2 - 1) = 28.$

3\. $\xi = 2;\quad x_4 = 0;$

$\Omega_4 = 7\,(2 - 0) + 11\,(1 - 0) = 25;$

$\xi = 2;\quad x_4 = 1;$

$\Omega_4 = 7\,(2 - 1) + 0 = 7;$

$\xi = 2;\quad x_4 = 2;$

$\Omega_4 = 7 \cdot 0 + 8\,(2 - 1) = 8.$

5\. $\xi = 4;\quad x_4 = 0;$

$\Omega = 7\,(4 - 0) + 11\,(1 - 0) = 39;$

$\xi = 4;\quad x_4 = 1;$

$\Omega_4 = 7\,(4 - 1) + 0 = 21;$

$\xi = 4;\quad x_4 = 2;$

$\Omega_4 = 7\,(4 - 2) + 8\,(2 - 1) = 22.$

4\. $\xi = 3;\quad x_4 = 0;$

$\Omega_4 = 7\,(3 - 0) + 11\,(1 - 0) = 32;$

$\xi = 3;\quad x_4 = 1;$

$\Omega_4 = 7\,(3 - 1) + 0 = 14;$

$\xi = 3;\quad x_4 = 2;$

$\Omega_4 = 7\,(3 - 2) + 8\,(2 - 1) = 15.$

6\. $\xi = 5;\quad x_4 = 0;$

$\Omega_4 = 7\,(5 - 0) + 11\,(1 - 0) = 46;$

$\xi = 5;\quad x_4 = 1;$

$\Omega_4 = 7\,(5 - 1) + 0 = 28;$

$\xi = 5;\quad x_4 = 2;$

$\Omega_4 = 7\,(5 - 2) + 8 = 29.$

Таблица 6.6

x_1	x_2	Ω_2
0	0	87
	1	76
	2	71
	3	66
	4	72
1	2	61
	3	56
	4	62
2	2	51
	4	52
3	2	58
	3	36
	4	42
4	2	65
	3	43
	4	32
5	4	39
	5	28

Таблица 6.7

$\xi = x_1$	Λ_2	x_2	x_3	x_4
0	66	3	3	1
1	56	3	3	1
2	46	3	3	1
3	36	3	3	1
4	32	4	3	1
5	28	5	3	1

Таблица 6.8

$x_0 = \xi$	x_1	Ω_1
2	2	46
	3	54

$\Lambda_1(x_0) = 46 \qquad x^*_1 = 2$

Второй шаг.

Выполняется аналогично первому. Вспомогательные результаты представлены в табл. 6.4, итоговые — в табл. 6.5.

$$\Lambda_3(\xi) = \min_{x_3} \{f_3(x_3 - \xi) + g_3(x_3 - m_3) + \Lambda_4(x_3)\}.$$

Третий шаг

$$\Lambda_2(\xi) = \min_{x_2} \{f_2(x_2 - \xi) + g_2(x_2 - m_2) + \Lambda_3(x_2)\}.$$

Четвертый шаг

$$\Lambda_1(x_0) = \min_{x_1} \{f_1(x_1 - x_0) + g_1(x_1 - m_1) + \Lambda_2(x_1)\}.$$

Из табл. 6.7 третьего шага по значению $x_1^* = \xi = 2$ находим оптимальные значения остальных переменных $x_2^* = 3$; $x_3^* = 3$; $x_4^* = 1$.

Динамическое программирование при непрерывных переменных

В отличие от других вычислительных методов динамическое программирование больше всего подходит для задач, где переменные — дискретны. В то же время оно может быть приспособлено к задачам с непрерывными переменными. Для этого непрерывные переменные квантуют, причем шаг дискретизации выбирают из конкретного содержания задачи.

Рассмотрим, как решается задача о распределении ресурсов при непрерывных переменных. Определим последовательность функций состояния

$$\Lambda_k(\xi) = \max \sum_{j=1}^{k} f_j(x_j) \qquad (1.16)$$

при условии

$$\sum_{j=1}^{k} a_j x_j \leqslant \xi, \quad k = 1,\ 2,\ \ldots,\ n.$$

Основное рекуррентное соотношение имеет вид:

$$\Lambda_k(\xi) = \max_{0\leqslant x\leqslant \frac{\xi}{a_k}} [f_k(x_k) + \Lambda_{k-1}(\xi - a_k x_k)]; \quad k = 2,\ 3,\ \ldots,\ n. \qquad (1.17)$$

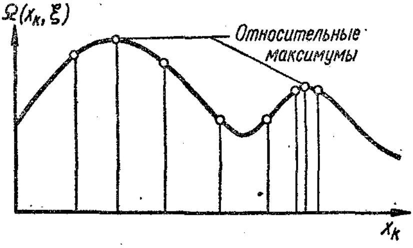


Рис.6.6.

Главное отличие, по сравнению со случаем дискретных переменных $x_1, \ldots, x_n$, возникает в ходе вычисления максимума по x_k и способе построения таблицы значений $\Lambda_k(\xi)$.

Рассмотрим задачу определения $\Lambda_k(\xi)$ при фиксированном ξ. Обозначим

$$\Omega_k(x_k;\ \xi) = f_k(x_k) + \Lambda_{k-1}(\xi - a_k x_k). \qquad (1.18)$$

Тогда

$$\Lambda_k(\xi) = \max_{0\leqslant x_k\leqslant \frac{\xi}{a_k}} \Omega_k(x_k;\ \xi).$$

Если $\Lambda_{k-1}(\xi)$ известна, то значения функции $\Omega_k(x_k, \xi)$ определены для всех x_k. Предположим, что график $\Omega_k(x_k;\ \xi)$ такой, как на рис. 6.6. Для определения $\Lambda_k(\xi)$ следует отыскать все относительные максимумы и выбрать абсолютный (глобальный).

Для этого функцию $\Omega_k(x_k;\ \xi)$ табулируют. При табулировании целесообразно сначала применять сетку с более грубым шагом, а в окрестности «подозрительных» точек перейти на более мелкий шаг.

Функции $\Lambda_k(\xi)$ приходится табулировать при всех допустимых значениях ξ, так как заранее неизвестно, какое из них может понадобиться при вычислении $\Lambda_{k+1}(\xi)$.

Задача вычисления $\Lambda_k(\xi)$ значительно упрощается при условии, что все $f_i(x_i)$ — выпуклы или вогнуты. Можно показать, что из выпуклости (вогнутости) всех $f_i(x_i)$; $i = 1, 2, \ldots, n$ следует выпуклость (вогнутость) $\Lambda_k(\xi)$.

Случай 1. Все функции $f_i(x_i)$ — выпуклы. Поскольку $\Lambda_{k-1}(\xi)$ — выпуклая функция по x, то и $\Lambda_{k-1}(\xi - a_k x_k)$ выпуклая относительно x_k и, следовательно, $\Omega_k(x_k;\ \xi) = f_k(x_k) + \Lambda_{k-1}(\xi - a_k x_k)$ также выпукла по x_k. Нас интересует $\max \Omega_k(x_k;\ \xi)$ при фиксированном ξ. Как известно, максимум выпуклой функции на замкнутом ограниченном множестве достигается в одной из крайних точек. В данном случае такими точками являются $x_k = 0$ и $x_k = \frac{\xi}{a_k}$.

Итак, в случае выпуклости всех $f_i(x_i)$, $i = 1, \ldots, n$ для нахождения $\max\limits_{0 \leqslant x_k \leqslant \frac{\xi}{a_k}} \Omega_k(x_k; \xi)$ достаточно проверить только крайние точки $x_k = 0$ и $x_k = \frac{\xi}{a_k}$.

Случай 2. Все функции $f_i(x_i)$ — вогнуты.

В таком случае $\Omega_k(x_k; \xi)$ при каждом фиксированном ξ есть вогнутая функция от x_k и, следовательно, всякий ее относительный максимум является также и глобальным.

Отыскать относительный максимум $\Omega_k(x_k; \xi)$ можно одним из известных методов, например градиентным.

§ 2. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ ДЛЯ ЗАДАЧ С НЕСКОЛЬКИМИ ОГРАНИЧЕНИЯМИ И ПЕРЕМЕННЫМИ

Рассмотрим задачу о распределении ресурсов, но уже при двух ограничениях [5]:

$$\sum_{j=1}^{n} a_{1j}x_j \leqslant b_1; \quad \sum_{j=1}^{n} a_{2j}x_j \leqslant b_2.$$

Поскольку в задаче имеется два вида ресурсов (b_1 и b_2), то необходимо ввести два параметра состояний ξ_1 и ξ_2.

Обозначим через

$$\Lambda_k(\xi_1; \xi_2) = \max_{x_1, \ldots, x_k} \sum_{i=1}^{n} f_i(x_i) \tag{2.1}$$

при условиях

$$\sum_{j=1}^{k} a_{1j}x_j \leqslant \xi_1; \quad \sum_{j=1}^{k} a_{2j}x_j \leqslant \xi_2; \quad x_i \geqslant 0 \quad (i = 1, 2, \ldots, k). \tag{2.2}$$

Запишем основное рекуррентное соотношение

$$\Lambda_k(\xi_1; \xi_2) = \max_{0 \leqslant x_k \leqslant \delta_k} \{f_k(x_k + \Lambda_{k-1}(\xi - a_{1k}x_k; \xi_2 - a_{2k}x_k)\},$$

где

$$\delta_k = \min\left\{\left[\frac{\xi_1}{a_{1k}}\right]; \left[\frac{\xi_2}{a_{2k}}\right]\right\}.$$

Одновременно с $\Lambda_k(\xi_1; \xi_2)$ определим и оптимальное решение $\hat{x}_k(\xi_1; \xi_2)$. На n-ом шаге определяем $\Lambda_n(b_1; b_2)$ и одновременно $x_n^0(b_1; b_2)$. Оптимальные значения остальных переменных $x_{n-1}^0, x_{n-2}^0, \ldots, x_1^0$ можно получить из таблиц предыдущих шагов при помощи соотношений:

$$x_{n-1}^0 = \hat{x}_{n-1}(b_1 - a_{1n}x_n^0; \quad b_2 - a_{2n}x_n^0);$$

$$x_{n-2}^0 = \hat{x}_{n-2}(b_1 - a_{1n}x_n^0 - a_{1n-1}x_{n-1}^0; \quad b_2 - a_{2n}x_n^0 - a_{2n-1}x_{n-1}^0)$$

и т. д.

Как видим, функции Λ_k (ξ_1; ξ_2) и x_k (ξ_1; ξ_2) являются функциями двух переменных. Если каждая из переменных ξ_1, ξ_2 может принимать 10^2 значений, то функцию Λ_k (ξ_1; ξ_2) приходится табулировать в 10^4 точках.

В случае трех параметров ξ_1, ξ_2 и ξ_3 при тех же предположениях требуется вычислять 10^6 значений функции Λ_k (ξ_1; ξ_2; ξ_3). Без специальных преобразований задачу, содержащую более трех параметров состояния, невозможно решить методом динамического программирования даже на сверхбыстродействующих ЭЦВМ из-за огромного объема вычислений.

Итак, наиболее серьезным препятствием для практического применения динамического программирования оказывается число параметров задачи. Это в свое время заставило Р. Беллмана заявить о так называемом *«проклятии многомерности»* [4, 5].

Задача с двумя переменными управления

Рассмотрим предыдущую задачу в несколько иной постановке, когда имеется две переменные управления x_i, y_i и два типа ресурсов, а доход f_i (x_i; y_i) есть функция количеств обоих типов ресурсов, распределяемых на данном шаге.

Итак, требуется найти

$$\max_{x_i \geqslant 0; y_i \geqslant 0} z = \sum_{i=1}^{n} f_i(x_i;\ y_i)$$

при условиях

$$\sum_{i=1}^{n} a_{1i} x_i \leqslant b_1; \quad \sum_{i=1}^{n} a_{2i} y_i \leqslant b_2.$$

Чтобы описать состояние системы, имеющей k шагов, на этот раз требуется два параметра ξ_1 и ξ_2. Определим функцию состояния

$$\Lambda_k(\xi_1;\ \xi_2) = \max_{\substack{x_1, \ldots, x_k \\ y_1, \ldots, y_k}} \sum_{i=1}^{k} f_i(x_i;\ y_i)$$

при условиях

$$\sum_{i=1}^{n} a_{1i} x_i \leqslant \xi_1; \quad \sum_{j=1}^{k} a_{2i} y_i \leqslant \xi_2.$$

Основное рекуррентное соотношение имеет вид

$$\Lambda_k(\xi_1;\ \xi_2) = \max_{x_k, y_k} [f_k(x_k,\ y_k) + \Lambda_{k-1})(\xi_1 - a_{1k} x_k;\ \xi_2 - a_{2k} y_k)], \qquad (2.3)$$

$$k = 2,\ 3,\ \ldots,\ n,$$

где максимум по x_k и y_k находят перебором всех возможных комбинаций.

Например, если x_k и y_k могут принимать по 100 значений, то для нахождения Λ_k (ξ_1; ξ_2) при фиксированных ξ_1 и ξ_2 следует перебрать

10^4 значений функции

$$\Omega_k(x_k, y_k, \xi_1, \xi_2) = [f_k(x_k, y_k) + \Lambda_{k-1}(\xi_1, \xi_2)] = f_k(x_k, y_k) + \\ + \Lambda_{k-1}(\xi_1 - a_{1k}x_k; \ \xi_2 - a_{2k}y_k).$$

Применение метода множителей Лагранжа для понижения размерности задачи

Метод, широко применяемый при решении задач нелинейного программирования, может быть применен для понижения размерности задач динамического программирования [45].

Рассмотрим такую задачу.

Найти

$$\max z = \sum_{i=1}^{n} f_i(x_i) \tag{2.4}$$

при условиях

$$\sum_{i=1}^{n} a_{1i}x_i = b_1, \tag{2.5}$$

$$\sum_{i=1}^{n} a_{2i}x_i = b_2. \tag{2.6}$$

Заметим, что пока мы рассматриваем ограничения-равенства. От задачи (2.4) перейдем к задаче с одним ограничением.

Найти

$$\max z_1 = \max_{\{x_i\}} \sum_{i=1}^{n} f_i(x_i) - \lambda \sum_{i=1}^{n} a_{2i}x_i \tag{2.7}$$

при условии

$$\sum_{i=1}^{n} a_{1i}x_i = b_1; \quad x_i \geqslant 0. \tag{2.8}$$

Задача (2.7) имеет только один параметр состояния и потому несравненно проще исходной.

Параметр λ — множитель Лагранжа; он играет роль цены на второй ресурс (b_2). Априори величина λ неизвестна, и потому задачу (2.7) приходится решать при нескольких произвольных значениях λ. Оптимальное решение задачи (2.7) будет зависеть от λ:

$$x_{i\,\text{опт}} = x_i^0(\lambda) \quad (i = 1, \ldots, n).$$

Если найденное решение $x^0(\lambda)$ удовлетворяет ограничению (2.6), которое было отброшено, то оно является искомым решением задачи (2.4). В противном случае значение λ нужно скорректировать. В частности, если окажется, что $\sum_{i=1}^{n} a_{2i}x_i^0(\lambda) > b_2$, то следует увеличить λ (так как по второму ресурсу имеется перерасход, то цену на него необходимо увеличить). В противном случае значение λ требуется уменьшить.

Для быстрого определения искомого λ применяют метод последовательных приближений, который основан на интерполяционной формуле (2.9). Если уже опробованы два значения λ_1, λ_2 и для них найдены оптимальные решения $\overset{*}{x}_1(\lambda_1)$, $\overset{*}{x}_2(\lambda_2)$, то на следующем шаге λ_3 оценивают из формулы

$$\lambda_3 = \frac{\lambda_2 - \lambda_1}{h_2 - h_1}(b_2 - h_1) + \lambda_1, \tag{2.9}$$

где

$$h_2 = \sum_{i=1}^{n} a_{2i} x_i^* (\lambda_2); \quad h_1 = \sum_{i=1}^{n} a_{2i} x_i^* (\lambda_1). \tag{2.10}$$

Покажем, что если оптимальное решение задачи (2.7) x^* удовлетворяет условию (2.6), то x^* — оптимально и для задачи (2.4).

Предположим, что это неверно, и обозначим через x^0 решение задачи (2.4). Тогда

$$z(x^0) = \sum_{j=1}^{n} f_j(x_j^0) > \sum_{j=1}^{n} f_j(x_j^*) = z(x^*).$$

Но, так как x^* — оптимальное решение для (2.7), то

$$z(x^*) = z(x^*) - \lambda \sum_{j=1}^{n} a_{2j} x_j^* \geqslant z(x^0) - \lambda \sum_{j=1}^{n} a_{2j} x_j^0.$$

А поскольку

$$\sum_{j=1}^{n} a_{2j} x_j^* = \sum_{j=1}^{n} a_{2j} x_j^0,$$

то должно быть

$$\sum_{j=1}^{n} f_j(x_j^*) = \sum_{j=1}^{n} f_j(x_j^0).$$

Основное рекуррентное соотношение для задачи (2.7) имеет вид:

$$\Lambda_k(\xi) = \max_{x_k} [f_k(x_k) - \lambda a_{2k} x_k + \Lambda_{k-1}(\xi - a_{2k} x_k)]. \tag{2.11}$$

Можно показать, что при возрастании λ от 0 до ∞ величина $\sum_{j=1}^{n} a_{2j} x_j(\lambda)$ убывает монотонно [5, 45]. Это важное свойство значительно облегчает поиск λ при решении конкретных задач. Теперь рассмотрим более общий случай ограничений.

Найти

$$\max_{\{x_j\} \geqslant 0} \sum_{j=1}^{n} f_j(x_j)$$

при условиях

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant c_i; \quad i = 1, 2, \ldots, M; \quad x_j \geqslant 0 \quad (j = 1, 2, \ldots, N).$$

Вводя $k < M$ множителей Лагранжа, можно так сформулировать новую задачу максимизации функции

$$\max_{\{x_j\}} \left(\sum_{j=1}^{n} f_j x_j - \sum_{i=1}^{k} \lambda_i \sum_{j=1}^{n} a_{ij} x_j \right), \tag{2.12}$$

при остальных $M - k$ ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant c_i, \qquad i = k + 1, \ \ldots, \ M. \tag{2.13}$$

Таким образом, задача сводится к определению функций от $M - k$ переменных вместе с поиском по k-мерному пространству переменных λ_i, удовлетворяющих первым k ограничениям. Исходная многомерная задача заменяется последовательностью задач значительно меньшей размерности, что позволяет существенно снизить требования к объему памяти ЦВМ.

Решение транспортной задачи методом динамического программирования

Рассмотрим транспортную задачу с двумя пунктами производства A_1, A_2 и n пунктами потребления B_1, ..., B_n.

Как обычно, x_{1j}— количество продукта, перевозимого из A_1 в B_j, x_{2j} — количество продукта, перевозимого из A_2 в B_j, $j = 1, \ldots, n$. Обозначим через $g_{ij}(x_{ij})$ стоимость перевозки по маршруту $A_i B_j$. Тогда транспортная задача формулируется так.

Найти

$$\min_{\{x_{ij}\}} \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij}(x_{ij}) \tag{2.14}$$

при условиях

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1, \ 2), \tag{2.15}$$

$$\sum_{i=1}^{2} x_{ij} = b_j \qquad (j = 1, \ \ldots, \ n). \tag{2.16}$$

Очевидно, если $g_{ij}(x_{ij}) = a_{ij} x_{ij}$, то задача легко решается методом линейного программирования. Однако здесь мы рассмотрим случай, когда $g_i(x_{ij})$ — нелинейные функции, и потому применим метод динамического программирования.

Покажем, что, несмотря на большое число ограничений, функцию состояния процесса можно описать одним параметром. Так как $x_{2j} = b_j - x_{1j}$, то ясно, что, определив x_{1j}, мы однозначно найдем и x_{2j}.

Рассмотрим случай, когда количество продуктов в пунктах A_1 и A_2 составляет ξ_1 и ξ_2 единиц. Поскольку $\xi_1 + \xi_2 = \sum_{j=1}^{n} b_j$, то $\xi_2 = \sum_{j=1}^{n} b_j - \xi_1$, и потому ξ_1 — есть единственный параметр состояния.

Введем последовательность функций

$$\Lambda_k(\xi_1) = \min_{\{x_{ij}\}} \sum_{j=1}^{n} g_{1j}(x_{1j}) + g_{2j}(x_{2j}), \tag{2.17}$$

где минимум берут по всем $\{x_{ij}\}$, удовлетворяющим условиям:

$$\sum_{j=1}^{k} x_{1j} = \xi_1; \quad \sum_{j=1}^{k} x_{2j} = \sum_{j=1}^{k} b_j - \xi_1; \quad x_{1j} + x_{2j} = b_j; \quad j = 1,\ 2,\ \ldots,\ k.$$

Основное рекуррентное соотношение тогда преобразуется:

$$\Lambda_k(\xi_1) = \min_{x_{1k}} [g_{1k}(x_{1k}) + g_{2k}(b_k - x_{1k}) + \Lambda_{k-1}(\xi_1 - x_{1k})], \tag{2.18}$$

где x_{1k} — удовлетворяет условию $0 \leqslant x_{1k} \leqslant \min\{\xi_1;\ b_k\}$.

При проведении вычислений функция $\Lambda_k(\xi_1)$ должна табулироваться для всех $\xi_1 = 0,\ 1,\ \ldots,\ \sum_{j=1}^{n} b_j$. На последнем шаге (при $k = n$) находим $\overset{*}{x}_{1n}$ и $\overset{*}{x}_{2n} = b_n - \overset{*}{x}_{1n}$. Остальные $\overset{*}{x}_{1k}$ находим последовательно из соотношений

$$x^*_{1,n-i} = \hat{x}_{1,n-i}\left(a_1 - \sum_{u=0}^{i-1} x^*_{1,n-u}\right); \quad i = 1,\ \ldots,\ n-1.$$

Рассмотрим теперь ту же задачу, но с тремя пунктами производства A_1, A_2, A_3 и n пунктами потребления. Функция состояния на этот раз определяется так:

$$\Lambda_k(\xi_1,\ \xi_2) = \min \sum_{i=1}^{3} \sum_{j=1}^{n} g_{ij}(x_{ij}), \quad k = 1,\ 2,\ \ldots,\ n, \tag{2.19}$$

где минимум берут по всем неотрицательным x_{ij}, удовлетворяющим условиям

$$\sum_{j=1}^{k} x_{1j} = \xi_1; \quad \sum_{j=1}^{k} x_{2j} = \xi_2; \quad \sum_{j=1}^{k} x_{3j} = \sum_{j=1}^{k} b_j - \xi_1 - \xi_2, \tag{2.20}$$

$$x_{1j} + x_{2j} + x_{3j} = b_j \quad (j = 1,\ 2,\ \ldots,\ k). \tag{2.21}$$

Рекуррентное соотношение записывается так:

$$\Lambda_k(\xi_1,\ \xi_2) = \min_{x_{1k}x_{2k}} \{g_{1k}(x_{1k}) + g_{2k}(x_{2k}) + g_{3k}(b_k - x_{1k} - x_{2k}) +$$

$$+ \Lambda_{k-1}(\xi_1 - x_{1k},\ \xi_2 - x_{2k})\}; \tag{2.22}$$

минимум в (2.22) берут по x_{1k} и x_{2k}, удовлетворяющим условиям

$$0 \leqslant x_{1k} \leqslant \min(\xi_1,\ b_k); \quad 0 \leqslant x_{2k} \leqslant \min(\xi_2,\ b_k);$$

$$x_{1k} + x_{2k} \leqslant b_k,$$

$$x_{3k} = b_k - x_{2k} - x_{1k} \leqslant \sum_{j=1}^{k} b_j - \xi_1 - \xi_2 \quad \text{или} \quad x_{1k} + x_{2k} \geqslant$$

$$\geqslant \xi_1 + \xi_2 - \sum_{j=1}^{k-1} b_j.$$

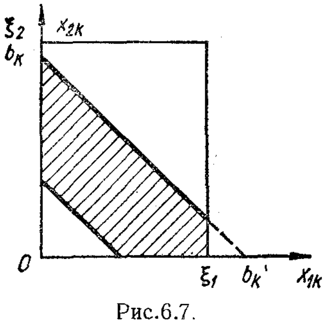


Рис.6.7.

Область, в которой отыскивается максимум, показана на рис. 6.7.

Функции $\lambda_k(\xi_1, \xi_2)$ должны табулироваться для всех целых ξ_1 и ξ_2, таких, что $\xi_1 + \xi_2 \leqslant \sum_{j=1}^{k} b_j$.

Для сокращения размерности задачи применим метод множителей Лагранжа. Допустим временно, что нет ограничений на количество продукта, отправляемого из пункта A_2, а вместо этого каждой отправляемой отсюда единице продукции приписана цена λ. Решаем задачу вида

$$\min z = \min \sum_{j=1}^{n} [g_{1j}(x_{1j}) + g_{2j}(x_{2j}) + g_{3j}(x_{3j})] + \lambda \sum_{j=1}^{n} x_{2j},$$

$$\sum_{j=1}^{n} x_{1j} = a_1, \qquad (2.23)$$

при условиях

$$x_{1j} + x_{2j} + x_{3j} = b_j \quad (j = 1, 2, \ldots, n),$$

$$x_{ij} \geqslant 0$$

для различных значений λ, пока не будет найдено такое, что оптимальное решение удовлетворяет $\sum_{j=1}^{n} x_{2j} = a_2$. Тогда автоматически удовлетворится и третье ограничение $\sum_{j=1}^{n} x_{3j} = a_3$.

Для решения задачи вводим функцию состояния

$$\Lambda_k(\xi) = \min_{\{x_{ij}\}} \sum_{j=1}^{k} [g_{1j}(x_{1j}) + g_{2j}(x_{2j}) + \lambda x_{2j} + g_{3j}(x_{3j})], \qquad (2.24)$$

где

$$\sum_{j=1}^{k} x_{1j} = \xi; \quad x_{1j} + x_{2j} + x_{3j} = b_j, \quad j = 1, 2, \ldots, n.$$

Основное рекуррентное соотношение динамического программирования можно записать:

$$\Lambda_k(\xi) = \min_{x_{1k}, x_{2k}} [g_{1k}(x_{1k}) + g_{2k}(x_{2k}) + \lambda x_{2k} + g_{3k}(b_k - x_{1k} -$$

$$- x_{2k}) + \Lambda_{k-1}(\xi - x_{1k})], \quad k = 2, 3, \ldots, n, \qquad (2.25)$$

где x_{1k} и x_{2k} удовлетворяют неравенствам $x_{1k} \geqslant 0$; $x_{2k} \geqslant 0$; $x_{1k} + x_{2k} \leqslant b_k$.

Таким образом, T-задача с тремя пунктами производства и n-пунктами потребления может быть сведена к последовательности n-шаговых задач с единственным параметром состояния.

Метод последовательных приближений

Одним из способов уменьшения трудностей, связанных с размерностью задачи, является метод последовательных приближений. Схема метода состоит в следующем. Для заданного функционального уравнения угадывается начальное приближенное решение.

Если предполагаемое решение не является истинным, то делается поправка, которая дает лучшее приближение. Пусть необходимо решить уравнение $T(u) = 0$ и пусть уравнение $S(u)$ легче решить, чем исходное. Тогда исходное уравнение записывается в виде

$$S(u) = S(u) - T(u).$$

Пусть за первое приближение u_0 взято решение уравнения $S(u) = 0$. Тогда следующее приближение u_1 определяется так:

$$S(u_1) = S(u_0) - T(u_0),$$

а $(n + 1)$-приближение находят из соотношения

$$S(u_{n+1}) = S(u_n) - T(u_n).$$

Если $S(u)$ выбрано правильно, а $T(u)$ обладает соответствующими свойствами, то последовательность $\{u_n\}$ будет сходиться к решению уравнения $T(u) = 0$.

Функциональное уравнение динамического программирования записывается так:

$$\Lambda(\xi) = \max_x [f(\mathbf{x}, \xi) + \Lambda(T\{\mathbf{x}, \xi\})], \tag{2.26}$$

где ξ — параметр состояния, $\mathbf{x}$ — решение (стратегия).

Если найти точное аналитическое решение $x(\xi)$ невозможно, задаются начальной функцией $\Lambda_0(\xi)$, а затем определяют последовательность функций $\Lambda_1(\xi)$, $\Lambda_2(\xi)$, ..., $\Lambda_n(\xi)$ с помощью рекуррентного соотношения:

$$\Lambda_{n+1}(\xi) = \max_{\bar{x}} [f(\mathbf{x}, \xi) + \Lambda_n\{T(\mathbf{x}, \xi)\}]_{(n=1,2,...)}. \tag{2.27}$$

Заметим, что в (2.27) входят фактически две неизвестные функции $\Lambda(\xi)$ и $\mathbf{x}(\xi)$. Рассмотрим так называемый *метод приближений в пространстве стратегий* [5]. Он начинается с произвольного выбора начальной политики (решения) x_0. Тогда соответствующая функция состояния $\Lambda_0(\xi)$ определяется как решение уравнения

$$\Lambda_0(\xi) = f(x_0;\ \xi) + \Lambda_0[T(x_0;\ \xi)],$$

где

$$x_0 = x_0(\xi). \tag{2.28}$$

Для получения следующего приближения определим $x_1 = x_1(\xi)$ как функцию, максимизирующую выражение

$$\max_{x_1} \{f(\xi;\ x_1) + \Lambda_0[T(\xi;\ x_1)]\}. \tag{2.29}$$

Затем определим $\Lambda_1(\xi)$ с помощью соотношения

$$\Lambda_1(\xi) = f(\xi;\ x_1) + \Lambda_0[T(\xi,\ x_1)]. \tag{2.30}$$

Продолжая таким образом, получаем две последовательности $\{x_n(\xi)\}$ и $\{\Lambda_n(\xi)\}$. Во многих случаях можно доказать монотонную сходимость этих функций к искомым: $\Lambda_n(\xi) \to \Lambda(\xi)$; $x_n(\xi) \to x(\xi)$.

В качестве примера применения этого метода рассмотрим задачу с двумя переменными управления. Требуется найти

$$\max f(x,\ y) = \sum_{i=1}^{N} f_i(x_i,\ y_i) \tag{2.31}$$

при условиях

$$\sum_{i=1}^{N} x_i = x;\ x_i \geqslant 0,\ \sum_{i=1}^{N} y_i = y;\ \ y \geqslant 0. \tag{2.32}$$

Пусть $\mathbf{X}_0 = \{x_i^0\}$ — начальное приближение в пространстве решений. Определяем максимум функции $\sum_{i=1}^{N} f_i(x_i^0;\ y_i)$ по всем y_i, удовлетворяющим (2.32). Вводим последовательность функций состояния $\Lambda_1(\xi)$; $\Lambda_2(\xi)$, ..., $\Lambda_n(\xi)$. Основное рекуррентное соотношение тогда преобразуется:

$$\begin{gathered}\Lambda_k(\xi) = \max_{0 \leqslant y_k \leqslant \xi} [f_k(x_k^0,\ y_k) + \Lambda_{k-1}(\xi - y_k)],\\ \Lambda_1(\xi) = f_1(x_1^0,\ \xi); \qquad k = 2,\ 3,\ \ldots,\ N.\end{gathered} \tag{2.33}$$

Этот процесс для каждого значения ξ дает множество значений $\{y_i^0\} = \mathbf{Y}_0$. Используя эти значения y_i^0, переходим к задаче максимизации функции $\sum_{i=1}^{k} f_i(x_i;\ y_i^0)$. Решается она с помощью рекуррентного соотношения, аналогичного (2.33):

$$\Lambda_k'(\xi') = \max_{0 \leqslant x_k' \leqslant \xi'} [f_k(x_k',\ y_k^0) + \Lambda_{k-1}(\xi' - x_k')]. \tag{2.34}$$

Многократное повторение этого процесса дает две последовательности $\{x^n\}$; $\{y^n\}$, $n = 1,\ 2,\ \ldots$, которые при некоторых общих условиях сходятся к оптимальным x^*, y^*.

Пример 6.3. Вернемся теперь к транспортной задаче и решим ее методом последовательных приближений. Число пунктов производства или складов равно трем (A_1, A_2, A_3), а пунктов потребления — десяти. Схема вычислений такова.

Продукты из пунктов A_3 произвольно распределяются по всем потребителям при единственном условии, что $\sum_{j=1}^{n} x_{3j} = a_3$ и $x_{3j} = b_j$. В данном примере распределяли его равномерно. Затем вычитаем из величин спроса всех потребителей полученные ими количества и для остатков решаем задачу с двумя складами A_1, A_2. Полученное решение $\{x_{1j}\}$ используем на второй итерации, где решается T-задача для другой пары складов A_2 и A_3.

На следующей итерации фиксируем решение $\{x_{2j}\}$ предыдущего шага и решаем задачу для пары A_1, A_3. Процесс продолжается до тех пор, пока не получится устойчивое распределение для всех трех вариантов пар.

Условия задачи приведены в табл. 6.9, а результаты после каждой итерации — в табл. 6.10.

Таблица 6.9

Потребители	A_1			A_2			A_3			Спрос
	организационные расходы	x	x^2	организационные расходы	x	x^2	организационные расходы	x	x^2	
1		+1,0		+2,0	+3,1			+7,0		25
2	+1,0	+2,0			+4,1			+3,0		40
3		+3,0	+0,01		+2,1			+9,0		60
4		+1,5			+1,1	+0,10		+1,0		30
5		+2,5			+2,6			+1,0		20
6	+10,0	+5,0	—0,01		+3,0		+5,0	+2,0		30
7		+3,0		+5,0	+1,0	+0,20		+4,0		35
8		+6,0			+2,0		+6,0	+3,0		30
9	+8,0	+6,0	—0,05		+2,0			+5,0		25
10		+6,0			+5,0	+0,01		+6,0		40
										335

Функции стоимости перевозок:

$$g_{ij}(x) = a_{ij}x + b_{ij}x^2 + c_{ij}(x),$$

где $c_{ij}(x)$ — фиксированные издержки, так называемые организационные расходы. Коэффициенты a_{ij} приведены в столбце x (табл. 6.9), а b_{ij} — в столбце x^2;

x_1 — запас на складе 1 = 100,

x_2 — запас на складе 2 = 80,

x_3 — запас на складе 3 = 155.

Результаты, полученные методом последовательных приближений, приведены в табл. 6.10.

Результаты трех последних итераций полностью совпадают. Это свидетельствует о сходимости процесса. Для сравнения в табл. 6.10 приведено точное решение, найденное другим методом.

Таблица 6.10

	Итерация 1	Итерация 2	Итерация 3	Итерация 4	Итерация 5	Итерация 6	Итерация 7	Точное решение
1	10 0 15	10 15 0	10 15 0	25 0 0	25 0 0	25 0 0	25 0 0	25 0 0
2	25 0 15	25 0 15	40 0 0	40 0 0	40 0 0	40 0 0	40 0 0	40 0 0
3	5 40 15	5 55 0	5 55 0	0 60 0	0 60 0	0 60 0	0 60 0	5 55 0
4	15 0 15	15 0 15	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30
5	5 0 15	5 0 15	0 0 20	0 0 20	0 0 20	0 0 20	0 0 20	0 0 20
6	0 15 15	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30
7	20 0 15	20 0 15	35 0 0	35 0 0	35 0 0	35 0 0	35 0 0	30 0 5
8	0 15 15	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30	0 0 30
9	0 10 15	0 10 15	0 10 15	0 10 15	0 20 5	0 20 5	0 20 5	0 25 0
10	20 0 20	20 0 20	10 0 30	0 10 30	0 0 40	0 0 40	0 0 40	0 0 40
	Затраты = = 1126.25	Затраты = = 966,00	Затраты = = 921,25	Затраты = = 874,00	Затраты = = 853,00	Затраты = = 853,00	Затраты = = 853,00	Затраты = = 847,75

§ 3. ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ

Общие сведения

Задачи управления запасами составляют один из наиболее многочисленных классов экономических задач исследования операций, решение которых имеет важное народнохозяйственное значение. Особенно повышается значение этих задач в период массового внедрения АСУ. Правильное и своевременное определение оптимальной стратегии управления запасами, а также нормативного уровня запасов позволяет высвободить значительные оборотные средства, замороженные в виде запасов, что в конечном счете повышает эффективность используемых ресурсов.

Приведем основные характеристики моделей задач управления запасами [39].

Элементами (системы) задачи управления запасами являются: 1) система снабжения; 2) спрос на предметы снабжения; 3) возможность пополнения запасов; 4) функции затрат; 5) ограничения; 6) принятая стратегия управления запасами.

Рассмотрим подробнее каждый из этих элементов. Системы снабжения бывают: децентрализованные однокаскадные; централизованные многокаскадные.

Спрос на предметы снабжения бывает: стационарный или нестационарный; детерминированный или случайный.

Различают такие способы пополнения запасов: мгновенная поставка; задержка поставок на фиксированный интервал времени; задержка поставок на случайный интервал времени.

Функции затрат составляют в совокупности критерий эффективности принятой стратегии управления запасами и учитывают расходы на хранение, стоимость поставок, затраты, связанные с заказом каждой новой партии, затраты на штрафы.

Приведем возможные варианты составляющих функции затрат. Расходы на хранение бывают: пропорциональные среднему уровню положительного запаса за период времени существования положительного запаса; пропорциональные остатку к концу периода.

Стоимость поставки бывает: пропорциональной объему поставки; постоянной; пропорциональной числу номенклатур; пропорциональной необходимому приросту интенсивности производства.

Штрафы бывают таких видов: пропорциональные средней положительной недостаче за период; пропорциональные положительной недостаче к концу периода; постоянные; нелинейные функции от средней недостачи и продолжительности ее существования.

Ограничения в задаче управления запасами бывают: на максимальный объем запасов; максимальный вес; максимальную стоимость; среднюю стоимость; число поставок в заданом интервале времени; объем поставки; вероятность недостачи.

Простейшие модели управления запасами

Детерминированный стационарный спрос. Рассмотрим простейшую модель управления запасами с постоянной интенсивностью спроса μ и поставок λ. График изменения запасов показан на рис. 6.8.

Полный цикл работы системы имеет продолжительность T. Обозначим через $\hat{Y}$ предельный запас на складе. Считая расходы на хранение (штрафы) пропорциональными среднему запасу (дефициту) и

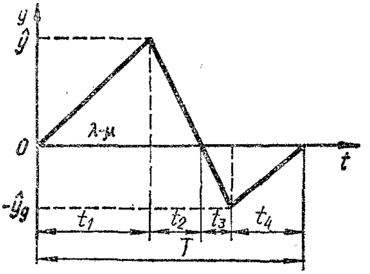


Рис.6.8.

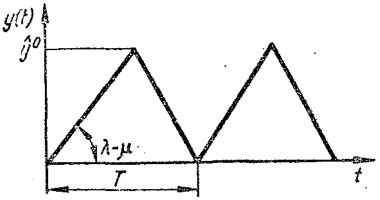


Рис. 6.9.

времени их существования соответственно, получим для функции затрат за цикл следующее выражение:

$$L_T = g + s\int_0^{t_1+t_2} y(t)\,dt - p\int_{t_1+t_2}^{T} y(t)\,dt, \tag{3.1}$$

где g — фиксированные расходы, связанные с запуском или заказом партии; s — удельные расходы на хранение единицы продукта в единицу времени; p — удельный штраф за дефицит единицы запаса в течение единицы времени.

Очевидно, что

$$y(t) = \begin{cases} (\lambda-\mu)t \text{ при } 0 \leqslant t \leqslant t_1; \\ \hat{Y} - \mu(t-t_1) \text{ при } t_1 < t \leqslant t_1+t_2+t_3; \\ -\hat{Y}_д + (\lambda-\mu)(t-t_1-t_2-t_3) \text{ при } t_1+t_2+t_3 < t \leqslant T. \end{cases} \tag{3.2}$$

Максимальный дефицит $\hat{Y}_д$ выражается через $\hat{Y}$ как

$$\hat{Y}_д = \frac{T-(t_1+t_2)}{t_1+t_2}\hat{Y}. \tag{3.3}$$

Подставив $t_1 = \frac{\hat{Y}}{\lambda-\mu}$ и $t_2 = \frac{\hat{Y}}{\mu}$, получим

$$\hat{Y}_д = \frac{\mu}{\lambda}[(\lambda-\mu)T - \hat{Y}]. \tag{3.4}$$

Перепишем функцию затрат с учетом линейности изменения уровня запаса:

$$L_T = g + \frac{s\lambda\hat{Y}^2}{2\mu(\lambda-\mu)} + \frac{p\lambda}{2\mu(\lambda-\mu)}\left[\frac{\mu}{\lambda}(\lambda-\mu)T - \hat{Y}\right]^2, \tag{3.5}$$

откуда затраты в единицу времени

$$L_{\text{ср}} = \frac{L_T}{T} = \frac{1}{T}\left[g + \frac{(p+s)\lambda\hat{Y}^2}{2\mu(\lambda-\mu)}\right] + \frac{p\mu}{2\lambda}(\lambda-\mu)T - p\hat{Y}; \qquad (3.6)$$

$$\frac{\partial L_{\text{ср}}}{\partial \hat{Y}} = \left[\frac{(p+s)\lambda\hat{Y}}{T\mu(\lambda-\mu)} - p\right] = 0; \qquad (3.7)$$

$$\frac{\partial L_{\text{ср}}}{\partial T} = \frac{p\mu}{2\lambda}(\lambda-\mu) - \frac{1}{T^2}\left[g + \frac{(p+s)\hat{Y}^2\lambda}{2\mu(\lambda-\mu)}\right] = 0. \qquad (3.8)$$

Решение системы (3.7) и (3.8) приводит к выражениям для оптимальных $\hat{Y}_0$ и T^0:

$$\hat{Y}^0 = \sqrt{\frac{2\mu g\left(1-\frac{\mu}{\lambda}\right)}{s\left(1+\frac{s}{p}\right)}}; \qquad (3.9)$$

$$T^0 = \sqrt{\frac{2g\left(1+\frac{s}{p}\right)}{\mu s\left(1-\frac{\mu}{\lambda}\right)}}. \qquad (3.10)$$

При этом достигается минимум затрат в единицу времени

$$L^0 = \sqrt{\frac{2\mu g s\left(1-\frac{\mu}{\lambda}\right)}{1+\frac{s}{p}}}. \qquad (3.11)$$

Из полученных соотношений при различных допущениях получаются следующие известные формулы теории запасов: *а*) недостачи (дефицит) полностью исключаются (рис. 6.9). Тогда, положив $p \to \infty$ и подставив $\frac{s}{p} = 0$ в (3.9) — (3.11), получим

$$\hat{Y}_1^0 = \sqrt{\frac{2\mu g}{s}\left(1-\frac{\mu}{\lambda}\right)}; \quad T_1^0 = \sqrt{\frac{2g}{\mu s\left(1-\frac{\mu}{\lambda}\right)}}; \quad L_1^0 = \sqrt{2\mu g s\left(1-\frac{\mu}{\lambda}\right)}; \qquad (3.12)$$

б) высокая интенсивность восполнения запаса, случай мгновенной поставки (рис. 6.10). Положив $\lambda \to \infty$, $\frac{\mu}{\lambda} = 0$, в (3.9) — (3.10) получим

$$Y_2^0 = \sqrt{\frac{2\mu g}{s\left(1+\frac{s}{p}\right)}}; \quad T_2 = \sqrt{\frac{2g\left(1+\frac{s}{p}\right)}{\mu s}}; \quad L_2^0 = \sqrt{\frac{2\mu g s}{1+\frac{s}{p}}}; \qquad (3.13)$$

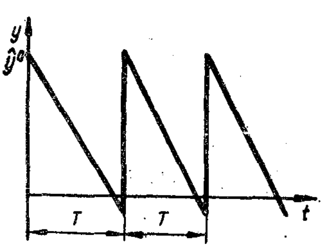


Рис.6.10.

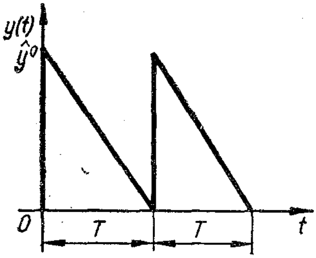


Рис.6.11.

в) дефицит не допускается, заказы выполняются мгновенно (рис. 6.11). Подставив $\lambda \to \infty$, $p \to \infty$, получим

$$Y^0 = \sqrt{\frac{2\mu g}{s}};\quad T^0 = \sqrt{\frac{2g}{\mu s}};\quad L^0 = \sqrt{2\mu g s}. \tag{3.14}$$

Равенства (3.14) называются *формулами Уилсона,* а величина Y^0 — *экономическим размером партии* [39,46].

Детерминированная модель при переменных издержках производства (переменная цена товара)

В приведенной модели предполагалось, что, кроме постоянных затрат на подготовку производства g, производственные издержки не учитываются. В более общем случае можно предположить, что издержки производства или приобретения запасов являются неубывающей функцией времени, т. е. вводим функцию

$$f(y) = \begin{cases} 0, & \text{если } y = 0; \\ g + c_1 y, & 0 < y < y^*; \\ g + c_2 y, & y > y^*. \end{cases} \tag{3.15}$$

Иногда вводятся скидки при закупках товаров в определенных количествах, т. е. стоимость 1 единицы продукции зависит от приобретенного объема товара. Так, предположим, что при закупке партии до 1000 шт. стоимость $c_1 = 1$ руб./шт.; а при закупке партии свыше 1000 шт. стоимость $c_2 = 95$ коп./шт.

При таких условиях иногда выгоднее превысить оптимальный размер партии, чтобы воспользоваться преимуществами скидки.

Итак, пусть μ — интенсивность спроса; s — затраты за хранение единицы продукции; g — постоянные расходы на заказ; p — штраф за дефицит единицы запаса в 1 единицу времени.

Примем функцию издержек производства в виде

$$f(y) = \begin{cases} 0, & \text{если } y = 0; \\ g + c_1 y, & \text{если } 0 < y < y^*; \\ g + c_2 y, & \text{если } y \geqslant y^*. \end{cases} \tag{3.16}$$

Общие издержки за период

$$L(y) = s\frac{1}{2}yT + f(y),$$

а издержки в единицу времени

$$L_T(y) = \frac{L(y)}{T} = \frac{1}{2}sy + \frac{1}{T}f(y) =$$

$$= \begin{cases} \frac{1}{2}sy + \frac{g\mu}{y} + c_1\mu, & \text{если} \quad y < y^*, \\ \frac{1}{2}sy + \frac{g\mu}{y} + c_2\mu, & \text{если} \quad y \geqslant y^*. \end{cases} \tag{3.17}$$

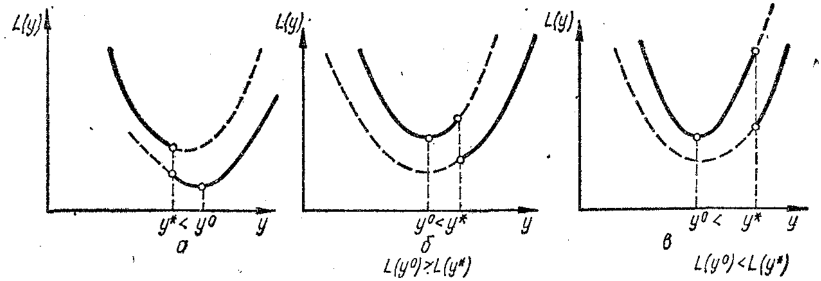


Рис.6.12.

Здесь использовано соотношение $T = \frac{y}{\mu}$ и $\frac{1}{T} = \frac{\mu}{y}$ (рис. 6.11). Следовательно, функция $L_T(y)$ терпит разрыв в точке $y = y^*$, и, очевидно, минимум $L_T(y)$ достигается либо в точке, где $\frac{\partial L(y)}{\partial y} = 0$, либо в точке разрыва. Определим эти точки:

$$\frac{\partial L_T(y)}{\partial y} = \frac{1}{2}s - \frac{g\mu}{y^2} = 0,$$

отсюда

$$y^0 = \sqrt{\frac{2g\mu}{s}}.$$

Рассмотрим случаи, когда $y^0 > y^*$ и $y^0 < y^*$. Если $y^0 \leqslant y^*$, то оптимальное значение L_T достигается при $y = y^0$ и тогда

$$L^0 = c_1\mu + \sqrt{2sg\mu}. \tag{3.18}$$

С другой стороны, при заказе товара в количестве y^* единиц по более низкой цене затраты в соответствии с (3.16) составят

$$c_2\mu + \frac{g\mu}{y^*} + \frac{1}{2}sy^*. \tag{3.19}$$

Сравнивая (3.18) с (3.19), видим, что выгодно заказывать товар партией y^* тогда и только тогда, когда $L(y^0) > L(y^*)$ (рис. 6.12), т. е. если

$$c_2\mu + \frac{g\mu}{y^*} + \frac{1}{2}sy^* < c_1\mu + \sqrt{2sg\mu} \tag{3.20}$$

или

$$c_1 - c_2 \geqslant \frac{1}{\mu}\left(\frac{g\mu}{y^*} + \frac{1}{2}sy^* - \sqrt{2sg\mu}\right).$$

Задача управления многономенклатурными запасами при ограничении на емкость склада

Рассмотрим задачу создания многономенклатурных запасов при ограничениях на суммарную емкость складов.

Пусть для i-го вида продукта (запаса) затраты на заказ составляют g_i, на хранение единицы продукта — s_i, спрос детерминированный с интенсивностью μ_i ($i = 1, \ldots, n$).

Предположим, что поставки заказов осуществляются мгновенно и дефицит не допускается, причем заказы по разным продуктам выполняются независимо. Тогда общие затраты по всем номенклатурам в единицу времени $\left(\text{при замене } T_i = \frac{y_i}{\mu_i}\right)$ определяются соотношением

$$L = \sum_{i=1}^{n} \frac{1}{2} s_i y_i + \frac{g_i \mu_i}{y_i}, \tag{3.21}$$

где y_i — размер заказа по i-й номенклатуре.

Если на запасы наложено ограничение и средний суммарный уровень запасов не должен превышать емкости складов C, то необходимо искать min L при ограничении

$$Y_{\text{ср}}^{\Sigma} = \frac{1}{2} \sum_{i=1}^{n} y_i \leqslant C. \tag{3.22}$$

Сначала определим оптимальный размер заказа по каждой номенклатуре по формуле Уилсона:

$$Y_i^0 = \sqrt{\frac{2\mu_i g_i}{s_i}}, \quad i = 1, \ldots, n. \tag{3.23}$$

Если $\frac{1}{2} \sum_{i=1}^{n} Y_i^0 \leqslant C$, то ограничение (3.22) выполняется и (3.23) определяет искомые значения $\{Y_i^0\}$. В противном случае, необходимо искать минимум (3.21) при ограничении $\frac{1}{2} \sum_{i=1}^{n} y_i = C$. Для этого применим метод множителей Лагранжа. Составим функцию

$$L_1 = \sum_{i=1}^{n} \left(\frac{1}{2} s_i y_i + \frac{g_i \mu_i}{y_i} \right) + \lambda \left(\sum_{i=1}^{n} y_i - 2C \right). \tag{3.24}$$

Оптимальные значения $\overset{*}{y}_1, \ldots, \overset{*}{y}_n$ определяются решением следующей системы уравнений:

$$\begin{cases} \frac{1}{2} s_i - \frac{g_i \mu_i}{y_i^2} + \lambda = 0, \\ 2C - \sum_{i=1}^{n} y_i = 0, \end{cases} \tag{3.25}$$

отсюда

$$y_i^* = \sqrt{\frac{2\mu_i g_i}{s_i + 2\lambda^*}}, \quad i = 1, \ldots, n, \tag{3.26}$$

причем λ^* является решением системы (3.25).

Модель управления запасами при вероятностном спросе и мгновенных поставках

Рассмотрим некоторые задачи управления запасами при вероятностном спросе.

Простейшим случаем управления запасами при вероятностном спросе является однократное принятие решения на пополнение запасов.

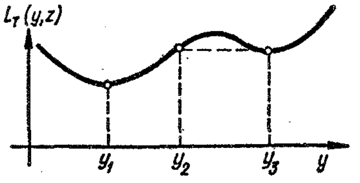


Рис.6.13.

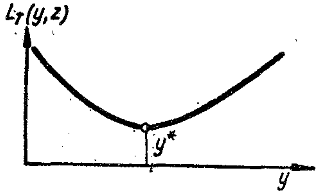


Рис.6.14.

I вариант. Рассмотрим следующую модель управления запасами при вероятностном спросе и мгновенных поставках. Пусть z — запас материальных средств к началу операции; y — запас после пополнения ($y \geqslant z$); $x \geqslant 0$ — случайный спрос за время операции T; $f(x)$ — плотность распределения спроса; $c(y-z)$ — расходы на пополнение запасов.

Предполагаем, что заказ на пополнение выполняется мгновенно. Если к концу операции на складе остается часть невостребованного запаса $(y-x) > 0$, то система снабжения несет избыточный расход $s_T(y-x)$ по хранению. При $x \geqslant y$ $s_T(y-x) = 0$.

При неполном удовлетворении спроса $x > y$ система платит штраф $p_T(x-y)$. Тогда математическое ожидание суммарных расходов системы

$$L_T(y, z) = \int\limits_0^y [s_T(y-x) f(x)\,dx + p_T \int\limits_y^\infty (x-y) f(x)\,dx + c(y-z)]. \quad (3.27)$$

Найдем, при каких значениях $y \geqslant z$ величина $L_T(y, z)$ будет минимальной. Для этого определим

$$\frac{\partial L_T}{\partial y}(y, z) = \int\limits_0^y [s'_T(y-x) f(x)\,dx - \int\limits_y^\infty p'_T(x-y) f(x)\,dx + c'(y-z) \quad (3.28)$$

(здесь учтено, что $s_T(0) = p_T(0) = 0$) и положим $\dfrac{\partial L_T}{\partial y} = 0$.

В общем случае функция $L_T(y, z)$ при фиксированных z может иметь несколько минимумов.

Обозначим через y_1 абсциссу абсолютного минимума $L_T(y, z)$, а через y_3, y_5, y_7 — следующие за ним относительные минимумы, причем $L(y_1) < L(y_3) < \ldots < L(y_{2n-1})$ (рис. 6.13).

Пусть далее y_2, y_4, y_6 — точки, удовлетворяющие следующим условиям: $y_1 < y_2 < y_3 < y_4 \ldots$; $L_T(y_2) = L_T(y_3)$, $L_T(y_4) = L_T(y_5)$ и т. д.

В этом случае оптимальная стратегия управления будет выглядеть так:

а) при $z < y_1$ заказывать $(y_1 - z)$;

б) при $y_1 \leqslant z \leqslant y_2$ ничего не заказывать;

в) при $y_2 < z < y_3$ заказывать $(y_3 - z)$ и т. д.

Приведем достаточные условия, при которых оптимальная стратегия имеет более простую форму, отвечающую одному минимуму $L_T(y, z)$ [35]:

а) $L_T(0, z)$ — не является относительным минимумом, и

$$L'_T(0, z) = -\int_0^\infty p'_T(x) f(x)\,dx + c'(0) < 0; \qquad \textbf{(3.29)}$$

б) уравнение $\frac{\partial L_T}{\partial y}(y, z) = 0$ имеет не более одного вещественного корня;

в) $L_T(y, z) \to \infty$; $y \to \infty$.

Обозначим через y^* решение уравнения $\frac{\partial L_T}{\partial y} = 0$ (рис. 6.14). Тогда оптимальная стратегия единственная и будет следующей:

1) при $z < y^*$ заказывать $y^* - z$;

2) при $z \geqslant y^*$ ничего не заказывать.

Поясним физический смысл условий: *а*) — экономическая целесообразность создания положительного запаса; *в*) — неэффективность чрезмерно больших запасов.

Условие *б*) может быть заменено менее жестким условием, если ограничиться некоторым классом распределений.

Так, если правый крайний относительный минимум в точке y^* является и абсолютным, то оптимальная стратегия будет иметь вид:

а) ничего не заказывать при $z \geqslant y^*$;

б) при $z < y^*$ заказывать $y^* - z$.

II вариант. Допустим, что стоимость пополнения запасов равна $g + c(y - z)$ при $y > z$ и 0 при $y = z$. Как видим, в этом варианте в сравнении с вариантом I добавился член g (плата за заказ). В таком случае заказ целесообразно производить лишь при условии

$$L_T(z, z) - L_T(y, z) \geqslant g. \qquad \textbf{(3.30)}$$

Если уравнение (3.30) имеет одно единственное решение $\hat{y}$, то оптимальная стратегия, как видно из рис. 6.15, имеет вид [39]:

а) при $z < \hat{y}$ заказывать $y^* - z$;

б) при $z \geqslant \hat{y}$ ничего не заказывать.

В литературе эта стратегия называется стратегией двух уровней или (S, s)-стратегией [39, 45].

Определение оптимальных уровней запасов y^* при вероятностном спросе и мгновенных поставках. Рассмотрим частный случай модели, когда функции $p_T(u)$; $s(u)$ и $c(u)$ — линейны. В этом случае величину y^* можно определять аналитически. Итак, рассмотрим функцию затрат за период $L_T(y, z)$ при условиях, что $c(u)$, $s_T(u)$ и $p_T(u)$ линейны.

Тогда

$$L_T(y,\ z) = s_T\int\limits_0^y (y-x)\,f(x)\,dx + p_T\int\limits_y^\infty (x-y)\,f(x)\,dx + c\,(y-z),$$

$$\frac{\partial L_T}{\partial y} = s_T\int\limits_0^y f(x)\,dx - p_T\int\limits_y^\infty f(x)\,dx + c = s_T F(y) - p_T\,[1-F(y)] + \\ + c = 0. \qquad \textbf{(3.31)}$$

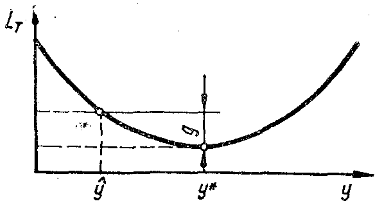


Рис.6.15.

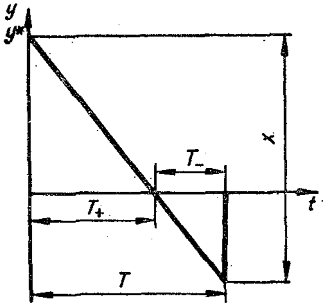


Рис.6.16.

Откуда для отыскания оптимального y^* получаем

$$F(y) = \frac{p_T - c}{p_T + s_T}. \qquad \textbf{(3.32)}$$

В частности, для спроса распределенного по закону Рэлея,

$$p(x) = \frac{x}{\sigma^2}\,e^{-\frac{x^2}{2\sigma^2}}$$

имеем

$$F(y) = 1 - e^{-\frac{y^2}{2\sigma^2}} = \frac{p_T - c}{p_T + s_T}.$$

Отсюда

$$y^* = \sigma\sqrt{2\ln\frac{p_T + s_T}{c + s_T}}.$$

Для показательного распределения спроса $f(x) = \frac{1}{\mu T}\,e^{-\frac{x}{\mu T}}$ получим

$$F(y^*) = 1 - e^{-\frac{y}{\mu T}} = \frac{p_T - c}{p_T + s_T},$$

откуда

$$y^* = \mu T \ln\frac{p_T + s_T}{c + s_T}.$$

Рассмотрим теперь случай дискретного распределения спроса

$$L_T(y) = s_T \sum_{x=0}^{y-1} (y-x)\, p(x) + p_T \sum_{x=y}^{\infty} (x-y)\, p(x) + c\,(y-z). \qquad (3.33)$$

Соответственно

$$L_T\ y+1) = s_T \sum_{x=0}^{y} (y-x+1)\, p(x) + p_T \sum_{x=y+1}^{\infty} (x-y-1) \times$$

$$\times\, p(x) + c\,(y-z+1). \qquad (3.34)$$

Вычислим приращение

$$\Delta L_T(y) = L_T(y+1) - L_T(y) = s_T \sum_{x=0}^{y} p(x) - p_T \sum_{x=y+1}^{\infty} p(x) + c =$$

$$= (p + s_T) \sum_{x=0}^{y} p(x) - p_T + c. \qquad (3.35)$$

Докажем существование и единственность оптимального решения y^*, для чего исследуем знак приращения $\Delta L_T(y)$. При $y = 0$

$$\Delta L_T(y) = -p_T + c < 0,$$

при $y \to \infty$

$$\Delta L_T(y) = c + s_T > 0. \qquad (3.36)$$

Итак, монотонность функции $\Delta L_T(y)$ * обеспечивает однократность смены знака приращения. Очевидно, выбор y * должен производиться из условий

$$\Delta L_T(y^* - 1) \leqslant 0 \quad \text{и} \quad \Delta L_T(y^*) \geqslant 0, \qquad (3.37)$$

которые можно свести к системе

$$\sum_{x=0}^{y^*-1} p(x) \leqslant \frac{p_T - c}{p_T + s_T} \leqslant \sum_{x=0}^{y^*} p(x). \qquad (3.38)$$

Определим теперь расходы за период так же, как и в детерминированном случае (т. е. с учетом положительности запаса и наличия дефицита) (рис. 6.16):

а) при $x \leqslant y^*$ средний положительный запас равен $y_+ = \frac{1}{2}[y^* + (y^* - x)] = y^* - \frac{x}{2}$, а время существования запаса $T_t = T$;

б) при $x \geqslant y^*$ получим средний положительный запас $y_+ = \frac{1}{2} y^*$, средний дефицит $y_- = \frac{1}{2}(x - y^*)$; время существования запаса $T_+ = \frac{y^*}{x} T$, а время дефицита $T_- = \frac{x - y^*}{x} T$.

Общие расходы в единицу времени

$$L(y^*) = \frac{1}{2} L_T(y^*, z) = s \sum_{x=0}^{y^*} \left(y^* - \frac{x}{2}\right) p(x) + s \sum_{x=y^*+1}^{\infty} \frac{1}{2} \frac{(y^*)^2}{x} p(x) +$$

$$+ p_T \sum_{x=y^*+1}^{\infty} \frac{1}{2} \frac{(x - y^*)}{x} p(x) + \frac{c}{T}(y^* - z).$$

Модель управления многономенклатурными запасами при вероятностном спросе и мгновенных периодических поставках

Выше была исследована модель системы управления однономенклатурными запасами при вероятностном спросе и мгновенных поставках. В этом разделе мы рассмотрим систему хранения многокомпонентных запасов, состоящих из N номенклатур, при условии, что запасы возобновляются периодически одновременно по всем номенклатурам.

Известно распределение спроса $f_i(x)$, $i = 1, \ldots, N$ по каждой номенклатуре за время периода T (период между поставками), а также расходы по хранению избыточного запаса $s_i(Y_i - x)$, где $\{Y_i\}$, $i = 1, \ldots, N$ — искомый оптимальный уровень запаса по каждой номенклатуре на начало периода. Предположим, что затраты на возобновление нормативных запасов по всем номенклатурам $i = \overline{1, N}$ определяются по формуле

$$\sum_{i=1}^{N} c_i(Y_i - z_i), \tag{3.39}$$

и задержкой в поставках можно пренебречь (мгновенные поставки).

Затраты на хранение следует, естественно, принять пропорциональными остаткам $(Y_i - x_i)$ на конец периода. Штраф за недостачу запаса система несет по i-й номенклатуре, для которой имеется максимум взвешенного ожидания потерь вследствие дефицита:

$$\max_i p_T \int_{y_i}^{\infty} (x - Y_i) f_i(x)\,dx. \tag{3.40}$$

Целевая функция — сумма затрат на период T определится из (3.41)

$$L_T(Y_1, Y_2, \ldots, Y_n) = \sum_{i=1}^{N} \left[c_i(Y_i - z_i) + s_i \int_0^{Y_i} (Y_i - x) f_i(x)\,dx \right] +$$

$$+ \max_i p_i \int_{Y_i}^{\infty} (x - Y_i) f_i(x)\,dx, \tag{3.41}$$

где $Y_1, \ldots, Y_N$ — нормативные уровни запасов по номенклатурам.

Из (3.41) следует, что для всех номенклатур j, удовлетворяющих условию

$$p_j \int_{Y_j}^{\infty} (x-Y_j) f_j(x_j) dx < \max_i p_i \int_{Y_j}^{\infty} (x-Y_i) f_i(x)\, dx, \tag{3.42}$$

транспортные издержки $c_i\,(Y_i - z_i)$ и издержки хранения $s_i\,(y_i - x_i)$ можно сократить до выполнения равенства в (3.42).

Предположим, что

$$\max_i p_i \int_{Y_i}^{\infty} (x-Y_i) f_i(x)\, dx = p_1 \int_{Y_1}^{\infty} (x-Y_1) f_1(x)\, dx. \tag{3.43}$$

Дифференцируя $L_T\,(y_1, \ldots, y_n)$, получаем в оптимальной точке

$$\frac{d}{dY_i} p_i \int_{Y_i}^{\infty} (x-Y_i) f_i(x)\, dx = -p_i \left[\int_{Y_i}^{\infty} f_i(x)\, dx\right] dY_i =$$

$$= -p_1 \left[\int_{Y_1}^{\infty} f(x)\, dx\right] dY_1,$$

откуда

$$\frac{dY_1}{dY_i} = \frac{p_i \int\limits_{Y_i}^{\infty} f_i(x)\, dx}{p_1 \int\limits_{Y_1}^{\infty} f_1(x)\, dx}. \tag{3.44}$$

Найдем оптимальные уровни запасов $Y_1^0, \ldots, Y_n^0$.

Для этого вычислим

$$\frac{dL}{dY_1} = \frac{\partial L}{\partial Y_1} + \sum_{j=2}^{N} \frac{\partial L_j}{\partial Y_j} \cdot \frac{\partial Y_j}{\partial Y_1} = \sum_{j=1}^{N} \frac{\partial L}{\partial Y_j} \cdot \frac{\partial Y_j}{\partial Y_1}. \tag{3.45}$$

Подставляя $\frac{dY_j}{dY_1}$ из (3.44), а также учитывая, что $\frac{\partial L}{\partial Y_j} = c_j + s_j \int_0^{Y_j} f_j(x)\, dx$, получаем

$$\frac{dL}{dY_1} = \sum_{i=1}^{N} \left[c_i + s_i \int_0^{Y_i} f_i(x)\, dx\right] \frac{p_1 \int\limits_{Y_1}^{\infty} f_1(x)\, dx}{p_i \int\limits_{Y_i}^{\infty} f_i(x)\, dx} - p_1 \int_{Y_1}^{\infty} f_1(x)\, dx = 0. \tag{3.46}$$

Итак, мы пришли к следующей системе уравнений относительно неизвестных $Y_1, \ldots, Y_N$:

$$\sum_{i=1}^{N} \frac{1}{p_i} \left[\frac{c_i + s_i}{\int\limits_{Y_i}^{\infty} f_i(x)\, dx} - s_i\right] - 1 = 0;$$

$$p_1\int\limits_{Y_1}^{\infty}(x-Y_1)\,f_1(x)\,dx = p_i\int\limits_{Y_i}^{\infty}(x-Y_i)\,f_i(x)\,dx; \tag{3.47}$$

$$i = 2, \ \ldots, \ N.$$

Необходимым условием существования решения этой системы в области $Y_1, \ldots, Y_N > 0$ является

$$\sum_{i=1}^{N}\frac{c_i}{p_i} < 1, \text{ так как } \frac{c_i+s_i}{\int\limits_{Y_i}^{\infty} f_i(x)\,dx} - s_i \geqslant \frac{c_i}{\int\limits_{Y_i}^{\infty} f_i(x)\,dy} \geqslant c_i.$$

Решение системы $\{Y_i^*\}$ отыскивается методом Ньютона [47].

Пусть дана система нелинейных алгебраических уравнений вида

$$\mathbf{V}(\mathbf{Y}) = 0. \tag{3.48}$$

Предположим, что известно m-е приближение $\mathbf{Y}^{(m)}$ корня $\mathbf{Y}^*$. Тогда

$$\mathbf{Y}^* = \mathbf{Y}^{(m)} + \Delta^{(m)}. \tag{3.49}$$

Подставив (3.49) в (3.48) и разложив $\mathbf{V}(\mathbf{Y}^{(m)} + \Delta^{(m)})$ в ряд Тэйлора в окрестности $\mathbf{Y}^m$, получим

$$\mathbf{V}(\mathbf{Y}^m + \Delta^m) = \mathbf{V}(\mathbf{Y}^m) + \mathbf{V}'(\mathbf{Y})\,\Delta^m = \mathbf{V}(\mathbf{Y}^m) + \mathbf{W}(\mathbf{Y})\,\Delta^m = 0,$$

где

$$\mathbf{W}(\mathbf{Y}) = \begin{bmatrix} \frac{\partial v_1}{\partial y_1}, & \frac{\partial v_1}{\partial y_2}, & \ldots, & \frac{\partial v_1}{\partial y_N} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial v_N}{\partial y_1}, & \frac{\partial v_N}{\partial y_2}, & \ldots, & \frac{\partial v_N}{\partial y_N} \end{bmatrix}.$$

Если

$$\det \mathbf{W}(\mathbf{Y}) \neq 0, \text{ то } \Delta^{(m)} = -\mathbf{W}^{-1}(\mathbf{Y}^m)\,\mathbf{V}(\mathbf{Y}^m). \tag{3.50}$$

Тогда

$$\mathbf{Y}^{m+1} = \mathbf{Y}^m + \Delta^m = \mathbf{Y}^{(m)} - \mathbf{W}^{-1}(\mathbf{Y}^m)\,\mathbf{V}(\mathbf{Y}^m).$$

В качестве начальных приближений Y_i^0 можно использовать y_i^0, вычисленные из соотношения, выведенного для модели с одним видом запасов (3.32)

$$F(y_i^0) = \frac{p_i - c_i}{p_i + s_i}, \quad i = 1, \ \ldots, \ N.$$

Если по некоторой номенклатуре i величины p_i и $x_{i\text{ср}}$ малы, то создавать запасы по ней нецелесообразно. Критерий для проверки этого условия получим в виде следующего равенства:

$$p_i x_{i\text{ср}} = p_i\int\limits_{0}^{\infty} x f_i(x)\,dx < p_1\int\limits_{y_1}^{\infty}(x-y_1)\,f_1(x)\,dx. \tag{3.51}$$

Таким образом, алгоритм отыскания $Y_1^*, \ldots, Y_N^*$ состоит в следующем:

1. Производится проверка условия

$$\sum_{i=1}^{N} \frac{c_i}{p_i} < 1. \tag{3.52}$$

Если оно выполняется, то переходим к решению системы нелинейных алгебраических уравнений (3.47). В противном случае исключаем номенклатуры, по которым запас не выгоден, т. е. для которых $\frac{c_i}{p_i} > 1$. Этот процесс устранения производится до тех пор, пока не станет выполняться (3.52).

2. Решаем систему нелинейных уравнений (3.47), и определяем оптимальные уровни запасов по всем номенклатурам $Y_1^*, Y_2^*, \ldots, Y_N^*$.

§ 4. ДИНАМИЧЕСКИЕ ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ

Выше были рассмотрены статические задачи управления запасами, где рассматривалось функционирование модели за один период.

В ряде таких задач удалось получить аналитические выражения для оптимального запаса y^0.

В случае, если рассматривается функционирование системы за n периодов, причем спрос непостоянен, приходим к *динамическим* моделям управления запасами. Эти задачи, как правило, не поддаются аналитическому решению, однако оптимальные уровни запасов по периодам могут быть определены, используя метод динамического программирования.

Нестационарный детерминированный спрос (конечношаговая модель)

Рассматривается задача управления запасами на n периодов, когда спрос за i-й период ($i = 1, \ldots, n$) определяется величиной d_i.

Вводим следующие обозначения: y_k — остаток запаса от (k —1)-го периода; d_k — суммарный спрос за k-й период; x_k — запас, создаваемый в k-й период (или заказ в k-м периоде).

Предположим, что заявка на пополнение запаса исполняется мгновенно, причем расходы на выполнение заказа величины x обозначим через $c_k\,(x_k,\ y_k)$.

Пусть $s_k\,(x_k + y_k - d_k)$ — расходы по хранению избыточного запаса в k-ом периоде.

Тогда суммарные расходы по снабжению за n периодов

$$L_{nT} = \sum_{k=1}^{n} c_k\,(y_k,\ x_k) + s_k\,(x_k + y_k - d_k), \tag{4.1}$$

причем $x_k + y_k - d_k = y_{k+1},\ k = 1,\ 2,\ \ldots,\ n$.

Требуется найти такие $\{x_k\}$ и $\{y_k\}$, при которых (4.1) обращается в минимум. Для минимизации L_{nT} воспользуемся методом динамического программирования. Будем последовательно минимизировать затраты за 1, 2, ..., n периодов, при условии, что на начало $(k+1)$-го периода имеется запас ξ.

Определим последовательность функций состояния

$$\Lambda_k(\xi) = \min_{\{x_j\}} \sum_{j=1}^{k} c_j(x_j, y_j) + s_j(x_j + y_j - d_j) \tag{4.2}$$

при условии, что

$$x_k + y_k - d_k = \xi.$$

Тогда основное рекуррентное соотношение таково:

$$\Lambda_k(\xi) = \min_{x_k} [f_k(x_k,\ y_k) + \Lambda_{k-1}(\xi - x_k + d_k)] =$$
$$= \min_{x_k} [f_k(x_k;\ \xi + d_k - x_k) + \Lambda_{k-1}(\xi - x_k + d_k)]. \tag{4.3}$$

На первом шаге при $k = 1$

$$\Lambda_1(\xi) = \min_{x_1} f_1(x_1,\ y_1) = \min_{x_1} [c_1(x_1,\ y_1) + s_1(x_1 + y_1 - d_1)] \tag{4.4}$$

при условии $\xi = x_1 + y_1 - d_1$.

Вычислив последовательно $\Lambda_1(\xi_1)$, $\Lambda_2(\xi_2)$, ..., на последнем шаге определим $\Lambda_n(\xi = y_{n+1})$ и $\overset{*}{x}_n(y_{n+1})$.

Оптимальные $\overset{*}{x}_k$ находим по формуле

$$\overset{*}{x}_k = \hat{x}_k\left(y_{n+1} + \sum_{j=k+1}^{n} d_j - \overset{*}{x}_j\right). \tag{4.5}$$

Рассмотрим частный случай, значительно упрощающий вычислительную схему.

1. Допустим, что все функции $c_j(x_j,\ y_j)$ — вогнуты по x_j и y_j.
2. Будем считать функцию s_j линейной. Тогда

$$s_j(x_j + y_j - d_j) = s_j(y_{j+1}) = s_j y_{j+1}.$$

Общее выражение для функции затрат будет

$$L_{nT} = \min_{\{x_j\}} \sum_{j=1}^{n} c_j(x_j) + s_j y_{j+1} = \min_{x_j} \sum_{j=1}^{n} f_j(x_j,\ y_{j+1}) \tag{4.6}$$

при условии

$$x_j + y_j - y_{j+1} = d_j. \tag{4.7}$$

Предположим, что y_1 и y_{n+1} заданы, закупки производятся лишь в начале периодов, а все требования в каждый текущий период должны быть удовлетворены. Нужно отыскать последовательность $\{x_j\}$ и $\{y_j\}$, минимизирующую функцию суммарных затрат L_{nT} (4.6).

Рассмотрим процедуру решения. Так как ищется минимум суммы вогнутых функций $f_j(x_j,\ y_{j+1})$, то оптимальным решением (4.6) является одна из крайних точек допустимого множества решений, определяемых ограничениями (4.7).

Так как число ограничений равно n, а общее число переменных $\{x_j\}$ и $\{y_j\}$ равно $2n$, то оптимальное решение содержит не более чем n

ненулевых значений переменных. Ввиду того, что x_j и y_j не могут быть равны нулю одновременно (иначе не будет выполняться (4.7)), оптимальное решение $\{x_j^0\}$ и $\{y_j^0\}$ обладает следующим свойством:

$$x_j^0 y_j^0 = 0. \tag{4.8}$$

Условие (4.8) эквивалентно следующей паре условий:

$$\begin{cases} \text{если } y_j^0 = 0, & \text{то } \ x_j^0 > 0; \\ \text{если } y_j^0 > 0, & \text{то } \ x_j^0 = 0. \end{cases} \tag{4.9}$$

Поясним смысл условий (4.8) и (4.9).

Заказ на изготовление новой партии x_j не поступает, если в начале периода j имелся запас $y_j > 0$. (Временем выполнения заказа пренебрегаем). Отсюда следует, что $x_j^0 = 0$ или $x_j^0 = d_j$, или $x_j^0 = d_j + d_{j+1}$ и т. д., т. е. *заказ равен спросу за целое число периодов.*

Поскольку $y_{n+1} = 0$, решаем задачу управления запасами в прямом направлении. Пусть $\Lambda_k(\xi)$ — минимальные затраты в течение k периодов, если запас на начало $(k+1)$-го периода равен ξ.

Основное рекуррентное соотношение

$$\Lambda_k(\xi) = \min_{x_k} [f_k(x_k, \ \xi) + \Lambda_{k-1}(\xi + d_k - x_k)], \tag{4.10}$$

причем $\xi = y_{k+1} = x_k + y_k - d_k, \quad k = 1, \ \ldots, \ n.$

Но так как $x_k^0 y_k^0 = 0$, то минимум (4.10) в силу вогнутости достигается в одной из крайних точек $x_k = 0$ или $x_k = \xi + d_k$.

Поэтому

$$\Lambda_k(\xi) = \min \begin{cases} f_k(\xi + d_k; \ \xi) + \Lambda_{k-1}(0) \\ f_k(0; \ \xi) + \Lambda_{k-1}(\xi + d_k). \end{cases} \tag{4.11}$$

Аналогично

$$\Lambda_{k-1}(\xi + d_k) = \min \begin{cases} f_{k-1}(\xi + d_k + d_{k-1}; \ \xi + d_k) + \Lambda_{k-2}(0) \\ f_{k-1}(0; \ \xi + d_k) + \Lambda_{k-2}(\xi + d_k + d_{k-1}). \end{cases} \tag{4.12}$$

Выписав выражения, аналогичные (4.12), для Λ_{k-2}, Λ_{k-3}, ..., получим

$$\Lambda_k(\xi) = \min_{1 \leqslant i \leqslant k} g_k(i), \tag{4.13}$$

где

$$g_k(i) = \Lambda_{i-1}(0) + f_i\left(\xi + \sum_{n=i}^{k} d_n; \ \xi + \sum_{n=i+1}^{k} d_n\right) + \\ + \sum_{j=i+1}^{k} f_j\left(0; \ \xi + \sum_{n=j+1}^{k} d_n\right), \tag{4.14}$$

$g_k(i)$ — расходы за k периодов, при условии, что последний заказ сделан в начале периода i.

Рассмотрим частный случай вида функции производственных затрат $c(x_j)$. Пусть

$$c_j(x_j) = \begin{cases} 0, & x_j = 0 \\ A_j + cx_j, & x_j > 0. \end{cases}$$

Тогда

$$\sum_{j=1}^{n} c_j(x_j) = \sum_{j=1}^{n} A_j\delta_j + c\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} A_j\delta_j + c\sum_{j=1}^{n} (y_{j+1} - y_j + d_j) =$$

$$= \sum_{j=1}^{n} A_j\delta_j + c\underbrace{\left(\sum_{j=1}^{n} d_j + (y_{n+1} - y_1)\right)}_{\text{const}} \sim \sum_{j=1}^{n} A_j\delta_j, \tag{4.15}$$

причем $\delta_j = 1$, если $x_j > 0$; $\delta_j = 0$, если $x_j = 0$.

В этом случае задача упрощается и сводится к задаче линейного программирования.

Найти

$$\min_{\{x_j; y_j\}} \sum_{j=1}^{n} (A_j\delta_j + s_j y_{j+1}) \tag{4.16}$$

при условиях

$$x_j + y_j - y_{j+1} = d_j;\ x_j y_j = 0, \tag{4.17}$$

$$\text{причем } \delta_j = 1, \text{ если } x_j > 0;\ \delta_j = 0 \text{ при } x_j = 0. \tag{4.18}$$

Используя (4.17), (4.18), а также условие, что $\xi = 0$, получим следующее выражение:

$$g_k(i) = \Lambda_{i-1}(0) + A_i + \sum_{j=i}^{k-1} s_j \sum_{n=j+1}^{k} d_n \tag{4.19}$$

и

$$\Lambda_k(0) = \min_{1 \leqslant i \leqslant k} g_k(i). \tag{4.20}$$

Дальнейшее упрощение в вычислительной схеме может быть получено, если учесть следующее обстоятельство. Если при вычислении $\Lambda_{k-1}(0)$ оказалось, что заказ, с помощью которого удовлетворялся спрос (k — 1)-го периода, должен поступить в начале периода v, то заказ, удовлетворяющий спрос k-го периода, должен поступить также не ранее периода v.

Следовательно, при вычислении $\Lambda_k(0)$ достаточно рассматривать

$$\Lambda_k(0) = \min_{v \leqslant i \leqslant k} g_k(i).$$

Пример 6.4. Некоторое предприятие планирует поставку продукции в течение 7 месяцев в таких объемах: $d_1 = 90$ шт.; $d_2 = 125$; $d_3 = 140$; $d_4 = 100$; $d_5 = 45$; $d_6 = 60$; $d_7 = 130$. Стоимость хранения 1 единицы продукта в течение месяца составляет 2 руб./месяц. Стоимость наладки (или переналадки) оборудования $A_i = 300$ руб., $i = 1, 2, ..., 7$. Наладка проводится в начале только тех месяцев, когда изготовляется продукция; временем на выполнение заказа мы пренебрегаем. Требуется определить периоды времени, когда производится заказ, а также размер заказа.

Допустим, что на начало первого месяца не было никакого запаса, т. е. $y_1 = 0$. Итак, требуется найти

$$\min z = \min_{\{x_j\}\{y_j\}} \left[300 \sum_{j=1}^{6} \delta_j + 2\sum_{j=2}^{7} y_j\right],$$

где $\delta_j = 1$, если $x_j > 0$; $\delta_j = 0$, если $x_j = 0$, при ограничениях

$$\begin{cases} x_1 - y_2 = d_1 = 90, \\ x_j + y_j - y_{j+1} = d_j, & d_2 = 125, \quad d_3 = 140, \quad d_4 = 100, \\ \ldots\ldots\ldots\ldots & d_5 = 45, \quad d_6 = 60. \\ x_7 + y_7 = d_7 = 130, \end{cases}$$

Таблица 6.11

Периоды k		1	2	3	4	5	6	7
Спрос d_k		90	125	140	100	45	60	130
$g_k(i)$	i 1	300	550*	1100				
	2		660	880				
	3			850*	1050*	1230 *	1590	
	4				1150	1240	1480	
	5					1350	1470*	1990
	6						1530	1790
	7							1770*
$\Lambda_k(0)$		300	550	850	1050	1230	1470	1770
—		(1)	(1, 2)	(1, 2) (3)	(1, 2) (3, 4)	(1, 2) (3, 4, 5)	(1, 2) (3, 4) (5, 6)	(1, 2) (3, 4) (5, 6) (7)

Из соотношений (4.19) и (4.20) следует, что $\Lambda_1(0) = A_1 = 300$.

$$\Lambda_2(0) = \min \begin{Bmatrix} A_2 + \Lambda_1(0) = 300 + 300 = 600 \\ A_1 + 2d_2 = 300 + 2 \cdot 125 = 550 \end{Bmatrix} = 550.$$

Таким образом, только при двух периодах и нулевом запасе в конце периода выгодно иметь одну поставку, удовлетворяющую потребностям обоих периодов. Далее

$$\Lambda_3(0) = \min \begin{Bmatrix} A_3 + \Lambda_2(0) = 850, \\ A_2 + 2 \cdot d_3 + \Lambda_1(0) = 880, \\ A_1 + 2 \cdot d_3 + 2(d_2 + d_3) = 1100 \end{Bmatrix} = 850.$$

Результаты остальных вычислений приведены в табл. 6.11.

В последней строке таблицы в скобках указаны номера тех периодов, для которых спрос удовлетворяется вначале первого из них. Звездочки над $g_k(i)$ указывают на то, что данное значение минимально по i и равно $\Lambda_k(0)$ $(\min\limits_i g_k(i) = \Lambda_k(\xi);\ \xi = 0)$.

Оптимальное решение, согласно табл. 6.11, таково:

$$z^* = \Lambda_7(0) = 1770, \quad x_1^* = 90 + 125 = 215, \quad x_2^* = 0, \quad x_3^* = 140 + 100 = 240,$$

$$x_4^* = 0, \quad x_5^* = 45 + 60 = 105, \quad x_6^* = 0, \quad x_7^* = 130.$$

Такая модель называется *динамической моделью выбора объема партии.*

Детерминированная задача управления запасами с бесконечным числом шагов

Ранее была рассмотрена упрощенная задача управления запасами. Основное упрощающее предположение сводилось к следующему. Заказы на изготовление очередной партии могли поступать только в начале текущего периода, причем временем изготовления партии мы пренебрегали.

Рассмотрим теперь более реальную задачу [45]. Пусть время между сделанным заказом и его выполнением равно τ. Будем считать, что скорость спроса λ постоянна и не может быть потребности в двух единицах продукта одновременно. Таким образом, время между последующими требованиями есть $\Delta t = \frac{1}{\lambda}$.

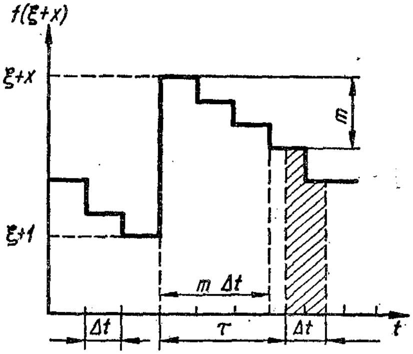


Рис. 6.17.

Пусть эта система может функционировать бесконечно долго и после каждого очередного требования следует решить, делать или не делать заказ. При этом потребуем, чтобы спрос удовлетворялся полностью. Обозначим через A — фиксированные затраты на заказ (затраты на наладку), через c — стоимость изготовления единицы продукта.

Затраты на хранение, как и выше, считаем пропорциональными величине запаса с коэффициентом пропорциональности k. Предполагаем, что величины A, c, k и τ не зависят от времени t. Задача заключается в определении моментов заказов и объемов заказов, которые минимизируют затраты на заказ и его хранение на все будущее время.

Заметим, что поскольку заказ, сделанный в момент t, не может быть выполнен ранее момента $t + \tau$, то и затраты на его хранение не должны включаться в общее выражение затрат на момент t.

Пусть ξ — имеющееся перед принятием решения количество единиц продукта. Через $x \gg 0$ обозначим величину заказа. Тогда $\xi + x$ — величина запаса плюс заказанное количество, после того как заказ был выполнен.

В связи с тем, что все требования выполняются из запаса, а все заказанное количество в момент t выполняется не ранее момента $t + \tau$, то величина запаса в момент $t + \tau$ составит $(\xi + x - m)$ единиц, где $m = \left[\frac{\tau}{\Delta t}\right] = [\lambda\tau]$ (знаком $[\lambda\tau]$ обозначена целая часть $\lambda\tau$).

Следующее требование (спрос) поступит в момент $(m + 1)\,\Delta t$, т. е. спустя $[(m + 1)\,\Delta t - \tau]$ единиц времени после выполнения заказа, который поступил в момент t. Следовательно, в интервале времени между $t + \tau$ и $t + (m + 1)\,\Delta t$ величина запаса равна $\xi + x - m$, а в интервале $t + (m + 1)\,\Delta t$ и $t + \tau + \Delta t$ она составляет $\xi + x - m - 1$ единиц.

Таким образом, затраты на хранение $f(\xi + x)$ в интервале $(t + \tau)$, $(t + \tau + \Delta t)$ выражаются соотношением:

$$f(\xi + x) = k\{(\xi + x - m)[(m + 1)\Delta t - \tau)] + \\ + (\xi + x - m - 1)[\tau - m\Delta t]\} = k(\xi + x - m)\Delta t - k(\tau - m\Delta t). \tag{4.21}$$

Для полного удовлетворения спроса необходимо выполнить условие $\xi + x \geqslant m + 1$. Для этого предположим, что $f(x + \xi) = \infty$ при $\xi + x < m + 1$. На рис. 6.17 представлена динамика изменения запасов.

Заметим, что затраты на хранение в интервале $t + \tau$, $t + \tau + \Delta t$ зависят только от величины $x + \xi$.

Пусть α, как и обычно, коэффициент приведения затрат в момент $t + \Delta t$ к моменту t. Обозначим через $\Lambda_t(\xi)$ суммарную величину всех будущих затрат, приведенных к моменту t, при использовании оптимальной стратегии управления запасами, начиная с момента времени t, на весь последующий (бесконечный) отрезок времени (сюда не включены затраты на хранение за интервал времени до $t + \tau$). Аналогично через $\Lambda_{t+\Delta t}(\xi)$ обозначим приведенную к моменту $t + \Delta t$ величину всех последующих затрат, начиная с этого момента, если принимаются только оптимальные решения.

Пусть y_j — величина запаса в момент $t + j\Delta t$, x_j — размер заказа, $f(x_j + y_j)$ — затраты на хранение в интервале между $t + j\Delta t$ и $t + j\Delta t + \tau$; $y_0 = \xi$; $x_0 = x$. Тогда

$$\Lambda_t(\xi) = \min_{x_1, x_2, \ldots} \sum_{j=0}^{\infty} \alpha^j [A\delta_j + cx_j + f(x_j + y_j)] \tag{4.22}$$

и

$$\Lambda_{t+\Delta t}(\xi) = \min_{x_1, x_2 \ldots} \sum_{j=1}^{\infty} \alpha^{j-1} [A\delta_j + cx_j + f(x_j + y_j)]. \tag{4.23}$$

Следовательно,

$$\Lambda_t(\xi) = \min_x \{A\delta + cx + f(x + \xi) + \\ + \alpha \min_{\{x_1, x_2, \ldots\}} \sum_{j=1}^{\infty} \alpha^{j-1} [A\delta_j + cx_j + f(x_j + y_j)], \tag{4.24}$$

где $x + \xi - 1$ есть величина запаса плюс заказанное количество до принятия решения в момент $t + \Delta t$.

Используя выражение для $\Lambda_{t+\Delta t}(\xi)$, получаем

$$\Lambda_t(\xi) = \min_x [A\delta + cx + f(x + \xi) + \alpha\Lambda_{t+\Delta t}(\xi + x - 1)]. \tag{4.25}$$

С другой стороны, если в (4.22) принять $i = j - 1$, то получим выражение (4.23), т. е.

$$\Lambda_t(\xi) = \Lambda_{t+\Delta t}(\xi) = \cdots = \Lambda_{t+j\Delta t}(\xi), \qquad j = 1, 2, \ldots \tag{4.26}$$

Это означает, что если наличный заказ равен ξ и начиная с момента времени t все последующие управления оптимальны, то величина всех будущих затрат не зависит явно от t. Причина этого в том, что рассматривается бесконечный интервал времени. Подобные процессы называются бесконечношаговыми и поэтому индекс t в выражении для $\Lambda_t(\xi)$ можно опустить.

В результате соотношение (4.25) преобразуется:

$$\Lambda(\xi) = \min_x \{A\delta + cx + f(\xi + x) + \alpha\Lambda(\xi + x - 1)\}. \tag{4.27}$$

Уравнение типа (4.27) называют *функциональным*. Отметим, что уравнение содержит неизвестную функцию $\Lambda(\xi)$ в обеих частях с разными значениями аргумента. Аналитическое решение этих уравнений удается получить в исключительных случаях. Однако для данной задачи оказывается возможным, исследовав (4.27), не только получить информацию о структуре оптимального решения $x_{\text{опт}}$, но и найти его явное решение.

Введем для удобства изложения обозначение

$$F(\xi + x) = c(x + \xi) + f(\xi + x) + \alpha\Lambda(\xi + x - 1). \tag{4.28}$$

Тогда (4.27) можно переписать в виде

$$\Lambda(\xi) = -c\xi + \min_x [A\delta + F(\xi + x) = -c\xi + \min\left\{\begin{array}{l} F(\xi) \\ A + \min\limits_x F(\xi + x). \end{array}\right. \tag{4.29}$$

Пусть абсолютный минимум $F(y)$ достигается при $y = R$, и пусть r $(r < R)$ есть наибольшее целое число, для которого еще выполняется $F(r) \geqslant A + F(R)$. Тогда из соотношения (4.29) следует, что если ξ лежит между $r < \xi < R$, то оптимальная стратегия состоит в следующем: ничего не заказывать, пока сумма ранее заказанного и имеющегося в наличии запаса не достигнет величины r. В этот момент заказывают $(R - r)$ единиц продукта.

Таким образом, при оптимальном управлении наличный запас вместе с заказом всегда будет лежать между r и R. Система управления запасами, обладающая таким свойством, называется *устойчиво управляемой* [45].

Приведенный выше анализ в общем случае не дает метода отыскания неизвестных r и R. Однако в данной задаче величину r можно определить довольно просто. Вспомним, что $f(\xi + x) = \infty$ при $\xi + x < m + 1$. Поэтому, если $r < m$ $(r = \xi + x - 1)$, то $f(r + 1) = \infty$; другими словами, если $r < m$, то к моменту выполнения заказа, сделанного в момент t, одно (или больше) требований окажутся неудовлетворенными.

С другой стороны, если $r > m$, то затраты можно уменьшить, положив $r = m$.

Таким образом, $r = m$ и очередной заказ следует делать сразу после поступления требования, снижающего суммарное количество заказанного и наличного продукта до величины m.

Покажем, как можно вычислить величину R. Если предположить, что числа r и R известны, то функциональное уравнение (4.29) можно решить в явном виде относительно $\Lambda(x)$.

Выпишем соотношения $r \leqslant \xi \leqslant R - 1$:

$$\begin{aligned} &\Lambda(R-1) = f(R-1) + \alpha\Lambda(R-2); \\ &\Lambda(R-2) = f(R-2) + \alpha\Lambda(R-3); \\ &\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ &\Lambda(r) = A + c(R-r) + f(R) + \alpha\Lambda(R-1). \end{aligned} \tag{4.30}$$

Начав с последнего уравнения и подставив его последовательно во все предыдущие, исключим все неизвестные $\Lambda(r)$, $\Lambda(r+1)$, .., $\Lambda(R-2)$. Окончательно получим

$$\Lambda(R-1) = f(R-1) + \alpha f(R-2) + \alpha^2 f(R-3) + \cdots +$$
$$+ \alpha^{R-r-2} f(r+1) + \alpha^{R-r-1}[A + c(R-r) + f(r)] + \alpha^{R-r}\Lambda(R-1),$$

откуда

$$\Lambda(R-1) = \frac{1}{1-\alpha^{R-r}}\left\{\alpha^{R-r-1}[c(R-r)+A] + \sum_{j=1}^{R-r-1} \alpha^{j-1} f(R-j) + \right.$$
$$\left. + \alpha^{R-r-1} f(R)\right\}. \tag{4.31}$$

Выражению (4.31) можно дать интересную интерпретацию. Отметим, что система приходит в прежнее состояние через каждые $(R-r)\,\Delta t$ единиц времени. Поэтому $(R-r)\,\Delta t$ — период цикла. Рассмотрим ситуацию, когда цикл начинается через время Δt после выполнения заказа, т. е. немедленно после того, как поступит требование, которое снижает суммарный запас до $R-1$. Следующий заказ будет сделан через время $(R-r-1)\,\Delta t$ после начала цикла. Поэтому выражение в фигурных скобках правой части (4.31) — затраты на цикл, приведенные к началу цикла, а $\Lambda(R-1)$ — приведенная сумма затрат по всем циклам за бесконечный интервал времени.

Есть несколько способов вычисления R или $R-r$. Например, систему с бесконечным числом шагов можно аппроксимировать системой с конечным числом шагов и по мере их добавления вычислять непосредственно минимум из (4.25). После того, как взято достаточно большое число шагов n, $\hat{x}_n(\xi)$ перестанет изменяться в зависимости от n. Тогда $R-r$ есть величина скачка, на который изменяется $\hat{x}(\xi)$.

Другой способ, позволяющий точно определить $R-r$, состоит в подстановке (4.31) в (4.27) при $\xi = r$ и последующем отыскании минимума (4.29) по $x = R - r$ при $\xi = r$.

Вопросы для самопроверки

1. В чем сущность вычислительного метода динамического программирования?
2. Сформулируйте принцип оптимальности Беллмана и поясните его смысл.
3. Для задач какой структуры возможно применение метода динамического программирования?
4. Укажите условия, при которых задача решается методом динамического программирования в обратном направлении и, наоборот, в прямом направлении.

5. Какой основной недостаток динамического программирования?

6. Назовите методы понижения размерности задач динамического программирования.

7. Укажите главные компоненты задачи управления запасами.

8. Выведите формулы для простейшей задачи управления запасами с постоянным спросом и периодическими поставками, когда период между поставками постоянен.

9. В какой модели управления запасами используется двухуровневая стратегия и как она формулируется?

10. В каких задачах управления запасами используется метод динамического программирования?

ЗАКЛЮЧЕНИЕ

Внедрение результатов решения операционной задачи является заключительным и одним из наиболее ответственных этапов операционного исследования.

На каждом из предыдущих этапов есть возможность построения моделей, принятия научных решений, а также корректировки решений. Но на этапе внедрения такой возможности уже нет. Задачи, возникающие на этом этапе, относятся к области искусства управления.

Возможность внедрения и управления процессом реализации любого решения появляется лишь тогда, когда оно одобрено руководством, имеющим соответствующие полномочия. Причем одобрение это зависит не только от качества выполненного решения и выгод, которые оно сулит, но и от характера и степени участия руководства в самом процессе операционного исследования, а также его отношения к полученным результатам.

Проведенные эксперименты, а также накопленный опыт внедрения результатов решения дали возможность определить некоторые факторы, влияющие на вероятность одобрения руководством решения операционных задач, а также разработать целый комплекс мероприятий, стимулирующих у руководства заинтересованность к внедрению законченных исследований. Эти положения таковы [1].

1. Операционная группа должна заботиться о том, чтобы решение исследуемой задачи оказалось приемлемым для руководства еще до того, как оно получено. Одобрение решения руководством в большой степени определяется его убеждением, что решение задачи действительно отвечает его основным запросам, а также пониманием метода отыскания этого решения. Поэтому участие руководителей в проведении операционного исследования не только желательно, но и необходимо.

В ходе проведения исследования необходимо индивидуально встречаться с каждым заинтересованным руководителем с целью получения консультаций и обсуждения возникающих вопросов.

2. Необходимо организовать контрольный, или консультативный, совет с участием руководителей организации, на котором регулярно заслушивать информацию операционистов о ходе операционного исследования. Систематическая оценка хода выполняемых исследований со стороны руководителей создает у них уверенность в том, что все существенные аспекты проблемы учтены операционистами. Кроме того, это позволяет руководителям постепенно усваивать сущность

применяемых методов и средств, что приводит к гораздо лучшему пониманию существа операционного исследования, чем ознакомление с пухлым отчетом о выполненном комплексе работ.

3. Желательно, чтобы исследование финансировалось теми руководителями, по заказу которых оно выполняется.

На первый взгляд может показаться, что организации выгодно бесплатно пользоваться плодами научных исследований, но это не всегда так.

Только при оплате выполняемых работ из своего собственного бюджета у руководителя организации появляется уверенность, что они служат его собственным интересам. В таком случае руководителя организации гораздо легче привлечь к участию в проведении исследования и внедрению полученных результатов. Поэтому всегда, когда имеется хоть малейшая возможность, руководство организации должно участвовать в расходах, связанных с проведением исследований.

4. По мере роста доверия к операционной группе со стороны руководства проблема, порождаемая чрезмерным доверием, может стать более серьезной, чем проблема недоверия. Несколько положительных результатов могут повлиять на руководство таким образом, что будут одобряться недостаточно проверенные решения.

Если операционистам не хватает времени на проведение тщательного исследования, то они обязаны точно проинформировать руководителей о тех отклонениях от нормальных методов исследования, которые были допущены. Это позволяет руководству сознательно идти на риск, одобряя (при необходимости) результаты, полученные в условиях нехватки времени.

5. Во всех допустимых случаях следует предпочитать устные отчеты о проделанной работе письменным, так как такая форма отчета позволяет использовать преимущества обратной связи, т. е. дает возможность получать вопросы при обсуждении и давать на них ответы.

6. На этапе внедрения обычно возникают задачи, которые невозможно предвидеть в период проведения исследования, вследствие чего требуется некоторая модификация найденного решения. Если изменения в решение вносятся недостаточно компетентными операционистами, которые не принимали участия в его отыскании, то эти изменения могут оказаться неверными или исказить весь смысл найденного решения. Поэтому изменения на этапе внедрения должны вноситься именно теми операционистами, которые проводили операционное исследование.

7. Операционисты должны разработать детальные инструкции для тех, кому предстоит реализовать найденное решение. Они должны составить календарный план-график процесса внедрения.

К разработке плана внедрения необходимо привлечь всех участников его будущего внедрения. Это позволяет избежать включения в план практически неосуществимых заданий и способствует появлению заинтересованности исполнителей в реализации намеченного плана.

Одновременно эта мера придает уверенность руководству высшего уровня, которое отвечает за выполнение плана, в его реализуемости.

Как показывает практика, при разработке сложных планов внедрения часто оказывается полезным создание специальной консультативной группы, которая включает руководителей подразделений, участвующих во внедрении. Чем больше обязанностей возлагается на такую группу, тем большего эффекта можно ожидать.

8. В условиях, когда реализация операционного решения задачи занимает много времени, состояние системы, для которой было получено решение, может существенно измениться. Так, например, может быть сделано неожиданное открытие в науке или технике или изобретение, которое коренным образом изменяет установившиеся понятия, технологию, организацию и т. д. Изменения подобного рода могут в свою очередь повлиять на характер самой задачи, а следовательно, и на эффективность ее решения. Изменения, которые могут возникать в системе организационного управления, относятся к следующим категориям (видам):

а) изменение полезности получаемых результатов, которое влияет на выбор критерия функционирования;

б) изменение набора управляемых переменных;

в) изменение ограничений, наложенных на управляемые переменные;

г) изменение значений параметров. Так, изменение среднего месячного спроса на какой-либо вид продукции может привести к изменению оптимального размера покупаемой партии этой продукции;

д) изменение структуры системы, т. е. изменения соотношений между критерием функционирования и управляемыми переменными и параметрами.

Системы, являющиеся объектом операционных исследований, почти всегда подвергаются существенным изменениям с течением времени, и потому методы внедрения, разработанные даже самым лучшим образом, постепенно теряют свою эффективность. Если в реализуемом методе внедрения не предусмотреть специальных мер коррекции, то внедренное решение может оказаться совершенно неадекватным новому состоянию системы, а потому неэффективным.

Следовательно, в процедурах реализации решения необходимо предусмотреть средства его коррекции и обеспечить условия, чтобы этими средствами пользовались только квалифицированные специалисты.

В заключение отметим некоторые актуальные направления работ в области исследования операций.

1. В области теории первостепенное значение приобретает разработка и совершенствование эффективных методов математического программирования, в частности, линейного, динамического, и дискретного, при решении задач большой размерности. Такие задачи постоянно возникают при оптимизации планирования и управления макроэкономическими системами на уровне отраслей, групп отраслей и экономики страны в целом. Проблема снижения или снятия «проклятия размерности» по-прежнему остается главным препятствием для успешного применения операционных методов при оптимизации управления большими организационными системами.

Для решения этой проблемы необходимо развитие различных методов декомпозиции, методов, использующих иерархические подходы, и эвристического программирования. Новые интересные методы решения крупных макроэкономических задач, возникающих в процессе создания ОГАС, предложены акад. В. М. Глушковым.

2. В области нелинейного программирования перспективным направлением является создание эффективных методов оптимизации при наличии ограничений, разработка специальных алгоритмов глобального поиска экстремума многомерных функций.

3. При разработке проектов сложных систем, в частности автоматических систем управления и АСУ, перед проектировщиком возникают проблемы принятия решений при наличии одновременно нескольких показателей качества (критериев). Поэтому разработка методов принятия решений при нескольких критериях оптимальности (многокритериальная оптимизация) и в условиях неопределенности по-прежнему остается одной из главных задач исследования операций.

В последние годы для принятия решений при проектировании и управлении в условиях неполной информации и наличия плохоформализуемых факторов все шире применяются диалоговые системы (планирования, проектирования и т. д.). В этих системах обеспечиваются возможности для удачного соединения развитых математических методов оптимизации и быстродействующих ЭВМ с опытом и неформализуемыми знаниями проектировщиков. Большой положительный опыт в этом направлении накоплен в Институте кибернетики АН УССР.

4. Для анализа и исследования больших систем организационного управления, функционирующих при воздействии случайных или неопределенных факторов, следует все шире использовать имитационное моделирование и деловые игры. Для успешного применения имитационного моделирования необходимо развитие языков моделирования и программного обеспечения, в частности создание программных комплексов моделирования, ориентированных на широкий класс моделей.

5. Актуальной задачей в области прикладных исследований является совершенствование методов разработки моделей исследования операций, в частности моделей управления запасами, распределения ресурсов, календарного планирования, СПУ, и накопление опыта в их применении для оптимизации управления в организационных системах.

НЕКОТОРЫЕ ВОПРОСЫ ЛИНЕЙНОЙ АЛГЕБРЫ

Линейная зависимость и независимость векторов, линейное пространство и базис

Определение 1.1. *Даны множество m-мерных векторов* $A_1, A_2, \ldots, A_n$ *и скаляры* $k_1, k_2, \ldots, k_n$. *Тогда m-мерный вектор* A_0 *вида* $A_0 = \sum_{i=1}^{n} k_i A_i$ *называется линейной комбинацией векторов* $A_1, A_2, \ldots, A_n$.

Заметим, что скаляры k_i, $i = \overline{1, n}$ здесь произвольны. Варьируя k_i, получим все возможные линейные комбинации.

Определение 1.2. *Система векторов* $A_1, A_2, \ldots, A_n$ *называется линейно-независимой, если равенство вида*

$$\sum_{i=1}^{n} k_i A_i = 0 \tag{1.1}$$

выполняется тогда и только тогда, когда $k_1 = k_2 = \ldots = k_n = 0$. В противном случае система называется линейно-зависимой.

Допустим, что условие (1.1) выполняется при некоторых $k_i \neq 0$. Выбрав такой вектор A_{i_1}, что $k_{i_1} \neq 0$, и перенеся его вправо, получим

$$A_{i_1} = \sum_{\substack{i=1 \\ i \neq i_1}}^{n} \left(-\frac{k_i}{k_{i_1}} \right) A_i = \sum_{\substack{i=1 \\ i \neq i_1}}^{n} k'_i A_i. \tag{1.2}$$

Следовательно, в случае линейно-зависимой системы векторов вектор A_i выражается в виде линейной комбинации остальных векторов системы.

Определение 1.3. Дано множество m-мерных векторов $A_1, \ldots, A_n$. *Множество R всех линейных комбинаций векторов* $\{A_1, A_2, \ldots, A_n\}$ *называется линейной оболочкой, или пространством, натянутым на данное множество векторов (порожденным данным множеством)*:

$$R = \left\{ A : A = \sum_{i=1}^{n} k_i A_i \right\}, \text{ где } k_i \in (-\infty, \infty). \tag{1.3}$$

Рассмотрим некоторые примеры.

Если $n = 2$ и заданы векторы A_1 и A_2, неколлинеарные друг другу, то их линейной оболочкой является плоскость.

Пусть $n = 3$ и векторы A_1, A_2 и A_3 не лежат в одной плоскости. Тогда их линейной оболочкой будет трехмерное пространство. Следовательно, любой вектор A этого пространства можно представить как линейную комбинацию заданных векторов (рис. П. 1). При исследовании некоторого линейного пространства, порожденного системой векторов, важно уметь находить такое порождающее множество векторов, которое содержит минимальное число векторов, необходимое для образования (задания) этого пространства. Такое наименьшее порождающее множество называется *базисом* и обладает многими важными свойствами.

Определение 1.4. *Пусть R есть некоторое пространство, порождаемое системой векторов* $A = \{A_1, A_2, \ldots, A_n\}$. *Тогда базисом этого пространства называется такое подмножество векторов* $A_x = \{A_{i_1}, \ldots, A_{i_m}\} \subseteq A$, *которое обладает следующими свойствами*:

а) *все векторы базиса* — линейно-независимы;

б) *они порождают то же пространство R, что и исходная система векторов* A. При этом размерность пространства m определяется числом векторов базиса.

Отметим некоторые важные свойства базиса.

1. Для любого конечного множества (системы) векторов A всякое его подмножество из максимального числа линейно-независимых векторов является базисом для A

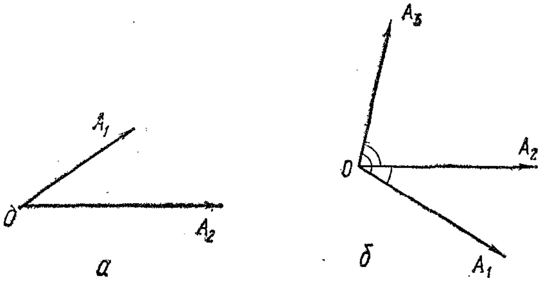


Рис. П.1.

2. Любой вектор A_j множества может быть выражен через базисные, причем единственным образом, т. е.

$$A_j = \sum_{i \in I_б} k_{ij} A_i,$$

где $I_б$ — множество индексов базисных векторов; k_{ij} — коэффициент разложения вектора A_j через базисные. Таким образом, базис пространства при $m = 2$ (т. е. плоскости) образует любая пара пересекающихся векторов A_1, A_2, принадлежащих этому пространству. Базис трехмерного пространства образуют любые три вектора A_1, A_2, A_3, не лежащих в одной плоскости, и т. д.

В общем случае для любого пространства R размерности m любое множество из m линейно-независимых векторов, принадлежащих R, является базисом этого пространства.

Выпуклые множества, крайние точки, выпуклые многогранники и конусы

Определение 1.5. *Пусть дано множество векторов $A_1, A_2, \ldots, A_m$. Тогда их выпуклой комбинацией называется вектор A_0, определяемый соотношениями*

$$A_0 = \sum_{i=1}^{m} A_i k_i, \quad k_i \geqslant 0, \quad i = \overline{1, m}, \quad \sum_{i=1}^{m} k_i = 1. \tag{1.4}$$

Дадим геометрическую интерпретацию выпуклой комбинации. Для этого рассмотрим некоторые частные случаи.

1) $m = 2$. Тогда $A_0 = k_1 A_1 + k_2 A_2$, где $k_1 \geqslant 0$, $k_2 \geqslant 0$ и $k_1 + k_2 = 1$ или

$$A_0 = k_1 A_1 + (1 - k_1) A_2. \tag{1.5}$$

Рассмотрим, что собой представляет множество выпуклых комбинаций вида (1.5), для чего необходимо изменять k_1 от 0 до 1. Очевидно, при $k_1 = 0$ $A_0 = A_2$, при $k_1 = 1$ $A_0 = A_1$, а при $0 < k_1 < 1$ A_0 расположен внутри угла A_1OA_2, и при изменении k от 0 до 1 конец вектора A_0 описывает отрезок A_1A_2 (рис. П.2). Следовательно, множество всех выпуклых комбинаций точек A_1 и A_2 представляет собой отрезок, их соединяющий.

2) $m=3$. Тогда $A_0 = k_1A_1 + k_2A_2 + k_3A_3$, где $k_i \geqslant 0$, $\sum_{i=1}^{3} k_i = 1$.

Очевидно, если один из k_i, например $k_3 = 0$, то этот случай сведется к предыдущему и вектор A_0 будет принадлежать соответствующей грани A_1OA_2 пирамиды, ребрами которой являются A_1, A_2 и A_3 (рис. П.3).

В случае же если все $k_i \neq 0$, то вектор A_0 принадлежит внутренней части пирамиды $OA_1A_2A_3$.

Таким образом, множество выпуклых комбинаций трех векторов A_1, A_2, A_3 представляет собой пирамиду, ребрами которой служат векторы A_1, A_2 и A_3.

Введем теперь определение выпуклого множества точек.

Определение 1.6. *Множество точек (векторов) R в n-мерном пространстве называется выпуклым, если для всех x_i, $x_j \in R$ любая их выпуклая комбинация $kx_i + (1 - k) x_j = x_0$ для всех $0 \leqslant k \leqslant 1$ будет также принадлежать множеству R.*

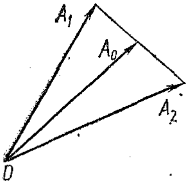


Рис. П.2.

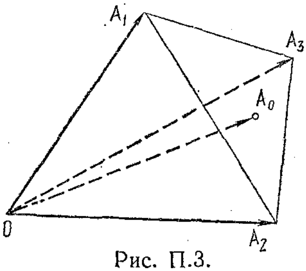


Рис. П.3.

Как было показано выше, выпуклая комбинация пары точек x_i, x_j представляет собой отрезок, их соединяющий. Поэтому можно дать еще одно эквивалентное определение выпуклого множества.

Множество R называется выпуклым, если для любой пары точек (x_i, x_j), принадлежащей множеству, отрезок, их соединяющий, будет также принадлежать этому множеству.

На рис. П.4 приводятся примеры некоторых выпуклых множеств, а на рис. П.5 — невыпуклых.

Среди выпуклых множеств важную роль в линейном программировании играют множества, границы которых представляют собой гиперплоскости.

Определение 1.7. *Множество R (x) называется замкнутым, если вместе с внутренними точками оно содержит и все граничные точки.*

Определение 1.8. *Множество R (x) называется ограниченным тогда и только тогда, когда существует такое конечное число $C > 0$, что для всех $x_i \in R$ (x) $\| x_i \| \leqslant C$.*

Определение 1.9. *Полупространством R (x) называется множество всех векторов x, определяемых условием*

$$R(x) = \left\{ x : A^T x = \sum_{j=1}^{n} a_{1j}x_j \leqslant b \right\}. \tag{1.6}$$

Так как уравнение

$$\sum_{j=1}^{n} a_{1j}x_j = b \tag{1.7}$$

определяет гиперплоскость в n-мерном пространстве, то полупространство представляет собой часть пространства $R^{(n)}$, границей которого служит гиперплоскость, определяемая (1.7). Пример полупространства для случая $n = 2$ (т. е. полуплоскость) приведен на рис. П.6.

Легко показать, что полупространство является выпуклым и замкнутым множеством.

Теперь рассмотрим множество, представляющее собой пересечение конечного числа полупространств.

Определение 1.9. *Множество R_0 (x), определяемое ограничениями вида*

$$R_i(x) : \sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \qquad i = \overline{1, m}, \tag{1.9}$$

называется многогранным выпуклым множеством.

Если это множество ограничено, то оно называется выпуклым многогранником.

Как видно из (1.9), выпуклое многогранное множество представляет собой пересечение конечного числа полупространств, т. е.

$$R_0(\mathbf{x}) = R_1(\mathbf{x}) \cap R_2(\mathbf{x}) \cap \ldots \cap R_m(\mathbf{x}), \qquad (1.10)$$

где $R_i(\mathbf{x})$ — определяется i-м ограничением типа (1.9). Следовательно, выпуклое многогранное множество является замкнутым (на рис. П.7 приведены примеры многогранников и многогранных множеств).

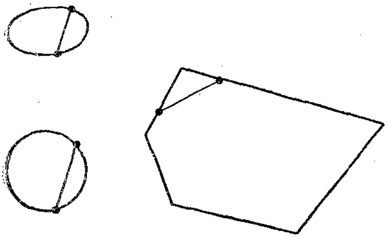


Рис. П.4.

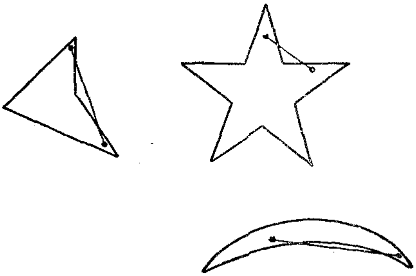


Рис. П.5.

Отметим, что выпуклые многогранные множества и многогранники играют важную роль в линейном программировании и представляют собой допустимые множества решений ЛП-задач.

Пусть $n > m$, выберем из матрицы $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n\}$ ограничений (1.9) некоторый базис, например составленный из векторов $\{\mathbf{A}_{s_1}, \mathbf{A}_{s_2}, \ldots, \mathbf{A}_{s_m}\}$.

Найдем решение системы уравнений вида

$$\sum_{j=1}^{m} a_{is_j} x_{s_j} = b_i. \qquad (1.11)$$

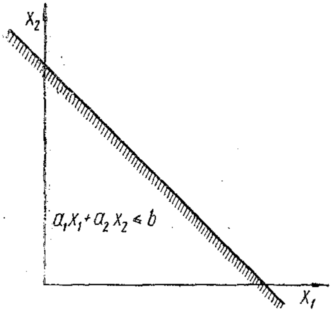


Рис. П.6.

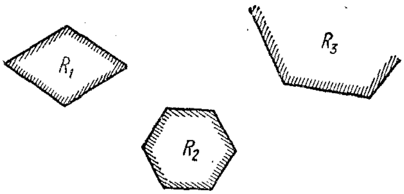


Рис. П.7.

В силу линейной независимости векторов $\{\mathbf{A}_{s_1}, \mathbf{A}_{s_2}, \ldots, \mathbf{A}_{s_m}\}$ это решение единственно. Оно называется *крайней точкой* выпуклого многогранного множества (многогранника).

Очевидно, любая крайняя точка представляет собой пересечение m гиперплоскостей в m-мерном пространстве.

Выпуклые многогранники (многогранные множества) обладают фундаментальным свойством, широко используемым в теории линейного программирования, которое формулируется в следующей лемме

Лемма. *Пусть $R(\mathbf{x})$ — выпуклый многогранник с крайними точками* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$. *Тогда его любая точка может быть представлена как выпуклая комбинация его* *крайних точек, т. е., если* $x_0 \in R(x)$, *то*

$$x_0 = \sum_{i=1}^{m} k_i x_i, \tag{1.12}$$

где $k_i \geqslant 0,\ \sum_{i=1}^{m} k_i = 1.$

Если $R_1(x)$ — выпуклое многогранное множество, то его любая точка может быть представлена в виде суммы выпуклой комбинации крайних точек и линейной комбинации с неотрицательными коэффициентами направляющих векторов неограниченных ребер множества, т. е.

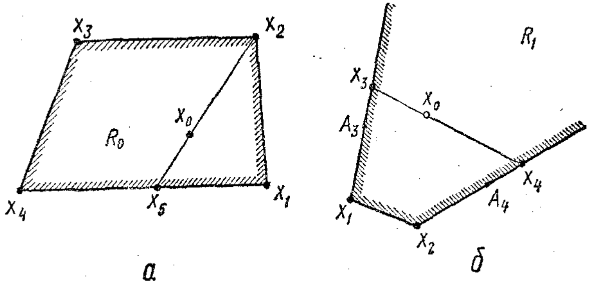


Рис. П.8.

$$x_0 = \sum_{i=1}^{m} k_i x_i, \tag{1.13}$$

где $k_i \geqslant 0,\ \sum_{i=1}^{m} \varepsilon_i k_i = 1,$

$$\varepsilon_i = \begin{cases} 1, \text{ если } x_i \text{ — крайняя точка } R_1(x); \\ 0, \text{ если } x_i \text{ — неограниченное ребро множества } R_1(x). \end{cases}$$

Доказательство леммы приведено в [57]. Покажем справедливость леммы на следующих примерах.

Пусть R_0 — выпуклый многогранник с крайними точками x_1, x_2, x_3, x_4 (рис. П.8, *а*). Выберем произвольную точку x_0 и, соединив с x_2, продолжим отрезок до пересечения с x_1x_4 в точке x_5. Поскольку точка x_5 принадлежит отрезку x_1x_4, то она является выпуклой комбинацией его крайних точек x_1 и x_4 и поэтому

$$x_5 = k_1x_1 + k_4x_4, \text{ где } k_1 + k_4 = 1,\ k_1 \geqslant 0,\ k_4 \geqslant 0. \tag{1.14}$$

Рассматривая точку x_0 отрезка x_2x_5, по той же причине получим

$$x_0 = k_2x_2 + k_5x_5, \quad k_2 \geqslant 0, \quad k_5 \geqslant 0, \quad k_2 + k_5 = 1. \tag{1.15}$$

Подставим (1.14) в (1.15) и получим

$$x_0 = k_2x_2 + k_5(k_1x_1 + k_4x_4) = k_2x_2 + k_5k_1x_1 + k_5k_4x_4,$$

или

$$x_0 = k_1'x_1 + k_2x_2 + k_4'x_4; \quad k_1' = k_5k_1; \quad k_4' = k_5k_4.$$

Так как $k_1' + k_2 + k_4' = k_5(k_1 + k_4) + k_2 = k_5 + k_2 = 1,$ то первая часть леммы справедлива.

Для проверки второй части леммы рассмотрим выпуклое многогранное множество $R_1(x)$, приведенное на рис. П.8, *б*. Здесь A_3 и A_4—направляющие векторы неограниченных ребер, x_1 и x_2 — крайние точки. Выберем точку x_3 на ребре A_3 и соединим x_3

с x_0 и продолжим до пересечения с неограниченным ребром A_4 в точке x_4. Очевидно

$$\begin{aligned} x_3 &= x_1 + k_3A_3; \quad (k_3 > 0), \\ x_4 &= x_2 + k_4A_4; \quad (k_4 > 0). \end{aligned} \tag{1.16}$$

Поскольку точка x_0 лежит на отрезке x_3x_4, то для нее выполняется соотношение:

$$x_0 = k_1x_4 + k_2x_3, \quad k_1 + k_2 = 1. \tag{1.17}$$

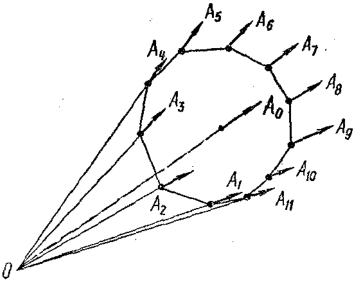


Рис. П.9.

Подставляя (1.16) в (1.17), получим

$$x_0 = k_1x_2 + k_1k_4A_4 + k_2x_1 + k_2k_3A_3 = k_1x_2 + \\ + k_2x_1 + k_3'A_3 + k_4'A_4,$$

где $k_1 + k_2 = 1$; $k_3' > 0$; $k_4' > 0$. Но поскольку точка x_0 была выбрана произвольно, то отсюда вытекает справедливость леммы.

Как мы выяснили, любая точка многогранника представляется как выпуклая комбинация крайних точек $x_1, x_2, \ldots, x_m$. Рассмотрим теперь сами крайние точки. Интересно отметить, что ни одна из них, например x_j, не может быть выражена через другие крайние точки, и в этом случае линейная комбинация вида (1.12) будет вырожденной:

$$k_j = 1, \quad k_i = 0 \text{ для всех } i \neq j.$$

Рассмотрим теперь частный случай ограничений вида (1.9), когда $m = n$.

Определение 1.10. *Множество точек R (x), удовлетворяющих ограничениям*

$$\sum_{j=1}^{m} a_{ij}x_j \leqslant b_i; \quad i = \overline{1, m}$$

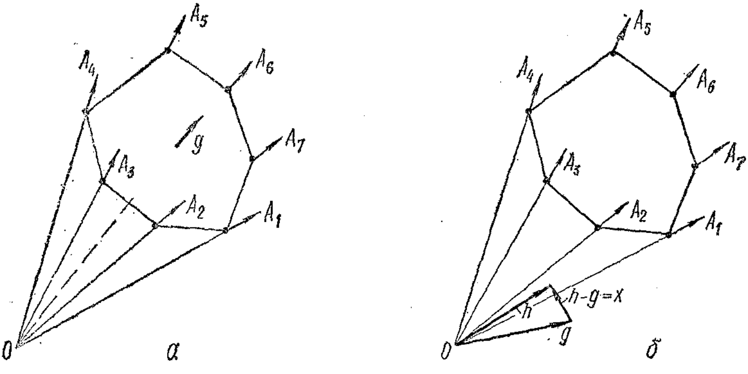


Рис. П.10.

или

$$A^{(m\times m)}x \leqslant b,$$

называется выпуклым многогранным конусом, если существует точка x_0, *такая, что* $Ax_0 = b$. Если матрица A — невырожденная, то эта точка x_0—единственна и называется *вершиной конуса,* а сам конус — невырожденным.

Как видим, выпуклый многогранный конус — это частный случай многогранного множества с единственной крайней точкой.

Ребром выпуклого многогранного множества называется множество, образованное пересечением (m — 1) из m гиперплоскостей, ограничивающих конус.

Существует и другое определение многогранного конуса.

Определение 1.11. Пусть задана система векторов $A_1, A_2, \ldots, A_n$. *Тогда множество всех неотрицательных комбинаций этих векторов вида* $A_0 = \sum_{j=1}^{n} A_j x_j$, $x_j \geqslant 0$, $j = \overline{1, n}$ *называют выпуклым многогранным конусом, натянутым на эти векторы.* Очевидно, векторы $A_1, A_2, \ldots, A_s$, которые не могут быть выражены как неотрицательная комбинация других векторов данной системы, представляют собой ребра этого конуса (рис. П.9).

Лемма Фаркаша. Исключительно важное значение для теории нелинейного программирования имеет следующая лемма Фаркаша, которая используется при доказательстве ряда фундаментальных положений нелинейного программирования [3,50].

Лемма. Пусть заданы векторы A_i ($i = \overline{1, m}$) и вектор g. Неравенства

$$g^T x < 0 \tag{1.18}$$

и

$$A_i^T x \geqslant 0; \quad i = \overline{1, m} \tag{1.19}$$

несовместны тогда и только тогда, когда вектор g *принадлежит выпуклому конусу,* натянутому на векторы A_i, $i = \overline{1, m}$.

Доказательство. Пусть вектор g принадлежит указанному конусу, т. е. его можно представить в виде (рис. П.10, *а*):

$$g = \sum_{i=1}^{m} \lambda_i A_i, \quad \lambda_i \geqslant 0, \quad i = \overline{1, m}. \tag{1.20}$$

Тогда, как легко видеть, неравенства (1.18) и (1.19) будут несовместны при любом x, так как в этом случае из условия (1.19) должно следовать и $g^T x = \sum_{i=1}^{m} \lambda_i A_i^T x \geqslant 0$. Итак, достаточность условий леммы доказана.

Допустим теперь, что представление (1.20) невозможно, и обозначим через h вектор конуса, натянутого на векторы $A_i i = \overline{1, m}$, ближайший к g в эвклидовой метрике (рис. П.10, *б*).

Нетрудно показать, что

$$h^T (h - g) = 0. \tag{1.22}$$

Поэтому, положив

$$x = h - g, \tag{1.23}$$

получим

$$g^T x = g^T (h - g) = -\| h - g \|^2 < 0. \tag{1.24}$$

Кроме того, в силу определения h для любого вектора v, принадлежащего тому же конусу, что и h, при любом θ из отрезка $[0, 1]$ будем иметь

$$\| h - g \|^2 \leqslant \| h - g + \theta (v - h) \|^2. \tag{1.25}$$

Отсюда следует, что

$$(h - g)^T (v - h) \geqslant 0, \tag{1.26}$$

а это с учетом (1.23) и (1.24) эквивалентно неравенству

$$x^T v \geqslant 0. \tag{1.27}$$

В частности, в качестве вектора v *можно выбрать любой из векторов* A_i ($i = 1, m$) и тем самым убедиться, что x удовлетворяет (1.19). Таким образом, допустив невозможность представления в виде (1.20), мы показали одновременно разрешимость системы неравенств (1.18), (1.19), что доказывает вторую часть леммы (т. е. необходимость условий).

Из леммы Фаркаша вытекает одно полезное следствие, используемое при доказательстве теоремы Куна — Таккера.

При любой матрице $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2 \dots \mathbf{A}_m]^T$ и *векторе* $\mathbf{g} = [g_1 g_2 \dots g_n]^T$ — *либо выполняется система неравенств*

$$\begin{cases} \mathbf{A}_i \mathbf{x} \leqslant 0, & (1.28) \\ \mathbf{g}^T \mathbf{x} < 0, & (1.29) \end{cases}$$

либо выполняется следующая система равенств:

$$\mathbf{A}^T \lambda + \lambda_0 \mathbf{g} = 0, \text{ где } \lambda \geqslant 0, \ \lambda_0 > 0. \qquad (1.30)$$
$$\lambda \neq 0.$$

Одновременно обе эти системы выполняться не могут.

Доказательство. Предположим, что система (1.28) — (1.29) несовместна. Это означает, что несовместна также и система

$$\begin{cases} \mathbf{A}_i \mathbf{x} \leqslant 0, \quad i = \overline{1,\ m}, \\ -\mathbf{g}^T \mathbf{x} > 0. \end{cases} \qquad (1.31)$$

Согласно лемме Фаркаша это будет тогда и только тогда, когда вектор $-\mathbf{g}$ принадлежит конусу, натянутому на векторы $\mathbf{A}_i$, $i = \overline{1, m}$. Следовательно, в этом случае $-\mathbf{g} = \sum\limits_{i=1}^{m} \lambda_i \mathbf{A}_i$, $\lambda_i \geqslant 0$. Откуда

$$\sum_i \lambda_i \mathbf{A}_i + \mathbf{g} = 0. \qquad (1.32)$$

Обозначив $\lambda = [\lambda_i]$ $i = \overline{1, m}$ и полагая $\lambda_0 = 1$, придем к соотношению

$$\mathbf{A}^T \lambda + \lambda_0 \mathbf{g} = 0. \qquad (1.33)$$

Первая часть утверждения следствия доказана. Вторая часть доказательства проводится аналогично доказательству второй части леммы Фаркаша.

Приложение 2

Приведение системы линейных уравнений к нормальной форме Смита

Важную роль в построении алгоритмов целочисленного программирования играет так называемая *«нормальная форма Смита»* систем линейных уравнений.

Для изложения этого вопроса нам потребуется ввести ряд матриц специального типа и рассмотреть их свойства.

Матрицы перестановок. Рассмотрим матрицу $\Pi^{(n \times n)}$, элементы которой π_{ij} удовлетворяют следующим условиям:

$$\pi_{ij} = 0 \text{ или } 1; \qquad (i,\ j) = 1,\ 2,\ \dots,\ n; \qquad (1)$$

$$\sum_{i=1}^{n} \pi_{ij} = 1; \qquad j = 1,\ 2,\ \dots,\ n; \qquad (2)$$

$$\sum_{j=1}^{n} \pi_{ij} = 1; \qquad i = 1,\ 2,\ \dots,\ n. \qquad (3)$$

Такая матрица называется матрицей перестановок. Очевидно, она представляет собой квадратную матрицу, в которой каждый столбец и каждая строка содержит только один единичный элемент.

Умножение матрицы $\Pi^{(n\times n)}$ на какую-либо матрицу слева (справа) приводит к перестановке строк (столбцов) этой матрицы.

Ниже приводятся примеры таких умножений:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix} = \begin{bmatrix} a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \qquad (4)$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a_{13} & a_{15} & a_{11} & a_{12} & a_{14} \\ a_{23} & a_{25} & a_{21} & a_{22} & a_{24} \\ a_{33} & a_{35} & a_{31} & a_{32} & a_{34} \\ a_{34} & a_{45} & a_{41} & a_{42} & a_{44} \\ a_{35} & a_{55} & a_{51} & a_{52} & a_{54} \end{bmatrix} \qquad (5)$$

Матрицы транспозиций. Матрица перестановок, в которой два и только два единичных элемента не лежат на главной диагонали, называется матрицей транспозиций. Умножение слева этой матрицы на матрицу **А** приводит к перестановке двух строк матрицы **А**, а умножение справа на **В** — к перестановке двух столбцов матрицы **В**.

Так, например, матрица

$$P_{2,5}^{(5\times 5)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

является матрицей транспозиций. Эта матрица переставляет строку 2 со строкой 5 или столбец 2 со столбцом 5, в зависимости от того, производится ли умножение слева или справа. Матрицу транспозиций, производящую перестановку i-й и j-й строк, будем обозначать через $P_{i,j}^{(n\times n)}$.

Унимодулярные матрицы. Унимодулярной матрицей назовем квадратную матрицу, определитель которой равен $+1$, -1 или 0. Невырожденной унимодулярной матрицей назовем матрицу, определитель которой равен $+1$ или -1.

Произведение двух (или более) унимодулярных матриц является унимодулярной матрицей.

Матрица перестановок является невырожденной унимодулярной матрицей. Если эта матрица получена из диагональной четным числом транспозиций, то ее определитель равен 1, а если нечетным, то -1.

Матрица транспозиций представляет собой невырожденную унимодулярную матрицу и ее определитель равен -1, так как она получена из диагональной нечетным числом транспозиций строк (столбцов).

Матрица вычитаний. Рассмотрим сначала единичную квадратную матрицу, в которой один нуль заменен числом $-\alpha$, где α — любое действительное число. Если $-\alpha$ находится в i-й строке и j-м столбце, то будем обозначать такую матрицу через $U_{i,j,\alpha}$. Например,

$$U_{3,5,\alpha}^{(5,5)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\alpha \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Такую матрицу будем называть элементарной матрицей вычитаний. Очевидно, она является невырожденной унимодулярной матрицей и ее определитель равен 1.

Элементарная матрица вычитаний обладает следующими свойствами. Умножение матрицы **А** на $U_{i,j,\alpha}$ слева приводит к вычитанию из i-й строки **А** ее j-й строки, умноженной на α. Умножение матрицы B справа на $U_{i,j,\alpha}$ приводит к вычитанию из j-го столбца матрицы i-го столбца, умноженного на α.

Следующие примеры иллюстрируют эти свойства:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} - \alpha a_{41} & a_{22} - \alpha a_{42} & a_{23} - \alpha a_{43} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} - \alpha b_{12} \\ b_{21} & b_{22} & b_{23} & b_{24} - \alpha b_{22} \\ b_{31} & b_{32} & b_{33} & b_{34} - \alpha b_{32} \end{bmatrix}$$

Рассмотрим теперь 2 элементарные матрицы вычитаний $U_{i,j,\alpha}$ и $U_{i,k,\beta}$, где $i \neq k$. Их произведение является матрицей, у которой на главной диагонали расположены единичные элементы, а все остальные элементы равны 0, за исключением элемента (i, j), равного $-\alpha$, и (i, k), равного $-\beta$.

Произведение элементарных матриц вычитаний будем называть составными матрицами вычитаний. Например:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha & 1 & 0 & -\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

Очевидно, любая составная матрица вычитаний имеет определитель, равный 1, и является, следовательно, унимодулярной. Умножение составной матрицы вычитаний на произвольную матрицу приводит к соответствующим преобразованиям в этой матрице.

Арифметически эквивалентные матрицы. Две матрицы $A^{(m \times n)}$ и $B^{(m \times n)}$ будем называть арифметически эквивалентными, если существуют две невырожденные унимодулярные матрицы $U^{(m \times n)}$ и $V^{(n \times n)}$такие, что $U^{(m \times m)} A^{(m \times n)} V^{(n \times n)} = B^{(m \times n)}$.

Нормальная форма Смита. Пусть A — матрица с целочисленными компонентами размера $(m \times n)$ ранга $r \leqslant \min (m \times n)$ и D — матрица $(m \times n)$ также ранга r, которая имеет следующую форму, называемую «нормальной формой Смита»:

$$D = \left.\underbrace{\begin{bmatrix} d_1 & 0 & \ldots 0 & 0 \ldots 0 \\ 0 & d_2 & \ldots 0 & 0 \ldots 0 \\ \cdot & \cdot & \cdot\,\cdot\,\cdot\,\cdot & \cdot\,\cdot\,\cdot\,\cdot \\ 0 & 0 & \ldots d_r & 0 \ldots 0 \\ 0 & 0 & \ldots 0 & 0 \ldots 0 \\ \cdot & \cdot & \cdot\,\cdot\,\cdot\,\cdot & \cdot\,\cdot\,\cdot\,\cdot \\ 0 & 0 & \ldots 0 & 0 \ldots 0 \end{bmatrix}}_{}\right\} m,$$

где $d_k \neq 0$, $k = \overline{1, r}$.

Справедлива следующая теорема.

Теорема. *Для любой матрицы* $A^{(m \times n)}$ *существует матрица* $D^{(m \times n)}$, *арифметически эквивалентная ей, т. е. связанная с матрицей соотношением*

$$U^{(m \times m)} A^{(m \times n)} V^{(n \times n)} = D^{(m \times n)}, \tag{7}$$

где U и V — невырожденные унимодулярные матрицы.

Доказательство. Для доказательства используем введенные выше матрицы транспозиций и вычитаний. Определим две матрицы U_1, V_1 так, чтобы

$$U_1AV_1 = \begin{bmatrix} \alpha_1 & 0^{(1\times(n-1))} \\ \cdots & \cdots \\ 0^{(m-1)\times 1} & A_1^{(m-1)\times(n-1)} \end{bmatrix} = D_1. \tag{8}$$

Для этого выполним следующие операции.

1. Соответствующей перестановкой строк и столбцов переводим наименьший по модулю элемент матрицы A в позицию (1,1). Пусть этим элементом оказался $a_{ij} = \lambda_1$. Такую перестановку можно выполнить путем последовательного умножения матрицы A слева на матрицу транспозиций $P_{i1}^{(1)}$, а затем справа на матрицу транспозиций $P_{1j}^{(1)}$. В результате элемент $a_{ij} = \lambda_1$ окажется на позиции (1,1).

2. Затем, используя соответствующие матрицы U и V, заменим каждый элемент первой строки и первого столбца остатками p_{ij}, определяемыми следующим образом:

$$p_{1j} = a_{1j} - \alpha_{1j}\lambda_1; \tag{9}$$

$$p_{i1} = a_{i1} - \alpha_{i1}\lambda_1, \tag{10}$$

где $\alpha_{1j} = \left[\dfrac{a_{1j}}{\lambda_1}\right]$; $\alpha_{i1} = \left[\dfrac{a_{i1}}{\lambda_1}\right]$, знаком $[x]$ — обозначена целая часть x.

Этого можно добиться умножением матрицы A справа на матрицы $[V_{1j,\alpha_{1j}}]$ для j таких, что $a_{1j} \neq 0$, а слева — на матрицы $[U_{i1,\alpha_{i1}}]$ для i таких, что $a_{i1} \neq 0$. Если все остатки p_{1j} и p_{i1} равны 0, то мы пришли к матрице D_1. В противном случае к полученной матрице вновь применим преобразования, аналогичные (9) и (10), выбирая в качестве λ_1' минимальный по абсолютной величине ненулевой элемент первой строки и первого столбца. Так как на каждом шаге выбираемый элемент уменьшается по абсолютной величине, то за конечное число шагов мы обязательно придем к матрице D_1 вида (8).

3. Если $A_1^{(m-1)\times(n-1)} = 0$, то на этом конец и $U = U_1$, а $V = V_1$ — искомые матрицы приведения.

Если же $A_1^{(m-1)\times(m-1)} \neq 0$, то над матрицей A_1 выполняют те же операции 1 и 2 для получения формы D_2:

$$D_2 = U_2D_1V_2 = U_2U_1AV_1V_2 = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & A_2^{(m-2)\times(n-2)} \end{bmatrix} \tag{11}$$

Таким образом, получаем последовательность матриц A_1, A_2, ..., размерности которых постепенно уменьшаются, по крайней мере на 1. В результате мы придем к такой матрице A_k, что $A_k = 0$. Теорема доказана.

Матрица $D = UAV$ имеет по построению ранг k . Кроме того, так как она получена путем умножения матрицы A на невырожденные унимодулярные матрицы, то она имеет тот же ранг, что и матрица A, т. е. $k = r$.

Пример. Рассмотрим матрицу $A^{3\times 4} = \begin{bmatrix} 1 & 1 & 1 & -4 \\ 1 & 1 & 1 & -3 \\ 2 & 0 & 2 & -1 \end{bmatrix}$

и приведем ее к нормальной форме Смита.

Первая итерация. Минимальным по абсолютной величине ненулевым элементом является 1 в позиции (1,1). Следовательно, этап 1 преобразования выполнять не нужно.

Для реализации этапа 2 построим матрицы вычитаний U_1 и V_1. Матрицу U_1 необходимо определить так, чтобы в первом столбце элементы на позициях (1,2) и (1,3) оказались нулевыми. Для этого из второй строки необходимо вычесть первую, а из третьей — удвоенную первую. Следовательно, матрица вычитаний U_1 должна иметь вид

$$U = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}.$$

Найдем произведение U_1A

$$U_1A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & -4 \\ 1 & 1 & 1 & 3 \\ 2 & 0 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -4 \\ 0 & 0 & 0 & 7 \\ 0 & -2 & 0 & 7 \end{bmatrix}.$$

Теперь необходимо построить матрицу вычитаний V_1 таким образом, чтобы все элементы первой строки, кроме диагонального, стали равными нулю. Для этого необходимо из второго и третьего столбца вычесть первый, а из четвертого вычесть первый столбец, умноженный на 4.

Таким образом, первая строка матрицы V_1 должна иметь вид [1 —1 —1 4], а остальные строки образуют единичную матрицу. Тогда

$$U_1AV_1 = \begin{bmatrix} 1 & 1 & 1 & -4 \\ 0 & 0 & 0 & 7 \\ 0 & 2 & 0 & 7 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & -1 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7 \\ 0 & -2 & 0 & -7 \end{bmatrix}.$$

Итак, мы получили $D_1 = U_1AV_1$, имеющую вид

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 \\ 0 & -2 & 0 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & A_1 \end{bmatrix}. \tag{12}$$

Так как $A_1 \neq 0$, то необходимо выполнить аналогичные преобразования над матрицей A_1.

Вторая итерация. Первый шаг. 1. Находим в матрице A_1 отличный от нуля, минимальный по модулю элемент. Им оказался элемент (3, 2), равный — 2. Его необходимо перевести на главную диагональ (на поз. (2, 2), для чего строки 3 и 2 необходимо поменять местами. С этой целью матрицу D_1 необходимо слева умножить на матрицу перестановок P_{23}. В результате получим

$$P_{23}D_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 \\ 0 & -2 & 0 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 7 \\ 0 & 0 & 0 & 7 \end{bmatrix}. \tag{13}$$

2. Теперь матрицу $P_{23}D_1$ необходимо преобразовать таким образом, чтобы во всех позициях второй строки и второго столбца, кроме диагональной, стояли нули. Заметим, что второй столбец уже приведен к требуемому виду. Для того, чтобы привести вторую строку к такому же виду, необходимо произвести умножение справа на матрицу вычитаний V_2. Частное от деления 7 на —2 равно —3. Поэтому вторая строка V_2 имеет вид [0 1 0 3]. Умножая $P_{23}D_1$ справа на V_2, получим

$$P_{23}D_1V_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 7 \\ 0 & 0 & 0 & 7 \end{bmatrix} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{V_2} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 \\ 0 & 0 & 0 & 7 \end{bmatrix}}_{A_1} = D_1'.$$

3. Так как в матрице $P_{23}D_1V_2$ во второй строке еще не все недиагональные элементы равны нулю, то необходимо повторить этапы 1—2. Переходим ко второму шагу.

Второй шаг. 1. Находим минимальный элемент во второй строке, он находится в поз. (2,4). Чтобы привести его в позицию (2,2), необходимо поменять местами столбцы второй и четвертый. Для этого умножаем справа матрицу D_1' на матрицу P_{24} и получим

$$P_{23}D_1V_2P_{24} = D_1'' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 \\ 0 & 0 & 0 & 7 \end{bmatrix} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{P_{24}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 7 & 0 & 0 \end{bmatrix}.$$

2. Преобразуем вторую строку и второй столбец к требуемому виду, чтобы все их недиагональные элементы были равны 0. Для этого, умножив слева матрицу D_1'' на U_2, а справа на матрицу вычитаний V_3, получим:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -7 & 1 \end{bmatrix}}_{U_2} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 \\ 0 & 7 & 0 & 0 \end{bmatrix}}_{D_1''} \times \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{V_3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 14 \end{bmatrix} = D_2.$$

Наконец, необходимо перевести элемент 14 на главную диагональ, для чего требуется поменять местами 3-й и 4-й столбцы. С этой целью умножим матрицу D_2 справа на матрицу перестановок P_{34} и получим

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 14 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 14 & 0 \end{bmatrix} = D_2' = D.$$

Как видим, матрица A приведена к нормальной форме Смита. Выпишем теперь все унимодулярные матрицы, которыми мы пользовались в процессе преобразований:

$$D = D_2P_{34} = U_2D_1''V_3P_{34} = U_2P_{23}D_1V_2P_{24}V_3P_{34} = U_2P_{23}U_1AV_1V_2P_{24}V_3P_{34}. \qquad (12)$$

Таким образом, искомые матрицы преобразований равны $U = U_2P_{23}U_1$; $V = V_1V_2P_{24}V_3P_{34}$, их нетрудно найти путем последовательного перемножения найденных составляющих матриц.

Приведенная форма Смита целочисленной матрицы

Нормальная форма Смита D, получаемая преобразованиями, указанными выше, не является единственной, она зависит от используемых матриц U и V и от выбора минимального по абсолютной величине ненулевого элемента матрицы A. В случае, когда в матрице имеется несколько минимальных ненулевых элементов, этот выбор осуществляется произвольным образом и потому эта матрица D будет иметь различный вид. Однако существует некоторая нормальная форма Смита специального вида, которая оказывается единственной для всякой матрицы A.

Теорема. *Для любой матрицы* $A^{(m\times m)}$, *элементами которой являются целые числа и ранг равен* r, *существуют две невырожденные унимодулярные матрицы* $U^{(m\times m)}$ *и* $V^{(m\times m)}$ *с целочисленными элементами, которые дают приведенную форму Смита*

$$UAV = \Delta, \qquad (13)$$

где

$$\Delta = \left[\begin{array}{ccccccc} \delta_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \delta_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \delta_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \multicolumn{4}{c}{\underbrace{\qquad\qquad\qquad\qquad\qquad}_{r}} & \multicolumn{3}{c}{\underbrace{\qquad\qquad\qquad\qquad}_{n-r}} \end{array}\right] \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} r \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} m-r \\ \\ \end{array} \tag{14}$$

и δ_i является делителем δ_{i+1}, $i = 1, 2, \dots, r - 1$. Матрица Δ называется приведенной формой Смита матрицы A. Это матрица единственная. Числа δ_i, $i = \overline{1, r}$ называют элементарными делителями матрицы A.

Доказательство. Предположим, что получена нормальная форма Смита $D^{(m\times n)}$. Покажем, как получить из нее приведенную форму $\Delta^{(m\times n)}$.

Пусть d_i, $i = 1, 2, \dots, r$, — ненулевые элементы матрицы D. Рассмотрим пару (d_k, d_l) из множества пар чисел d_l. Предположим, что $k < l$. Тогда $| d_k | \leqslant | d_l |$.

Обозначим через НОД (d_k, d_l) — наибольший общий делитель для значений $| d_k |$, $| d_l |$; НОК (d_k, d_l) — наименьшее общее кратное величин $| d_k |$, $| d_l |$. Воспользуемся известной теоремой Безу из арифметики [32]. Пусть a и b — два целых числа. Тогда всегда существуют два целых числа λ и μ, таких, что

$$\lambda a + \mu b = \text{НОД}\,(a,\ b). \tag{15}$$

Например, если $a = 35$, $b = 14$, то НОД $(35,14) = 7$, и тогда $1 \cdot 35 + (-2) \cdot 14 = 7$.

Опираясь на эту теорему, можно построить две матрицы $G^{(m\times m)}$ и $H^{(n\times n)}$, такие, что

$$G^{(m\times m)}D^{(m\times n)}H^{(n\times n)} = \Delta^{(m\times n)}, \tag{16}$$

где Δ — является приведенной формой Смита.

Рассмотрим пару (d_k, d_l) и построим невырожденную унимодулярную матрицу $G^{(m\times m)}$, которая задается следующим образом:

$$g_{kk} = 1; \quad g_{kl} = 1; \tag{17}$$

$$g_{lk} = -\frac{\mu d_l}{\text{НОД}\,(d_k,\ d_l)}; \quad g_{ll} = \frac{\lambda d_k}{\text{НОД}\,(d_k,\ d_l)},$$

где λ и μ — числа Безу, определяемые в соответствии с (15),

$$g_{ij} = 0,\ i \neq j,\quad i \neq k,\quad j \neq l,\quad i,\ j = 1,\ 2,\ \dots,\ m,$$

$$g_{ii} = 1,\ i \neq k,\quad i \neq l,\quad i = 1,\ 2,\ \dots,\ m.$$

Заметим, что

$$\begin{bmatrix} g_{kk} & g_{kl} \\ g_{lk} & g_{ll} \end{bmatrix} = g_{kk}g_{ll} - g_{kl}g_{lk} = \frac{\lambda d_k + \mu d_l}{\text{НОД}\,(d_k,\ d_l)} = 1.$$

Следовательно, матрица G является унимодулярной.

Аналогичным образом построим невырожденную унимодулярную матрицу $H^{(n\times n)}$, для которой

$$h_{kk} = \lambda; \quad h_{kl} = -\frac{d_l}{\text{НОД}\,(d_k,\ d_l)}; \quad h_{lk} = \mu; \quad h_{ll} = \frac{d_k}{\text{НОД}\,(d_k,\ d_l)}, \tag{18}$$

тогда как $h_{ij} = 0$, $i \neq j$, $i \neq k$, $j \neq l$, $i, j = 1, 2, \dots, n$, $h_{ii} = 1$, $i \neq k$, $i \neq l$, $i, j = 1, 2, \dots, n$.

Используя (15), получим, что

$$\begin{bmatrix} h_{kk} & h_{kl} \\ h_{lk} & h_{ll} \end{bmatrix} = h_{kk}h_{ll} - h_{lk}h_{kl} = \frac{\lambda d_k + \mu d_l}{\text{НОД}\,(d_k,\ d_l)}. \tag{19}$$

Таким образом, матрица **H** также является невырожденной унимодулярной матрицей с определителем, равным 1.

Рассмотрим теперь матрицы, образованные строками и столбцами, в которых имеются элементы d_k и d_l, для них можно записать:

$$\begin{bmatrix} g_{kk} & g_{kl} \\ g_{lk} & g_{ll} \end{bmatrix} \times \begin{bmatrix} d_k & 0 \\ 0 & d_l \end{bmatrix} \times \begin{bmatrix} h_{kk} & h_{kl} \\ h_{lk} & h_{ll} \end{bmatrix} =$$

$$= \begin{bmatrix} g_{kk}h_{kk}d_k + g_{kl}h_{lk}d_l & g_{kk}h_{kl}d_k + g_{kl}h_{ll}d_l \\ g_{lk}h_{kk}d_k + g_{ll}h_{lk}d_l & g_{lk}h_{kl}d_k + g_{ll}h_{ll}d_l \end{bmatrix}. \tag{20}$$

Подставляя (17) и (18) в (20), получим

$$\begin{bmatrix} \lambda d_k + \mu d_l & 0 \\ 0 & \dfrac{\lambda d_k^2 d_l + \mu d_k d_l^2}{[\text{НОД}\,(d_k,\ d_l)]^2} \end{bmatrix}. \tag{21}$$

Элемент в позиции (1,1) матрицы (21) есть НОД чисел d_k и d_l. Рассмотрим теперь элемент в позиции (2,2).

По хорошо известной теореме арифметики

$$\text{НОД}\,(a,\ b) \times \text{НОК}\,(a,\ b) = ab. \tag{22}$$

Следовательно, можно записать

$$\frac{\lambda d_k^2 d_l + \mu d_k d_l^2}{[\text{НОД}\,(d_k,\ d_l)]^2} = \frac{\lambda d_k + \mu d_l}{\text{НОД}\,(d_k,\ d_l)} \frac{d_k d_l}{\text{НОД}\,(d_k,\ d_l)} = \text{НОК}\,(d_k,\ d_l).$$

Следовательно матрица (21) запишется в виде

$$\begin{bmatrix} g_{kk} & g_{kl} \\ g_{lk} & g_{ll} \end{bmatrix} \times \begin{bmatrix} d_k & 0 \\ 0 & d_l \end{bmatrix} \times \begin{bmatrix} h_{kk} & h_{kl} \\ h_{lk} & h_{ll} \end{bmatrix} = \begin{bmatrix} \text{НОД}\,(d_k,\ d_l) & 0 \\ 0 & \text{НОК}\,(d_k,\ d_l) \end{bmatrix}.$$

Таким образом, преобразование, определяемое умножением слева на матрицу **G**, а справа на матрицу **H**, элементы которых задаются соотношениями (17) и (18), приводит к замене в матрице **D** элемента d_k на δ_k, а d_l на δ_l, где

$$\delta_k = \text{НОД}\,(d_k,\ d_l); \quad \delta_l = \text{НОК}\,(d_k,\ d_l). \tag{23}$$

С помощью этого преобразования элементы $d_1,\ d_2,\ \ldots,\ d_r$ в матрице **D** можно постепенно заменить элементами $\delta_1,\ \delta_2,\ \ldots,\ \delta_r$, для которых будет выполняться условие: δ_i является делителем δ_{i+1} для всех $i = \overline{1,\ r-1}$.

Замечания.

1. Так как оба определителя матриц **G** и **H** равны 1, то исходя из (16), получим, что

$$\det \mathbf{\Delta} = \det \mathbf{D}.$$

2. Из соотношения (23) следует, что если все элементы d_i $i = 1,\ 2,\ \ldots,\ r$ матрицы **D** взаимно простые, то метод получения приведенной формы Смита дает

$$\delta_1 = \delta_2 = \cdots = \delta_{r-1} = 1, \quad \delta_r = \det \mathbf{D}.$$

Элементарные понятия теории групп

Для исследования и анализа алгоритмов целочисленного программирования нам потребуется ряд элементарных понятий теорий групп, которые вводятся ниже.

1. Пусть имеется конечное множество E, на котором введена операция $*$, которая каждой паре элементов $(x, y) \in E$ ставит в соответствие $z \in E$. Эту операцию называют внутренней.

Определение 1. *Внутренняя операция называется замкнутой, если каждой паре* $(x, y) \in E \times E$ *ставится один и только один элемент* $z \in E$, т. е. $x * y = z$.

Определение 2. *Множество E с определенной на нем внутренней операцией $*$ имеет единицу, если в нем существует такой элемент e, что*

$$\forall\ a \in E : e * a = a * e = a.$$

Определение 3. *Внутренняя операция $*$ на множестве E ассоциативна, если*

$$\forall\ a,\ b,\ c \in E, \quad a * (b * c) = (a * b) * c.$$

Определение 4. *Если на множестве E с внутренней операцией $*$ имеется единица e и для каждого $a \in E$ существует один и только один элемент $b \in E$, такой, что*

$$a * b = b * a = e,$$

то элемент b называют обратным к a. Его обозначают через a^{-1}.

Определение 5. *Внутренняя операция $*$, определенная на множестве E, называется коммутативной, или абелевой, если*

$$\forall\ (a, b) \in E \times E : a * b = b * a.$$

Определение 6. *Множество E, на котором операция $*$ задана всюду, называется группоидом.*

Определение 7. *Группоид, в котором имеется единица, называют модулем.*

Определение 8. *Модуль, в котором внутренняя операция $*$ ассоциативна и для каждого его элемента содержится обратный к нему, называется группой.*

Определение 9. *Группа, удовлетворяющая свойству коммутативности, называется абелевой.*

Таким образом группа — это множество (конечное или бесконечное) с определенной на нем внутренней операцией $*$, обладающее свойствами замкнутости, ассоциативности, наличием единицы и обратного элемента для каждого элемента множества.

Среди модулей и групп мы будем рассматривать те, которые связаны с множествами действительных чисел и внутренними операциями$*$, представляющими собой сложение по модулю n.

Рассмотрим множество относительных целых чисел

$$Z = \{, \ldots, -3, -2, -1, 0, 1, 2, \ldots\}.$$

Пусть

$$n \in N_0;\ r \in N;\ a,\ b,\ q,\ q' \in Z,$$

где $N_0 = [0, 1, \ldots]$; $N = [1, 2, \ldots]$.

Два числа называют сравнимыми, или «конгруентными» по модулю n, если их разность при делении на n равна нулю (или же если каждое из них при делении на n дает один и тот же остаток).

Пусть $a = nq + r$, $b = nq' + r$, тогда $a - b = n\ (q - q')$.

Конгруентность чисел a и b будем записывать так:

$$a \simeq b\ (\mathrm{mod}\ n). \qquad (1)$$

Все числа вида $a + kn$, где a, $k \in Z$, $n \in N_0$, относятся к одному и тому же классу, называемому классом вычетов по модулю n.

Существует, таким образом, n классов множеств частных типа Z/R, для которых рассматриваемое отношение — сравнение по модулю n обозначается через Z/n

Таким образом

$$a_0 = 0 \pmod n$$
$$a_1 = 1 \pmod n$$
$$a_2 = 2 \pmod n$$
$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot$$
$$a_{n-1} = n - 1 \pmod n$$

Множество частных Z/n содержит n элементов, каждый из которых представляет собой множество всех элементов, входящих в соответствующий класс эквивалентности.

В качестве примера рассмотрим случай $n = 7$, приведенный в следующей таблице

Таблица 1

a k	0	1	2	3	4	5	6
. . . . .	. . . .	. . . .	. . . .	. . . .	. . . .	. . . .	. . . .
$k = -2$	-14	-13	-12	-11	-10	-9	-8
$k = -1$	-7	-6	-5	-4	-3	-2	-1
$k = 0$	0	1	2	3	4	5	6
$k = 1$	7	8	9	10	11	12	13
$k = 2$	14	15	16	17	18	19	20
. . . . .	. . . .	. . . .	. . . .	. . . .	. . . .	. . . .	. . . .

Каждый столбец таблицы представляет собой соответствующий класс эквивалентности (множество конгруентных чисел по модулю n), его обозначение, которое приводится в заголовке соответствующего столбца, является произвольным. Классы эквивалентности с определенной на них операцией сложения по модулю n образуют коммутативную группу. В качестве примера рассмотрим группы для $n = 1, 2, 3, 4$. Представителя каждого класса будем обозначать числом, заключенным в фигурные скобки.

Таблица 2

x_j x_i	$\{0\}$
$\{0\}$	$\{0\}$

Таблица 3

x_j x_i	$\{0\}$	$\{1\}$
$\{0\}$	$\{0\}$	$\{1\}$
$\{1\}$	$\{1\}$	$\{0\}$

Таблица 4

x_j x_i	{0}	{1}	{2}
{0}	{0}	{1}	{2}
{1}	{1}	{2}	{0}
{2}	{2}	{0}	{1}

Таблица 5

x_j x_i	{0}	{1}	{2}	{3}
{0}	{0}	{1}	{2}	{3}
{1}	{1}	{2}	{3}	{0}
{2}	{2}	{3}	{0}	{1}
{3}	{3}	{0}	{1}	{2}

Циклическая группа. Циклической группой порядка n называется группа (E' $*$), в которой все элементы $x \in E$ могут быть получены следующим образом:

$$\begin{aligned}
&x_1 = 0\\
&x_2 = a * x_1 = a\\
&x_3 = a * x_2 = a * a\\
&\cdots\cdots\cdots\cdots\cdots\cdots\\
&x_{n-1} = a * x_{n-2} = \underbrace{a * a * \cdots * a}_{n-1}\\
&x_n = a * x_{n-1} = \underbrace{a * a * \cdots * a}_{n} \qquad (2)\\
&x_{n+1} = a * x_n = a
\end{aligned}$$

Число a называется порождающим элементом группы, 0 — является ее единицей.

Таблица 6

x_j x_i	0	1/6	2/6	3/6	4/6	5/6
0	0	1/6	2/6	3/6	4/6	5/6
1/6	1/6	2/6	3/6	4/6	5/6	0
2/6	2/6	3/6	4/6	5/6	0	1/6
3/6	3/6	4/6	5/6	0	1/6	2/6
4/6	4/6	5/6	0	1/6	2/6	3/6
5/6	5/6	0	1/6	2/6	3/6	4/6

Пример. Рассмотрим пример циклической группы, в которой порождающий элемент $a = \frac{1}{6}$. Операция $*$ есть сложение по модулю 1. Соответствующие результаты приводятся в табл. 6, ее произвольный элемент (i, j) определяется из условия:

$$x_{ij} = x_i * x_j = (x_i + x_j) \bmod n.$$

Все циклические группы являются коммутативными, или абелевыми, поскольку для них всегда выполняются равенства

$$x_i + x_j = x_j + x_i \pmod n.$$

Полученные результаты для классов эквивалентности по модулю n на множестве целых чисел Z легко распространяются на множество вещественных чисел R. Два числа a и b $(a,\ b \in R)$ сравнимы по модулю n, т. е. $a \simeq b \pmod n$, если a и b дают при делении на n один и тот же остаток. Так, например,

$$-2{,}35 \simeq 0{,}35 \simeq 1{,}65 \simeq 3{,}65 \ldots \pmod 2$$
$$-1{,}42 \simeq -0{,}42 \simeq 0{,}58 \simeq 1{,}58 \ldots \pmod 1.$$

Приложение 4

Теорема об отделимости выпуклого множества (конечно-мерная теорема Хана-Банаха). *Пусть A — непустое выпуклое множество в $R^{(n)}$, не содержащее начала. Тогда существует гиперплоскость, определяемая уравнением $\mathbf{c}^T\mathbf{x} = 0$, такая, что при $\mathbf{x} \in A$ $\mathbf{c}^T\mathbf{x} \geqslant 0$.*

Если при этом множество A — замкнуто, то найдется такое $\varepsilon > 0$, что $\mathbf{c}^T\mathbf{x} \geqslant \varepsilon$ при всех $x \in A$.

Данная теорема утверждает, что любое выпуклое множество A может быть отделено некоторой гиперплоскостью от остальной части пространства $R^{(n)}$.

Важную роль в теории выпуклого программирования играет следующая теорема Фана.

Теорема. *Пусть B — непустое выпуклое множество в $R^{(n)}$, $f_i(\mathbf{x})$, $i = \overline{1, m}$* — выпуклые функции в $R^{(n)}$. Если система уравнений

$$f_i(\mathbf{x}) \leqslant 0; \quad i = \overline{1, m}, \tag{4.1}$$

$$h_j(\mathbf{x}) = 0; \quad j = \overline{1, k} \tag{4.2}$$

не имеет решений на множестве B, то существуют такие неотрицательные $\lambda_i \geqslant 0$, $i = \overline{1, m}$ и μ_j, $j = \overline{1, k}$, что

$$\sum_i \lambda_i f_i(\mathbf{x}) + \sum_j \mu_j h_j(\mathbf{x}) \geqslant 0 \text{ при всех } \mathbf{x} \in B, \tag{4.3}$$

причем все λ_i и μ_j не обращаются в нуль одновременно.

Доказательство. Введем следующее множество $V(\mathbf{x})$:

$$V(\mathbf{x}) = \{(\mathbf{y}, \mathbf{z}) : f_i(\mathbf{x}) \leqslant y_i, \quad i = \overline{1, m}, \quad h_j(\mathbf{x}) = z_j, \quad j = \overline{1, k}\}.$$

Это множество не пусто, причем, по предположению, множество $V(\mathbf{x})$ не содержит начала (так как система (4.1)—(4.2) — несовместна). Покажем, что множество $V = \bigcup\limits_x V(\mathbf{x})$ — выпукло.

Действительно, пусть $(\mathbf{y}_1, \mathbf{z}_1) \in V(\mathbf{x}_1)$ и $(\mathbf{y}_2, \mathbf{z}_2) \in V(\mathbf{x}_2)$. Тогда

$$(\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2) \geqslant \lambda \mathbf{f}(\mathbf{x}_1) + (1 - \lambda)\mathbf{f}(\mathbf{x}_2) \geqslant \mathbf{f}(\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2),$$
$$0 < \lambda < 1.$$

Кроме того,

$$\lambda \mathbf{z}_1 + (1 - \lambda)\, \mathbf{z}_2 = \lambda \mathbf{h}(\mathbf{x}_1) + (1 - \lambda)\, \mathbf{h}(\mathbf{x}_2) = \mathbf{h}(\lambda \mathbf{x}_1 + (1 - \lambda)\, \mathbf{x}_2),$$

где $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ \ldots,\ f_m(\mathbf{x})]^T$,

$\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}),\ h_2(\mathbf{x}),\ \ldots,\ h_k(\mathbf{x})]^T$.

Следовательно,

$$\lambda (\mathbf{y}_1,\ \mathbf{z}_1) + (1 - \lambda)(\mathbf{y}_2,\ \mathbf{z}_2) \in V(\lambda \mathbf{x}_1 + (1 - \lambda)\, \mathbf{x}_2) \quad \text{при } 0 < \lambda < 1, \tag{4.4}$$

и множество V *— выпукло.*

В силу теоремы отделимости существуют такие λ и μ, не равные нулю, одновременно, что

$$\lambda^T \mathbf{y} + \mu^T \mathbf{z} \geqslant 0 \text{ при всех } (\mathbf{y},\ \mathbf{z}) \in V. \tag{4.5}$$

Но так как y_i, $i = \overline{1, m}$, могут быть выбраны произвольно большими, то для выполнения (4.5) должно быть $\lambda_i \geqslant 0$.

Пусть ε — произвольный положительный скаляр, и пусть

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \varepsilon \cdot \bar{1},\ \mathbf{z} = \mathbf{h}(\mathbf{x}) \text{ при } \mathbf{x} \in B, \tag{4.6}$$

где $\bar{1} = [1,\ 1, \ldots, 1]^T$. Тогда $(\mathbf{y},\ \mathbf{z}) \in V(\mathbf{x})$ и, подставляя (4.6) в (4.5), получим

$$\lambda^T \mathbf{f}(\mathbf{x}) + \mu^T \mathbf{h}(\mathbf{x}) + \varepsilon \lambda^T \bar{1} \geqslant 0 \text{ при } \mathbf{x} \in B. \tag{4.7}$$

Так как ε произвольно, то устремляя его к нулю, получим

$$\lambda^T \mathbf{f}(\mathbf{x}) + \mu^T \mathbf{h}(\mathbf{x}) \geqslant 0 \quad \text{для всех } \mathbf{x} \in B. \tag{4.8}$$

Теорема доказана.

Следствие. Пусть B и $f(\mathbf{x})$ определяются условиями теоремы Фана. Пусть $g_j(\mathbf{x})$, $j = \overline{1, l}$,— выпуклые функции на множестве B. Если система $\mathbf{f}(\mathbf{x}) < 0$, $\mathbf{g}(\mathbf{x}) \leqslant 0$, $\mathbf{h}(\mathbf{x}) = 0$ не имеет решения при $\mathbf{x} \in B$, где $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}),\ \ldots,\ g_l(\mathbf{x})]^T$, то существуют такие векторы $\lambda \geqslant 0$, $\mathbf{v} \geqslant 0$ и $\mu \in R^{(k)}$, не равные нулю, одновременно, при которых $\lambda^T \mathbf{f}(\mathbf{x}) + \mathbf{v}^T \mathbf{g}(\mathbf{x}) + \mu^T \mathbf{h}(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{l} v_j g_j(\mathbf{x}) + \sum_{j=1}^{k} \mu_s h_s(\mathbf{x}) \geqslant 0$ при всех $\mathbf{x} \in B$.

Данное следствие используется при доказательстве теоремы Куна — Таккера.

Приложение 5

Элементы теории двойственности геометрического программирования

Рассмотрим некоторые важные результаты теории двойственности геометрического программирования, позволяющие разрабатывать методы решения задач. Результаты опираются на обобщение классического неравенства для среднего арифметического и геометрического и используют идеи выпуклого программирования.

Сформулируем геометрическое неравенство, которое является ключом к доказательству основной леммы геометрического программирования. Это неравенство устанавливается в следующей теореме.

Теорема 1. *Пусть* $\mathbf{x}$ *— произвольный вектор с* n *компонентами, а* δ *— произвольный вектор с* n *неотрицательными компонентами. Эти два вектора удовлет-*

воряют геометрическому неравенству:

$$\sum_{i=1}^{n} x_i\delta_i \leqslant \left(\sum_{i=1}^{n} \delta_i\right) \ln \left(\sum_{i=1}^{n} e^{x_i}\right) + \sum_{i=1}^{n} \delta_i \ln \delta_i - \sum_{i=1}^{n} \delta_i \ln \left(\sum_{i=1}^{n} \delta_i\right), \tag{5.1}$$

где предполагается, что $\delta \ln \delta = 0$, когда $\delta = 0$. При этом неравенство становится равенством тогда и только тогда, когда существует такое неотрицательное число B, что

$$\delta_i = Be^{x_i}, \quad i = 1, 2, \ldots, n.$$

Доказательство. Доказательство обобщенного геометрического неравенства (5.1) базируется на том факте, что показательная функция e^x — выпукла. Так, $e^z + (x - z)\, e^z \leqslant e^x$ для произвольных вещественных чисел x и z, так как прямая, касательная к экспоненте e^x в точке z, лежит строго ниже этой кривой. Причем это неравенство становится равенством тогда и только тогда, когда $x = z$.

Полагая $y = e^z$, получим, что $y + (x - \ln y)\, y \leqslant e^x$ для любого x и любого положительного y, причем равенство достигается тогда, когда $x = \ln y$. Таким образом,

$$\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} (x_i - \ln y_i)\, y_i \leqslant \sum_{i=1}^{n} e^{x_i}$$

для любого вектора $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ и любого положительного вектора $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, причем строгое равенство имеет место лишь в случае, когда $x_i = \ln y_i$, $i = \overline{1, n}$.

Выбирая произвольный, но фиксированный положительный вектор, мы заключаем, что

$$\sum_{i=1}^{n} \theta\delta_i + \sum_{i=1}^{n} (x_i - \ln \theta\delta_i)\, \theta\delta_i \leqslant \sum_{i=1}^{n} e^{x_i} \tag{5.2}$$

для произвольного x и положительного θ. При этом равенство в (5.2) будет иметь место тогда и только тогда, когда

$$x_i = \ln \theta\delta_i, \quad i = 1, 2, \ldots, n. \tag{5.3}$$

Преобразуя левую часть неравенства (5.2), находим, что

$$f(\theta) \leqslant \sum_{i=1}^{n} e^{x_i}, \tag{5.4}$$

где

$$f(\theta) = \left[\sum_{i=1}^{n} \delta_i (1 + x_i - \ln \delta_i)\right] \theta - \left[\sum_{i=1}^{n} \delta_i\right] \theta \ln \theta \tag{5.5}$$

и $\theta > 0$. Дифференцируя соотношение (5.5) по θ, получим

$$f'(\theta) = \sum_{i=1}^{n} \delta_i (x_i - \ln \delta_i) - \left(\sum_{i=1}^{n} \delta_i\right) \ln \theta \tag{5.6}$$

и

$$f''(\theta) = -\frac{\sum_{i=1}^{n} \delta_i}{\theta}. \tag{5.7}$$

Из (5.7) следует, что $f(\theta)$ строго вогнута для $\theta > 0$, так как $f''(\theta) < 0$. Поэтому из (5.6) вытекает, что $f(\theta)$ имеет единственную точку максимума θ', которая определяется соотношением

$$\ln \theta' = \frac{\sum_{i} \delta_i (x_i - \ln \delta_i)}{\sum_{i=1}^{n} \delta_i}. \tag{5.8}$$

Подставляя значение θ' в (5.5) и используя (5.4), получим, что

$$\left(\sum_{i=1}^{n} \delta_i\right)\theta' \leqslant \sum_{i=1}^{n} e^{x_i}.$$

Тогда из (5.8) следует, что

$$\ln\left(\sum_{i=1}^{n} \delta_i\right) + \ln\theta' \leqslant \ln\left(\sum_{i} e^{x_i}\right),$$

или

$$\sum_{i=1}^{n} \delta_i x_i \leqslant \left(\sum_{i=1}^{n} \delta_i\right)\ln\left(\sum_{i} e^{x_i}\right) + \sum_{i=1}^{n} \delta_i \ln\delta_i - \left(\sum_{i=1}^{n} \delta_i\right)\ln\left(\sum_{i=1}^{n} \delta_i\right)$$

для произвольного x и положительного δ. Согласно соотношениям (5.3) и (5.8), это неравенство становится равенством тогда и только тогда, когда

$$x_i = \ln\delta_i + \frac{\sum_{j=1}^{n} \delta_j (x_j - \ln\delta_j)}{\sum_{j=1}^{n} \delta_j}, \quad i = \overline{1, n} \tag{5.9}$$

или (что эквивалентно), когда $x_i = \ln\delta_i - \ln B$, где $B > 0$.

Итак, теорема 1 доказана в случае, когда все компоненты δ — положительны. Если же все компоненты δ — нули, то геометрическое неравенство также удовлетворяется, т. е. обе его части равны 0.

Геометрическое неравенство (5.1) эквивалентно простому обобщению классического неравенства для среднего арифметического и геометрического. Это можно показать, выбрав вектор δ, компоненты которого удовлетворяют условию нормализации

$$\sum_{i=1}^{n} \delta_i = 1. \tag{5.10}$$

В этом случае геометрическое неравенство принимает следующий вид:

$$\sum_{i=1}^{n} \delta_i (x_i - \ln\delta_i) \leqslant \ln\left(\sum_{i=1}^{n} e^{x_i}\right), \tag{5.11}$$

так как x произволен, а δ_i — положительны, то можно выбрать

$$x_i = \ln(\delta_i T_i), \quad i = \overline{1, n}. \tag{5.12}$$

Тогда из (5.11) следует, что $\sum_{i=1}^{n} \delta_i \ln T_i \leqslant \ln\left(\sum_{i=1}^{n} \delta_i T_i\right)$. (5.13)

Поскольку показательная функция является монотонно возрастающей, из неравенства (5.13) следует, что

$$\prod_{i=1}^{n} T_i^{\delta_i} \leqslant \sum_{i=1}^{n} \delta_i T_i. \tag{5.14}$$

Неравенство (5.14) является *классическим геометрическим неравенством*, оно справедливо для любых $T_i > 0$, $i = \overline{1, n}$, и положительных δ_i, удовлетворяющих условию нормализации $\sum_{i=1}^{n} \delta_i = 1$. Левая часть (5.14) представляет собой средневзвешенное геометрическое, а правая — средневзвешенное арифметическое для положительных чисел T_i, $i = \overline{1, n}$.

Неравенство (5.14) становится строгим равенством тогда и только тогда, когда для всех $i = \overline{1, n}$ $T_i = T$.

Используем геометрическое неравенство для доказательства основной леммы геометрического программирования. Она заключается в следующем.

Лемма. *Если* t *удовлетворяет ограничениям прямой задачи* A, *а* δ *удовлетворяет ограничениям соответствующей двойственной задачи* B, то $g_0(\mathrm{t}) \geqslant v(\delta)$.

Кроме того, при тех же условиях $g_0(\mathrm{t}) = v(\delta)$ *тогда и только тогда, когда*

$$\delta_i = \begin{cases} \dfrac{c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}}{g_0(\mathrm{t})}, & i \in I[0], \\ \lambda_k(\delta)\, c_i t_1^{a_{i1}} \ldots t_m^{a_{im}}, & i \in I[k], \quad k = \overline{1, p} \end{cases} \tag{5.15}$$

Здесь $g_0(\mathrm{t})$ и $v(\delta)$ — целевые функции прямой и двойственной задач, задаваемые соотношениями (6.19) и (6.24) главы 5 соответственно.

Доказательство. Так как все члены $u_i(\mathrm{t})$ позинома $g_k(\mathrm{t})$ положительны, то можно задать

$$x_i = \ln u_i(\mathrm{t}), \quad i \in I[k].$$

Применив геометрическое неравенство, получим

$$\left(\sum_{i \in I[k]} \delta_i\right) \ln g_k(\mathrm{t}) \geqslant \sum_{i \in I[k]} \delta_i \left\{\ln u_i(\mathrm{t}) - \ln(\delta_i)\right\} + \left\{\sum_{i \in I[k]} \delta_i\right\} \ln \left\{\sum_i \delta_i\right\}. \tag{5.16}$$

Нетрудно показать, что это неравенство эквивалентно неравенству

$$g_k(\mathrm{t})^{\lambda_k(\delta)} \geqslant \left[\prod_{i \in I[k]} \left(\frac{u_i(\mathrm{t})}{\delta_i}\right)^{\delta_i}\right] \lambda_k(\delta)^{\lambda_k(\delta)}, \tag{5.17}$$

где

$$\lambda_k(\delta) = \sum_{i \in I[k]} \delta_i.$$

В частности,

$$g_0(\mathrm{t}) \geqslant \prod_{i \in I[0]} \left(\frac{u_i(\mathrm{t})}{\delta_i}\right)^{\delta_i}, \tag{5.18}$$

поскольку δ удовлетворяет условию нормализации $\lambda_0(\delta) = 1$. Из неравенства (5.17) также следует, что

$$1 \geqslant g_k(\mathrm{t})^{\lambda_k(\delta)} \geqslant \left\{\prod_{i \in I[k]} \left(\frac{u_i(\mathrm{t})}{\delta_i}\right)^{\delta_i}\right\} \cdot \lambda_k(\delta)^{\lambda_k(\delta)}, \quad k = 1, \ldots, p, \tag{5.19}$$

поскольку t удовлетворяет ограничениям $g_k(\mathrm{t}) \leqslant 1$.

Умножая неравенство (5.18) на произведение неравенств (5.19), получим

$$g_0(\mathrm{t}) \geqslant \left[\prod_{i=1}^{n} \left(\frac{u_i(\mathrm{t})}{\delta_i}\right)^{\delta_i}\right] \cdot \prod_{k=1}^{p} \lambda_k^{\lambda_k}. \tag{5.20}$$

Учитывая, что $u_i(\mathrm{t})$ является i-м членом прямой задачи А, $u_i = c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots \ldots t_m^{a_{im}}$, и подставляя его выражение в (5.18), будем иметь:

$$g_0(\mathrm{t}) \geqslant v(\delta)\, t_1^{\sum\limits_{i=1}^{n} a_{i1}\delta_i} \ldots t_m^{\sum\limits_{i=1}^{n} a_{im}\delta_i}.$$

Так как вектор δ удовлетворяет условиям ортогональности, то последнее неравенство сводится к (5.21):

$$g_0(\mathrm{t}) \geqslant v(\delta). \tag{5.21}$$

Итак, первая часть леммы доказана.

Перейдем к доказательству второй части леммы. Допустим, что $g_0(\mathrm{t}) = v(\delta)$, где t и δ удовлетворяют соответственно ограничениям прямой и двойственной задач. Тогда неравенство (5.18) и все неравенства (5.19) должны быть равенствами. Поэтому

$$g_k(\mathrm{t})^{\lambda_k} = 1, \quad k = 1, 2, \ldots, p \tag{5.22}$$

и в соответствии со вторым утверждением теоремы существуют такие неотрицательные числа

$$B_k, \quad k = 0, 1, \ldots, p, \text{ что}$$

$$\delta_i = B_k u_i(\mathrm{t}), \quad i \in I[k], \quad k = 0, 1, \ldots, p. \tag{5.23}$$

Суммируя (5.23) по $I[k]$, получим

$$\lambda_k(\delta) = B_k g_k(\mathrm{t}), \quad k = 0, 1, \ldots, p, \tag{5.24}$$

откуда $B_0 = \dfrac{1}{g_0(\mathrm{t})}$, так как $\lambda_0(\delta) = 1$.

Рассмотрим теперь соотношение (5.24) для случая, когда $k \neq 0$. Если $B_k = 0$, то из (5.24) следует, что $\lambda_k(\delta) = 0$ и $\delta_i = 0$ для всех $i \in I[k]$. Следовательно, в этом случае соотношения (5.15) удовлетворяются. Если же $B_k \neq 0$, то и $\lambda_k \neq 0$ и потому, как следует из (5.22), $g_k(\mathrm{t}) = 1$. Но тогда в силу равенства (5.24) $B_k = \lambda_k(\delta)$, и потому

$$\delta_i = \lambda_k(\delta) \cdot u_i(\mathrm{t}) = \lambda_k(\delta) \cdot c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}.$$

Следовательно, соотношения (5.15) справедливы. Таким образом, мы показали, что если $g_0(\mathrm{t}) = v(\delta)$, то соотношения (5.15) выполняются.

Теперь допустим, что t и δ дополнительно к ограничениям прямой и двойственной задач удовлетворяют и соотношениям (5.15). Тогда из второго утверждения теоремы 1 следует, что неравенство (5.17) становится равенством, а кроме того, все неравенства в соотношении (5.19) также представляют собой равенства. Действительно, если $\lambda_k(\delta) = 0$, то и $\delta_i = 0$ для всех $i \in I[k]$, и потому оба неравенства (5.17) — (5.18) становятся равенствами.

Поэтому рассмотрим случай, когда $\lambda_k(\delta) \neq 0$. Тогда из (5.15) следует, что $g_k(\mathrm{t}) = 1$, это обстоятельство вместе со вторым утверждением теоремы 1 дает основания для вывода, что все неравенства (5.17) и (5.18) будут равенствами. Поэтому

$$g_0(\mathrm{t}) = v(\delta).$$

На этом доказательство геометрической леммы завершается.

Основная лемма ГП показывает, что условный минимум прямой функции $g_0(\mathrm{t})$ больше или равен условному максимуму двойственной задачи $v(\delta)$. Чтобы показать, что эти величины фактически равны, необходимо показать, что существуют такие векторы t и δ, которые удовлетворяют как соотношениям (5.15), так и ограничениям прямой и двойственной задачи соответственно.

Это устанавливается в следующей основной теореме геометрического программирования, связывающей оптимальные решения пары двойственных задач.

Теорема 2. Пусть прямая задача А совместна, и существует решение t*, такое, что $g_k(\mathrm{t}^*) < 1$, $k = \overline{1, p}$ и существует оптимальное решение. Тогда:

1) соответствующая двойственная задача совместна и существует точка, удовлетворяющая двойственным ограничениям, в которой достигается условный максимум двойственной функции $v(\delta)$;

2) максимальное значение целевой функции двойственной задачи равно минимальному значению целевой функции прямой задачи;

3) если t* — минимизирующая точка прямой задачи А, то существуют неотрицательные множители Лагранжа μ_k^*, $k = \overline{1, p}$, такие, что функция Лагранжа

$$L(\mathrm{t}, \mu) = g_0(\mathrm{t}) + \sum_{k=1}^{p} \mu_k [g_k(\mathrm{t}) - 1] \tag{5.25}$$

обладает тем свойством, что

$$L(\mathrm{t}^*, \mu) \leqslant L(\mathrm{t}^*, \mu^*) = g_0(\mathrm{t}^*) \leqslant L(\mathrm{t}, \mu^*), \tag{5.26}$$

для произвольных $t_j > 0$ и произвольных $\mu_k \geqslant 0$. Кроме того, существует максимизирующий вектор δ^* двойственной задачи с компонентами, определяемыми из условий

$$\delta_i^* = \begin{cases} \dfrac{c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}}{g_0(\mathbf{t})}, & i \in I\,[0], \\ \dfrac{\mu_k c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}}}{g_0(\mathbf{t})}, & i \in I\,[k], \quad k = \overline{1,\ p}, \end{cases} \tag{5.27}$$

где $\mathbf{t} = \mathbf{t}^*$, $\mu = \mu^*$.

Далее

$$\lambda_k(\delta^*) = \frac{\mu_k^*}{g_0(\mathbf{t}^*)}. \tag{5.28}$$

4) Если δ^* — максимизирующая точка двойственной задачи В, то любая минимизирующая точка t^* прямой задачи А удовлетворяет системе уравнений

$$c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}} = \begin{cases} \delta_i^* v(\delta^*), & i \in I\,[0], \\ \dfrac{\delta_i^*}{\lambda_k(\delta^*)}, & i \in I\,[k], \end{cases} \tag{5.29}$$

где k пробегает все положительные целочисленные значения, для которых $\lambda_k(\delta^*) > 0$.

Доказательство. Для доказательства перейдем к преобразованной прямой задаче A_z.

Как известно, показательная функция $u_i = e^{\sum\limits_j a_{ij} z_j}$ является выпуклой на всем пространстве E_m. Отсюда следует, что при положительных c_i функция вида

$$g(\mathbf{z}) = \sum_i c_i e^{\sum\limits_{j=1}^{m} a_{ij} z_j}$$

выпукла, как положительная линейная комбинация выпуклых функций. Аналогичным образом, область, определяемая неравенством $g_k(\mathbf{z}) = c_k e^{\sum\limits_j a_{kj} z_j} \leqslant 1$ будет выпуклой. Поэтому преобразованная прямая задача A_z выпукла, что позволяет применить к задаче A_z теорему Куна — Таккера.

Допустим теперь, что прямая задача А совместна и существует минимизирующая точка $\mathbf{t}^*$, которая удовлетворяет ограничениям прямой задачи. Тогда и преобразованная задача A_z совместна и преобразованная прямая функция $g_0(\mathbf{z})$ имеет точку минимума $\mathbf{z}^*$,

где $z_j^* = \ln t_j^*$, $j = \overline{1,\ m}$

и

$$g_k(\mathbf{z}) \leqslant 1, \quad k = \overline{1,\ p}.$$

Применяя теорему Куна—Таккера, убеждаемся, что существуют неотрицательные множители Лагранжа μ_k^*, $k = \overline{1,\ p}$, такие, что функция Лагранжа

$$L(\mathbf{z}, \mu) = g_0(\mathbf{z}) + \sum_{k=1}^{p} \mu_k (g_k(\mathbf{z}) - 1) \tag{5.30}$$

обладает свойством:

$$L(\mathbf{z}^*, \mu) \leqslant g_0(\mathbf{z}^*) = L(\mathbf{z}^*, \mu^*) \leqslant L(\mathbf{z}, \mu^*) \tag{5.31}$$

для произвольного $\mathbf{z}$ и произвольных $\mu_k \geqslant 0$.

Таким образом, $\mathbf{z}^*$ является точкой безусловного минимума, и тогда

$$\frac{\partial L(\mathbf{z}^*, \mu^*)}{\partial z_q} = 0, \quad q = \overline{1,\ m}.$$

Отсюда следует, что

$$\sum_{i\in I[0]} a_{iq} e^{\sum\limits_j a_{ij} z_j^*} + \sum_{k=1}^{p} \mu_k^* \left(\sum_{i\in I[k]} a_{iq} c_i e^{\sum\limits_j a_{ij} z_j^*} \right) = 0, \quad q = \overline{1,\ m}. \tag{5.32}$$

Разделив эти уравнения на $g_0(z^*)$, убеждаемся в том, что вектор δ^* с компонентами

$$\delta_i^* = \begin{cases} \dfrac{c_i e^{\sum\limits_{j=1}^{m} a_{ij} z_j^*}}{g_0(z^*)}, & i \in I[0], \\ \dfrac{\mu_k^* c_i e^{\sum\limits_{j=1}^{m} a_{ij} z_j^*}}{g_0(z^*)}, & i \in I[k], \end{cases} \tag{5.33}$$

удовлетворяет условиям ортогональности для двойственной задачи.

Из соотношения (5.33) также ясно, что δ^* удовлетворяет и условиям неотрицательности, так как все $\mu_k^* \geqslant 0$. Таким образом, двойственная задача В совместна.

Используя (5.33), мы получаем, что

$$\lambda_k(\delta^*) = \frac{\mu_k^* g_k(z^*)}{g_0(z^*)}, \quad k = 1,\ 2,\ \ldots,\ p. \tag{5.34}$$

Далее из равенства $g_0(z^*) = L(z^*, \mu^*)$ следует, что

$$\mu_k^* (g_k(z^*) - 1) = 0,$$

поэтому

$$\lambda_k(\delta^*) = \frac{\mu_k^*}{g_0(z^*)}. \tag{5.35}$$

Поэтому соотношения (5.33) могут быть записаны в виде

$$\delta_i^* = \begin{cases} \dfrac{c_i e^{\sum\limits_j a_{ij} z_j^*}}{g_0(z^*)}, & i \in I[0], \\ \lambda_k(\delta^*) c_i e^{\sum\limits_{j=1}^{m} a_{ij} z_j^*}, & i \in I[k], \quad k = \overline{1,\ p}. \end{cases} \tag{5.36}$$

Используя (5.29), перепишем соотношения (5.35) в терминах переменных t^*, тогда на основании второго утверждения основной леммы мы заключаем, что $g_0(t^*) = v(\delta^*)$.

Итак, мы доказали первые три утверждения основной теоремы геометрического программирования.

Перейдем к доказательству четвертого утверждения.

Допустим, что t^* является минимизирующей точкой прямой задачи А, а δ^* — максимизирующей точкой соответствующей двойственной задачи В. Тогда в соответствии с только что доказанным $g_0(t^*) = v(\delta^*)$. Используя второе утверждение основной леммы, заключаем, что t^* и δ^* удовлетворяют соотношениям (5.29) и, следовательно, соотношению

$$c_i t_1^{a_{i1}} t_2^{a_{i2}} \ldots t_m^{a_{im}} = \begin{cases} \delta_i v(\delta), & i \in I[0], \\ \dfrac{\delta_i}{\lambda_k(\delta)}, & i \in I[k], \end{cases} \tag{5.37}$$

где k принимает такие положительные целые значения, при которых $\lambda_k(\delta) > 0$.

На этом доказательство основной теоремы закончено.

Основная теорема позволяет найти оптимальное решение δ^* двойственной задачи по известному оптимальному решению прямой t^* и наоборот, для этого используют соотношения (5.27) и (5.37).

СПИСОК ЛИТЕРАТУРЫ

1. Акоф Р., Сасиени М. Основы исследования операций. М. Мир, 1971.
2. Акоф Р., Райвет П. Исследования операций. М., Мир, 1966.
3. Аоки Масао. Введение в методы оптимизации. Основы и приложения. Пер. с англ. М., 1977.
4. Беллман Р. Динамическое программирование. Изд-во иностранной литературы, 1960.
5. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М., Наука, 1965.
6. Беллман Р., Калаба Р. Динамическое программирование и современная теория управления. М., Наука, 1969.
7. Бусленко Н. П. Моделирование сложных систем. М., Наука, 1968.
8. Бусленко Н. П., Коваленко И. Н., Калашников В. В. Лекции по теории сложных систем. М., Советское радио, 1973.
9. Вагнер Г. Основы исследования операций, т. 1. М., Мир, 1972; т. 2—3. М., 1973.
10. Вентцель Е. С. Исследование операций. М., Советское радио, 1972.
11. Вентцель Е. С. Элементы динамического программирования. М., Наука, 1964.
12. Волкович В. Л. Методы принятия решений по множеству критериев оптимальности (обзор). Труды семинара. Сложные системы. Вып. 1. К., Наукова думка, 1968.
13. Вопросы анализа и процедуры принятия решений. Сб. переводов. М., Мир, 1976.
14. Гермейер Ю. Б. Введение в теорию исследования операций. М., Наука, 1971.
15. Гермейер Ю. Б. О свертывании векторных критериев эффективности в единый критерий при наличии неопределенности в параметрах свертывания.— В кн.: Кибернетику — на службу коммунизму. М., Энергия, 1971.
16. Гидрович С. Р., Сыроежин И. М. Игровое моделирование экономических процессов (деловые игры). М., Экономика, 1976.
17. Гольштейн Е. Г. Теория двойственности в математическом программировании и ее приложения. М., Наука, 1971.
18. Гольштейн Е. Г., Юдин Д. Б. Новые направления в линейном программировании. М., Наука, 1966.
19. Даффин Р., Питерсон Э., Зенер К. Геометрическое программирование. М., Мир, 1972.
20. Емельянов С. В., Дудин Е. Б., Ларичев О. П., Малевич А. А., Наппельбаум Э. П., Озерной В. М. Подготовка и

принятие решений в организационных системах управления.— В кн.: Итоги науки, серия Техническая кибернетика, 1969.

21. Ермольев Ю. М. Методы решения нелинейных экстремальных задач.— Кибернетика, № 4, 1966.
22. Ермольев Ю. М. Методы стохастического программирования. Наука, Главная редакция физ.-мат. литературы. М., 1976.
23. Зуховицкий С. И., Авдеева Л. И. Линейное и выпуклое программирование. М., Наука, 1966.
24. Карр Ч., Хоув Ч. Количественные методы принятия решений в управлении и экономике. М., Мир, 1966.
25. Карпелевич Ф. И., Садовский Л. Е. Элементы линейной алгебры и линейного программирования. М., Наука, 1967.
26. Калихман И. Л. Сборник задач по линейной алгебре и линейному программированию. М., Высшая школа, 1969.
27. Конвей Р. В., Максвелл В. Л., Миллер Л. В. Теория расписаний. Пер. с англ. М., Наука, 1975.
28. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование. М., Наука, 1970.
29. Кофман А. Методы и модели исследования операций. М., Мир, 1966.
30. Кофман А., Крюон Р. Массовое обслуживание. Теория и приложения. М., Мир, 1965.
31. Кофман А., Фор Р. Займемся исследованием операций. М., Мир, 1965.
32. Кофман А., Анри-Лабордер А. Методы и модели исследования операций. Пер. с франц./Под ред. Н. П. Бусленко. М., Мир, 1977.
33. Лэсдон Л. Оптимизация больших систем. Главная редакция физ.-мат. литературы изд-ва «Наука». М., 1975.
34. Морз Ф. М., Кимбел Д. Е. Методы исследования операций. М., Советское радио, 1956.
35. Нейман Дж., Моргенштерн О. Теория игр и экономическое поведение. М., Наука, 1970.
36. Озерной В. М. Принятие решений (обзор).— Автоматика и телемеханика, № 11, 1971.
37. Оптнер С. Л. Системный анализ для решения деловых и промышленных проблем. М., Советское радио, 1969.
38. Растригин Л. А. Статистические методы поиска. М., Наука, 1968.
39. Рыжиков Ю. И. Управление запасами. М., Наука, 1969.
40. Саати Т. Элементы теории массового обслуживания и ее приложения. М., Советское радио, 1971.
41. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. М., Мир, 1973.
42. Финкельштейн Ю. Ю. Приближенные методы и прикладные задачи дискретного программирования. М., Наука, 1976.
43. Фишберн П. Теория полезности для принятия решений. Пер. с англ. М., Наука, 1978.
44. Форд Л., Фалкерсон Д. Потоги в сетях. М., Мир, 1966.
45. Хедли Дж. Нелинейное и динамическое программирование. М., Мир, 1967.
46. Хедли Дж., Уайтин Т. Научное управление запасами. М., Мир, 1967.
47. Химмельблау Д. Прикладное нелинейное программирование. М., Мир, 1975.

48. Ховард Р. Динамическое программирование и марковские цепи. М., Советское радио, 1964.
49. Черчмен Р., Акоф Р., Арноф Л. Введение в исследование операций. М., Наука, 1968.
50. Численные методы условной оптимизации/ Под ред. Ф. Гилла и У. Мюррея. М., Мир, 1977.
51. Чуев Ю. В., Мельников П. М., Петухов С. И., Степанов Г. Ф., Шор Я. Б. Основы исследования операций в военной технике. М., Советское радио, 1965.
52. Чуев Ю. В., Спехова Г. П. Технические задачи исследования операций. М., Советское радио, 1971.
53. Шеннон Р. Имитационное моделирование систем.— Искусство и наука. Пер. с англ. М., Мир, 1978.
54. Шкурба В. В., Подчасова Т. П., Пшичук А. Н., Тур Л. П. Задачи календарного планирования и методы их решения. К., Наукова думка, 1966.
55. Уайльд Д. Дж. Методы поиска экстремума. М., Мир, 1966.
56. Эрроу Дж., Удзава Д. Исследование по линейному и нелинейному программированию. М., Изд-во иностранной лит-ры, 1962.
57. Юдин Д. Б., Гольштейн Е. Г. Задачи и методы линейного программирования. М., Советское радио, 1964.
58. Юдин Д. Б., Гольштейн Е. Г. Задачи линейного программирования транспортного типа. М., Наука, 1969.
59. Юдин Д. Б. Математические методы управления в условиях неполной информации. М., Советское радио, 1974.
60. Янг С. Системное управление организацией. М., Советское радио, 1972.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Абелева группа 374
Аддитивная функция 12
Аксиомы полезности 21
Асимптотический метод Гомори 234

Балансовая модель 55
Беллман принцип оптимальности 312
Блочное программирование 119

Графический метод решения ЛП-задач 63

Двойственность в линейном программировании 86
— в нелинейном программировании 284
Данциг 65
Декомпозиция Данцига — Вульфа 119
Динамическая модель управления запасами 345
Динамическая модель выбора объема партии 349
Допустимое базисное решение 60

Задача выбора маршрута на сети 17
— массового обслуживания 16
— о коммивояжере 253
— о выборе траектории 252
— о назначениях (выбора) 159
— о нахождении максимального потока на сети 183
— определения оптимального ассортимента 53
— о покрытии 212
— распределения ресурсов 15
— ремонта и замены оборудования 15
— управления запасами 14
— сетевого планирования и управления 16
— транспортная 134
— транспортная с ограниченными пропускными способностями 171
— транспортная на сети 190

Имитационная модель 36

Каноническая форма ЛП-задачи 57
Квадратичное программирование 286
Критерий Вальда 26
— Гурвица 26
— Лапласа 26
— Сэвиджа 26
Координирующая задача 122
Конус 364
Крайняя точка 362

Лемма основная геометрического программирования 381
— о крайней точке выпуклого многогранника 362
— Фаркаша 365

Метод венгерский 159, 163
— ветвей и границ 245
— вычеркивания 146
— для нахождения опорных планов 147
— двойственный симплекс-метод 111
— Жордана — Гаусса 68
— искусственных переменных 78
— минимального элемента 148
— множителей Лагранжа 269
— обратной матрицы 100
— отсекающих плоскостей Гомори 218
— потенциалов 150
— последовательных приближений 329
— северо-западного угла 147
— симплекс-метод 64, 72
— Форда — Фалкерсона 184

Опорный план 142
Основная теорема линейного программирования 61
— — двойственности 88
— — геометрического программирования 296
Основное рекуррентное соотношение 312

Позином 297
Приведенная форма Смита 371

Признак оптимальности опорного плана 75
— — псевдоплана 113
Принятие решений в условиях определенности 19
— в условиях неопределенности 25
— в условиях риска 24
Принцип рентабельности 88
Принципы разработки моделей 32
Псевдоплан 112

Разрез на сети 184
Разрешающие множители Канторовича 100
Расширенная форма ЛП-задачи 58

Седловая точка 277
— — условия существования 277
Системный подход 10
Сопряженный базис 112
Степень трудности 299

Теорема Куна-Таккера 274
— Фана 377
— Форда — Фалкерсона 184
— Хана — Банаха 377
— о седловой точке 277

Условия баланса 136
— вогнутости и выпуклости 277
— дополняющей нежесткости 276
— ортогональности 296
— нормализации 296
— разрешимости транспортной задачи 136, 139
— регулярности 274

Чувствительность моделей задач ЛП 96

Циклическая группа 376

Этапы исследования операций 11

ОГЛАВЛЕНИЕ

Предисловие 3
Введение 7

Глава 1. Основные принципы исследования операций 11

§ 1. Основные этапы операционного исследования 11
§ 2. Типичные классы задач 14
§ 3. Некоторые принципы принятия решений в задачах исследования операций 18
§ 4. Разработка математических моделей в задачах исследования операций 30
§ 5. Имитационное моделирование систем организационного управления 36
§ 6. Деловые игры как модели 50

Глава 2. Линейное программирование 53

§ 1. Постановка задачи линейного программирования и исследования ее структуры 56
§ 2. Симплекс-метод 64
§ 3. Нахождение допустимых базисных решений 78
§ 4. Двойственная задача линейного программирования 85
§ 5. Метод обратной матрицы 100
§ 6. Двойственный симплекс-метод 111
§ 7. Декомпозиционные методы решения задач линейного программирования большой размерности 119

Глава 3. Транспортная задача линейного программирования 134

§ 1. Постановка и основные свойства транспортной задачи 134
§ 2. Метод потенциалов 150
§ 3. Венгерский метод 159
§ 4. Транспортные сети 179
§ 5. Транспортная задача в сетевой постановке 190
§ 6. Алгоритм декомпозиции для решения транспортной задачи 201

Глава 4. Дискретное программирование 209

§ 1. Математические модели задач дискретного программирования 210
§ 2. Метод отсекающих плоскостей 218
§ 3. Асимптотическое целочисленное программирование 234
§ 4. Метод ветвей и границ 245

Глава 5. Нелинейное программирование 264

§ 1. Классический метод определения условного экстремума 265
§ 2. Метод множителей Лагранжа 269
§ 3. Задача нелинейного программирования при ограничениях-неравенствах 272
§ 4. Двойственность в задачах оптимизации 281
§ 5. Квадратичное программирование 285
§ 6. Геометрическое программирование 290

Глава 6. Динамическое программирование 308

§ 1. Сущность вычислительного метода 308
§ 2. Динамическое программирование для задач с несколькими ограничениями и переменными 322
§ 3. Задачи управления запасами 332
§ 4. Динамические задачи управления запасами 345
Заключение 355
Приложение 1 359
Приложение 2 366
Приложение 3 374
Приложение 4 377
Приложение 5 378
Список литературы 385
Предметный указатель 388

Юрий Петрович Зайченко

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Редактор Ж. Г. Давиденко
Художественный редактор С. П. Духленко
Технический редактор Т. И. Трофимова
Корректор А. И. Супрун

Информ. бланк № 4111

Сдано в набор 31.10.78. Подп. в печать 7.06.79. БФ 08664. Формат $60\times90^{1}/_{16}$. Бумага типогр. № 3. Лит. гарн. Выс. печать. 24,5 печ. л. 23,79 уч.-изд. л. Тираж 10000 экз. Изд. № 4397. Зак. № 8—3164.
Цена 1 р. 10 к.

Головное издательство издательского объединения «Вища школа», 252054, Киев-54, ул. Гоголевская, 7.

Отпечатано с матриц Головного предприятия Республиканского производственного объединения «Полиграфкнига» Госкомиздата УССР, 252057, Киев-57; ул. Довженко, 3 в Киевской книжной типографии научной книги, 252004, Киев-4, Репина, 4.
Зак. 9-719.

1 руб. 10 коп.