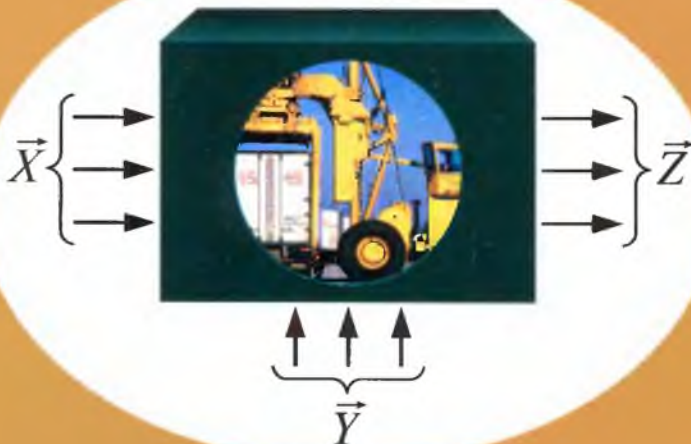


Б.А. Гладких

**МЕТОДЫ ОПТИМИЗАЦИИ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
ДЛЯ БАКАЛАВРОВ ИНФОРМАТИКИ**



Часть I

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ИНФОРМАТИКИ

Б. А. Гладких

МЕТОДЫ ОПТИМИЗАЦИИ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
ДЛЯ БАКАЛАВРОВ ИНФОРМАТИКИ

Часть I

Введение в исследование операций
Линейное программирование



Томск — 2009

УДК 517.8
Г 522

Гладких Б. А. Методы оптимизации и исследование операций для бакалавров информатики. Ч. I. Введение в исследование операций. Линейное программирование: Учебное пособие. — Томск: Изд-во НТЛ, 2009. — 200 с.

ISBN 978-5-89503-410-1

Книга написана на основе лекций, в течение ряда лет читавшихся автором на факультете информатики Томского государственного университета.

В первую часть вошли общее введение в исследование операций, а также стандартные разделы, относящиеся к линейному программированию: общая задача, симплексный метод, теория двойственности, специальные задачи линейного программирования (транспортная и задача о назначениях), целочисленное линейное программирование.

Учебное пособие соответствует Государственному образовательному стандарту по направлениям «информационные технологии» и «прикладная информатика», но может быть использовано для студентов, обучающихся по другим инженерным и экономическим направлениям.

УДК 517.8

Рецензент:

доктор физ.-мат. наук, профессор К. И. Лившиц

ISBN 978-5-89503-410-1

© Б. А. Гладких, 2009

© Издательство НТЛ,
обложка, 2009

Оглавление

Предисловие	5
1. Введение в исследование операций	7
1.1. Предмет исследования операций	7
1.2. Краткий исторический очерк	10
1.3. Методология операционного исследования	18
1.4. Проблемы нахождения оптимальных решений	27
2. Примеры и каноническая форма задачи линейного программирования	32
2.1. Задача о производственном плане	32
2.2. Задача о диете	34
2.3. Каноническая форма	35
2.4. Исторические замечания	40
3. Повторение линейной алгебры	43
3.1. Преобразование Жордана	43
3.2. Системы линейных уравнений общего вида	50
3.3. Линейные (векторные) пространства	55
3.4. Решение системы линейных уравнений как процесс последовательного замещения векторов в базисе	62
3.5. Выпуклые множества в линейных пространствах	65
3.6. Исторические замечания	72
4. Симплексный метод	73
4.1. Графическая интерпретация задачи линейного программирования	73
4.2. Свойства планов задачи линейного программирования	76

4.3.	Теория симплексного метода	83
4.4.	Практический алгоритм симплексного метода	94
4.5.	Метод искусственного базиса	100
4.6.	Исторические замечания	107
5.	Теория двойственности	110
5.1.	Симметричные двойственные задачи	110
5.2.	Несимметричные двойственные задачи	113
5.3.	Первая теорема двойственности	115
5.4.	Вторая теорема двойственности	121
5.5.	Экономическая интерпретация двойственности	124
5.6.	Исторические замечания	128
6.	Транспортная задача	132
6.1.	Постановка и формы записи транспортной задачи	132
6.2.	Свойства транспортной задачи	136
6.3.	Построение исходных опорных планов	141
6.4.	Критерий оптимальности транспортной задачи	149
6.5.	Переход к новому опорному плану	155
6.6.	Исторические замечания	159
7.	Задача о назначении	161
7.1.	Постановка и формализация	161
7.2.	Свойства задачи о назначении	163
7.3.	Независимые нули и паросочетания	166
7.4.	Практический алгоритм венгерского метода	169
7.5.	Исторические замечания	176
8.	Дискретное линейное программирование	177
8.1.	Классификация задач и методов дискретного программирования	177
8.2.	Методы отсечения	179
8.3.	Метод ветвей и границ	188
8.4.	Исторические замечания	195
	Литература	197

Предисловие

Изучение математических методов оптимизации и исследования операций входит в образовательные программы многих направлений подготовки специалистов: математиков, инженеров, экономистов, программистов. И хотя содержание соответствующих курсов во многом пересекается, методический подход к изложению материала существенно различается.

Например, для математиков, в сферу компетенции которых входит теоретическое исследование тех или иных методов, принципиально важными являются вопросы сходимости и погрешности вычислений, в то время как для инженеров и экономистов, в основном пользующихся готовыми пакетами прикладных программ, внимание акцентируется на сфере применения алгоритмов и правильной интерпретации результатов.

Настоящий курс специально адаптирован для студентов, обучающихся по бакалаврским программам подготовки программистов. Этим определяется подбор материала и уровень строгости изложения. Личный опыт автора говорит о том, что для этой категории слушателей характерен конкретный, алгоритмический стиль мышления. Им хочется скорее реализовать изучаемые методы на компьютере, длинные математические выкладки и абстрактные доказательства вызывают реакцию отторжения. Для того чтобы пробудить интерес к предмету и держать его на протяжении всего курса, нужно строить изложение каждого вопроса от частного к общему, от конкретных примеров к теоретическим обобщениям, сохраняя при этом достаточный научный уровень.

Один из братьев Брэггов — известных физиков, нобелевских лауреатов — как-то заметил, что эффективность лекции определяется не тем количеством материала, которое лектор сумел в нее втиснуть, а тем, которое назавтра слушатель сможет за утренним кофе пересказать своим

близким. Исходя из этого принципа, мы старались сделать изложение предельно понятным, опуская несущественные для понимания основной идеи доказательства и частные случаи, иллюстрируя каждый шаг подробными примерами.

Учитывая небольшое число часов, отведенных образовательным стандартом на базовый бакалаврский курс оптимизации и исследования операций, мы включили в него минимально необходимый набор моделей и методов. Более глубокое изучение предмета возможно на магистерском уровне, для этого имеется множество прекрасных учебников и учебных пособий.

Глава 1

Введение в исследование операций

Название дисциплины, к изучению которой мы приступаем, состоит из двух частей: «Методы оптимизации» и «Исследование операций». И если понятие оптимизации в общих чертах известно каждому, то термин *«исследование операций — operations research»* требует дополнительных пояснений. Само это словосочетание довольно новое, оно возникло во время Второй мировой войны (см. далее краткий исторический очерк) и вначале имело прямое отношение к военным операциям. В послевоенное время границы понятия были существенно расширены, под операцией стала пониматься любая целенаправленная деятельность.

1.1. Предмет исследования операций

Исследование операций — прикладная математическая дисциплина, занимающаяся вопросами количественного обоснования решений по управлению целенаправленными процессами (операциями) в сложных системах.

В этом определении принципиально важными являются три ключевых момента.

- Во-первых, устанавливается *объект исследования*: предметом изучения являются *управляющие решения*. Такими решениями могут быть, например, расстановка сил на поле боя, план производства продукции в компании, ход в карточной игре, выбор метода лечения больного и т. д. Несмотря на содержательное разнообразие, эти примеры имеют нечто общее, состоящее в том, что последствия принимаемых решений не всегда могут быть точно установлены из-за наличия неуправляемых факторов, и действовать приходится в условиях существенной неопределенности.
- Во-вторых, подчеркивается специфический, субъективный *подход* к проблеме. Система рассматривается с точки зрения *целенаправленного* управления, то есть в анализируемой операции выделяется *оперирующая сторона* (лицо или группа лиц, принимающих решения), имеющая вполне определенную *цель*, которую она стремится достичь своими решениями. Понятие цели является центральным, определяющим при исследовании систем. Более того, само понятие «система» неотделимо от цели. Один и тот же объект реального мира, рассматриваемый с точки зрения различных целей, может представлять различные системы.
- В-третьих, определяется *метод* исследования: обоснование решения должно носить количественный характер, то есть исследование должно проводиться с помощью математических моделей и методов. Это дает возможность в ряде случаев находить не просто хорошие, а в некотором смысле наилучшие, оптимальные решения.

Коротко резюмируя фундаментальные особенности исследования операций как самостоятельной научной дисциплины, ее можно определить как *теорию оптимальных решений*.

Исследование операций образовалась как синтетическая наука на границе нескольких смежных дисциплин (см. рис. 1.1) и тесно с ними взаимодействует.

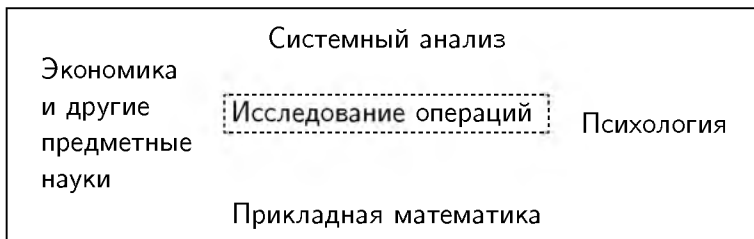


Рис. 1.1. Место исследования операций среди других наук

Прежде всего следует упомянуть *системный анализ* [1], принципы и методология которого лежат в основе любого операционного исследования.

Далее, поскольку сами операции возникают не на основе абстрактных рассуждений, а порождаются самой жизнью, то исследование операций сохраняет глубокую связь с *предметными науками* (экономикой, инженерией, военным делом и т. п.) и питается их идеями.

Очень интересной и продуктивной является связь с *психологией*. Так как в любой операции обязательно присутствует лицо, принимающее решение (ЛПР), то в теории принятия решений невозможно обойтись без таких психологических понятий, как субъективные предпочтения, уверенность, оптимизм, отношение к риску и т. д.

Как следует из определения, основным рабочим инструментом исследования операций является *прикладная математика*. Широко используются все ее разделы (математический анализ, теория вероятностей и математическая статистика, классические методы оптимизации, дискретная математика, теория графов и др.). Более того, специфические задачи исследования операций, возникшие в предметных областях, стимулировали появление новых разделов прикладной математики. Так, проблема планирования производства привела к созданию линейного программирования, частная задача анализа эффективности работы телефонных станций послужила основой для развития общей теории массово-

го обслуживания, военные конфликты стимулировали развитие общей теории игр и т. д.

Установить четкую границу раздела между математическими методами исследования операций и другими разделами прикладной математики не представляется возможным. Иногда исследование операций понимают в широком смысле, тогда появляются энциклопедические издания, содержащие почти все необходимые разделы прикладной математики [2, 3]. Этой же позиции придерживаются библиотечные работники, поэтому, скажем, библиографические карточки книг по линейному программированию и другим методам оптимизации следует искать в разделе «исследование операций».

Противоположная точка зрения [5] состоит в том, что под исследованием операций понимается только общая теория принятия решений, а другие разделы, например то же линейное программирование или теория массового обслуживания, образуют самостоятельные ветви прикладной математики.

Точно так же нет единого мнения насчет содержания курса исследования операций в вузовской программе, все определяется конкретными условиями и личными вкусами составителей образовательных стандартов. На наш взгляд, полезнее не спорить о понятиях, а научиться применять имеющийся богатый арсенал математических методов исследования операций на практике. С этой точки зрения весьма логичным является предпринятое в рамках данного курса объединение стандартных методов оптимизации, которые понимаются как способы нахождения решений в условиях определенности, с общими задачами принятия решений при неопределенности.

1.2. Краткий исторический очерк

В исследовании операций можно достаточно четко выделить три временных периода: 1) предыстория, 2) Вторая мировая война, 3) послевоенный период.

Предыстория Первые задачи теории принятия решений были сформулированы еще в XVIII веке как математические курьезы. К ним относится, в частности, знаменитый «петербургский парадокс» Даниэля Бернулли (Bernoulli, Daniel; 1700 – 1782), мы будем говорить о нем подробнее в разделе, посвященном принятию решений при риске.

В конце XIX века были сделаны попытки применить математический аппарат к экономике и социологии. К ним относятся, в частности, работы итальянского экономиста Вильфредо Парето (Pareto, Vilfredo; 1848-1923)¹. Понятия «множество Парето» и «оптимальность по Парето» широко используются в современной теории принятия решений.

В первой половине XX века началось систематическое изучение математических моделей принятия решений при неопределенности и противодействии, которое впоследствии оформилось в теорию игр. Решающую роль здесь сыграл выдающийся математик Джон фон Нейман (von Neumann, John; 1903 – 1957). Еще в 1929 г. он сформулировал и доказал основную теорему матричных игр — так называемую теорему о минимаксе, а в 1944 г. вместе с экономистом Оскаром Моргенштерном (Morgenstern, Oscar; 1902 – 1977) опубликовал классическую монографию «Theory of Games and Economic Behavior — Теория игр и экономическое поведение», которая стала библией по этому предмету для последующих поколений ученых.

К этому времени относятся и пионерские работы нашего соотечественника Леонида Витальевича Канторовича по линейному программированию (о них мы подробнее поговорим в гл. 2), исследования по моделям массового обслуживания, сетевому планированию и управлению и т. д. В целом, к середине XX века был накоплен багаж разрозненных научных результатов, достаточный для решения многих практических задач принятия решений.

¹Широкой публике имя Парето известно благодаря знаменитому «правилу 20–80», утверждающему, что во многих видах деятельности 80% результата достигается всего 20% усилий, а остальные 80% усилий тратятся на оставшиеся 20% результата.

Вторая мировая война

Мощный толчок развитию прикладных научных исследований дала Вторая мировая война. Собственно, тогда и появился термин «исследование операций», который был предложен военными, и означал научное, с применением математических моделей и методов, исследование военных операций. Исторически первым опытом успешного взаимодействия ученых с военными считается развертывание системы противовоздушной обороны Великобритании в 1938–1941 гг. с использованием новейшей радиолокационной техники. Для решения сложных технических и организационных проблем была создана междисциплинарная группа исследователей, ставшая известной по имени ее руководителя, будущего лауреата Нобелевской премии Патрика Блэккетта (Blackett, Patrick; 1897–1974), как «кружок Блэккетта». Результаты ее деятельности оказались исключительно эффективными, более того, здесь были сформулированы фундаментальные принципы любого операционного исследования: системный подход к проблеме, работа смешанными творческими коллективами, включающими специалистов различных отраслей знаний, использование математических моделей и методов.

Когда в войну вступили Соединенные Штаты, их военное командование изучило британский опыт и осознало целесообразность привлечения ученых к разработке военных операций. Многие талантливые математики и физики были мобилизованы в армию и использовались в качестве научных советников. В частности, группу по исследованию операций на флоте возглавил физик, директор вычислительного центра Массачусеттского технологического института Филипп Морз (Morse, Philip McCord; 1903–1985), считающийся «отцом исследования операций» в США. В главном штабе ВВС работал один из создателей линейного программирования математик Джордж Данциг и т. д. Имеется множество примеров успешно решенных военных задач — от оптимального размещения кораблей сторожевой охраны до обеспечения войск продовольствием и вооружением. К концу войны общее

число ученых в отделах и группах исследования операций в Англии, Америке и Канаде превышало 700 человек.

Послевоенный период

Хотя большинство специалистов, занимавшихся во время войны исследованиями операций, вернулись после ее окончания к своей довоенной деятельности, многие продолжили исследования и показали возможность использования новой синтетической науки о функциональных системах в мирное время.

Строго говоря, это время можно назвать мирным только относительно. Сразу после окончания Второй мировой войны началась «холодная война», начало которой положила знаменитая «Фултонская речь» Уинстона Черчилля, произнесенная им 5 марта 1946 г. в американском университетском городке Фултон, штат Миссури. Бывшие союзники по антигитлеровской коалиции, разделенные непреодолимыми идеологическими противоречиями, стали готовиться к третьей мировой войне, которая казалась неизбежной. Спешно создавались системы ракетно-ядерного оружия, изучались новые театры военных действий, моделировались сценарии будущих вооруженных конфликтов. Цифры военных расходов достигли астрономических значений, при этом существенная доля их тратилась на научные исследования и разработки.

Следует признать, что послевоенные годы стали золотым временем для ученых, занимавшихся прикладными математическими исследованиями во всех направлениях. При этом разработанные изначально для военных целей модели и методы стали с успехом применяться и в гражданских отраслях народного хозяйства. Созданные во время войны группы и отделы по исследованию операций постепенно трансформировались в самостоятельные консультационные фирмы — «фабрики мысли» [11], многие из которых достигли замечательных успехов.

Показательна в этом отношении судьба корпорации RAND (Research ANd Development — исследования и разработки). Она была основана BBC США и Douglas Aircraft Company в 1946 г. как чисто военный «проект RAND», но в 1948 г. преобразована в са-

мостоятельную некоммерческую компанию со значительно более широким полем деятельности. Штаб-квартира компании расположена в Санта-Монике, штат Калифорния, кроме того, имеются филиалы в нескольких городах США и Европы. В 2008 г. в ее штате было около 1600 сотрудников, годовой бюджет составлял более 200 млн долларов.

Главный офис RAND Corporation размещается в двух больших розовых зданиях, окна которых выходят на пляж. Во время обеденного перерыва многие занимаются плаванием, в одежде нет строгости, сами сотрудники общительны и держатся непринужденно. Кабинеты, как правило, изящно обставлены в соответствии с тем, как принято себе представлять профессорские апартаменты. Осуществляются строгие меры по обеспечению безопасности, но делается это исподволь. Несмотря на то, что это одно из важнейших вспомогательных учреждений страны, связанных с военной деятельностью, при его посещении скорее вспоминается курорт, чем Пентагон [11].

Эта исследовательская организация прославилась замечательными научными достижениями. За первые 20 лет существования ею было опубликовано около 11 тыс. книг, отчетов и статей. Более 30 исследователей, так или иначе связанных с RAND, удостоены Нобелевской премии. Здесь разрабатывались космические проекты, была предложена идея современных компьютерных сетей (Пол Бэрэн, 1962), созданы математические методы линейного программирования (Джордж Данциг, 1947), динамического программирования (Ричард Беллман, 1957), дифференциальных игр (Руфус Айзекс, 1954) и т. д. Выходцы из RAND не раз занимали высокие государственные посты (Джеймс Шлезингер — министр обороны в правительстве Никсона, Кондолиза Райс — государственный секретарь в правительстве Буша и др.). Научные исследования, проведенные в RAND, существенным образом влияли не только на науку, но и на национальную политику и экономику США.

В целом в послевоенное время исследование операций за рубежом переживало период расцвета. Бурно развивающаяся эко-

номика США и других передовых капиталистических стран охотно воспринимала научные методы управления, многие крупные компании пользовались услугами консультационных фирм либо организовали у себя исследовательские подразделения, помогающие развивать бизнес в условиях жесткой конкуренции.

Инновационные технологии управления стали использоваться также в государственном и муниципальном управлении. Так, например, в 1968 г. был основан Институт городского хозяйства, а в 1969 г. — Институт развития города Нью-Йорка. Оба эти института разработали весьма разнообразные и эффективные программы, в которых предусматривалось использование методов исследования операций.

Важнейшим фактором, повлиявшим на быстрое развитие прикладных математических исследований в послевоенное время, стало создание и широкое внедрение электронных вычислительных машин, позволивших реализовать сложные вычислительные и моделирующие алгоритмы.

Параллельно развивалась научно-образовательная инфраструктура в области исследования операций: были образованы соответствующие научные общества, основаны научные журналы, присваивались ученые степени, регулярно проводились национальные и международные конференции. После того как в 1968 г. Государственным банком Швеции по случаю своего 300-летия была учреждена Нобелевская премия по экономике, ею были отмечены выдающиеся достижения в области исследования операций, имеющие выраженную экономическую интерпретацию (Канторович и Купманс за теорию распределения ресурсов в 1975 г., Джон Нэш за исследования по теории игр в 1994 г. и др).

Исследование операций в Советском Союзе

Командно-административная система Советского Союза долгое время не признавала точных научных методов управления. Советская управленческая и экономическая наука,

пережившая яркий, но краткий расцвет в начале 1920-х годов², в годы культа личности Сталина была разгромлена и подменена схоластичной и крайне идеологизированной «теорией научного коммунизма». Согласно официальному учению, точные науки могли быть применены только к неодушевленным объектам, а любые общественные процессы представляли собой принципиально другую и неподвластную математике форму движения материи.

Интерес к исследованию операций пробудился в начале 1960-х годов, когда немного ослаб идеологический пресс и, благодаря усилиям выдающихся ученых – Андрея Николаевича Колмогорова (1903–1987), Акселя Ивановича Берга (1893–1979), Виктора Михайловича Глушкова (1923–1972) и ряда других, общая наука об управлении — кибернетика, бывшая до тех пор «реакционной лженаукой, формой современного механицизма»³, вышла из опалы и была поставлена «на службу коммунизму»⁴.

Инициаторами этого процесса стали военные исследовательские институты, которые были в курсе событий, происходящих за «железным занавесом» и получали по своим каналам зарубежные книги и журналы, недоступные рядовым читателям. В это время были переведены на русский язык классические зарубежные учебники [7, 8], написаны первые отечественные книги по линейному программированию [13, 14] и другим разделам исследования операций.

В середине 1960-х годов, после отставки Н. С. Хрущева новое советское руководство во главе с молодым еще генсеком Л. И. Брежневым и энергичным председателем правительства А. Н. Косыгиным взяло курс на экономические реформы с исполь-

²Всемирно известный американский экономист, Нобелевский лауреат Василий Васильевич Леонтьев (1905–1999) был выпускником Санкт-Петербургского университета. На создание класса математических моделей, носящих его имя, его вдохновил научно обоснованный баланс народного хозяйства СССР за 1923/24 годы.

³Философский словарь. — М.: Политиздат, 1954.

⁴Кибернетику — на службу коммунизму: Сб. статей / Под ред. акад. А. И. Берга. — М.; Л.: Энергия, 1961.

зованием научных методов управления и электронных вычислительных машин. В стране начался «кибернетический бум»: образованы научно-исследовательские кибернетические и экономико-математические институты, развернулось производство современных компьютеров, на всех уровнях велась разработка автоматизированных систем управления (АСУ), основаны специализированные журналы, в ряде вузов открылись новые факультеты и специальности. В частности, 1967 г. по инициативе известного философа и социолога Джермейна Михайловича Гвишиани был создан ВНИИСИ — Всесоюзный институт системных исследований (с 1992 г. ИСА — Институт системного анализа Российской академии наук). По задумке институт должен был стать советским аналогом RAND Corporation и объединить ученых различных областей.

К 1980-м годам ажиотаж вокруг АСУ стих, даже сама эта аббревиатура вышла из употребления, ее заменило словосочетание «информационная управляющая система — Management Information System (MIS)», однако суть дела от этого не изменилась. Всем стало ясно, что эффективное управление современным производством невозможно без использования компьютеров с соответствующим математическим и программным обеспечением. И хотя компьютеризация любого управленческого процесса начинается с информационно-учетных задач, на следующих этапах эволюции автоматизированных систем все более необходимыми являются средства, помогающие лицу, принимающему решения, сделать правильный выбор в той или иной ситуации. Совокупность таких средств получила название «система поддержки принятия решений — Decision Support System (DSS)». Приобретая готовую или создавая оригинальную информационную управляющую систему, нужно иметь в виду, что самым наукоемким и, как следствие, самым дорогим ее компонентом является DSS, алгоритмическую основу которой в значительной мере составляют математические методы исследования операций.

1.3. Методология операционного исследования

Независимо от того, какой конкретно выбран объект, процесс операционного исследования происходит по обобщенному алгоритму, представленному на рис. 1.2.



Рис. 1.2. Алгоритм операционного исследования

Постановка задачи

Первый и, по-видимому, самый ответственный этап — постановка задачи. На этом этапе необходимо четко выявить цель операции, обозначить границы системы, в которой будет осуществляться операция, а также определить факторы, влияющие на операцию.

Разумеется, этот процесс является творческим, здесь многое зависит от эрудиции и таланта исследователя, однако в прикладном системном анализе выработан ряд практических приемов и рекомендаций [1], которые помогают проводить целенаправленное выявление существенного в сложных реальных объектах.

Системный анализ всегда начинается с *целевыявления*, во вре-

мя которого формализуются и уточняются *истинные* цели оперирующей стороны. Нечего и говорить, что конечный успех всего дела самым существенным образом зависит от правильного целевыявления.

Вот хрестоматийный пример, приводимый во многих учебниках по исследованию операций. Руководству крупной компании, располагающейся в небоскребе, стали поступать жалобы от сотрудников на долгое ожидание лифтов в вестибюлях. Объективный хронометраж подтвердил существование проблемы: лифты старой конструкции не справлялись с потоком служащих, хотя время ожидания с точки производственной необходимости не было слишком большим. Руководство обратилось в строительную фирму, которая предложила на выбор три способа решения проблемы: 1) увеличить число лифтовых шахт; 2) заменить лифты на более быстрые; 3) установить компьютерную систему управления лифтами, оптимизирующую их движение в зависимости от времени суток, числа ожидающих на каждом этаже и т. д. Каждый из вариантов требовал миллионных затрат и долгого времени на реализацию.

Случилось так, что в момент обсуждения вариантов в кабинете оказался специалист из консалтинговой компании, который взялся разрешить проблемную ситуацию принципиально другим способом. Когда его спросили, сколько это будет стоить и сколько потребуется для этого времени, он пообещал сделать все к завтрашнему дню за тысячу долларов.

Когда наутро сотрудники пришли на работу, они увидели, что в лифтовых вестибюлях установлены удобные кресла, а на стенах развешаны зеркала. Женщины тут же устремились к ним и начали прихорашиваться, а мужчины сели в кресла и с интересом стали разглядывать крутящихся перед зеркалами женщин. И те и другие нашли себе занятие, ожидание лифта перестало быть томительным. Проблемная ситуация таким образом была легко разрешена потому, что истинной целью операции была не минимизация времени ожидания лифтов, которое объективно было вполне

приемлемым, а минимизация недовольства служащих.

Приведенный пример показывает особенность процесса целевыявления в реальных жизненных ситуациях. Для того, чтобы правильно определить цель некоторой операции, полезно проанализировать охватывающую ее операцию и т. д. При глубоком анализе проблемной ситуации истинная цель операции может оказаться далекой от той, которая виделась при поверхностном знакомстве. Эту мысль образно сформулировал один из генералов, давая характеристику находящейся при его штабе группе ученых: «Эти парни не возьмут в руки паяльника, пока не разберутся в стратегии действий союзников на тихоокеанском театре военных действий».

После того, как поставлена цель, возможен принципиально важный переход от бесконечно сложного реального объекта к конечной *системе*. Поскольку всякий реальный объект обладает необозримым множеством составляющих частей и связан с внешним миром бесчисленным количеством связей, то для того, чтобы иметь возможность исследовать его конечными методами, нужно выделить в объекте некоторое число самых главных, самых существенных *с точки зрения поставленной цели* элементов, свойств и отношений, отбросив все остальные как несущественные. Таким образом из реального мира вычленяется *система*, соответствующая цели операции; все остальное понимается как *окружающая среда*, взаимодействующая с системой через относительно небольшое число существенных связей (рис. 1.3, а).

Построение модели

Определив существенные свойства и связи, можно переходить к следующему шагу: построению модели системы на уровне черного ящика (см. рис. 1.3, б).

На этом уровне моделирования внешние связи системы делятся на две группы: входные \vec{X} , \vec{Y} и выходные \vec{Z} . По внешним связям в систему поступают воздействия (факторы), целиком определяющие ее поведение; выходные связи несут оперирующей стороне информацию о состоянии системы.

Далее, входные факторы следует подразделить на два типа:

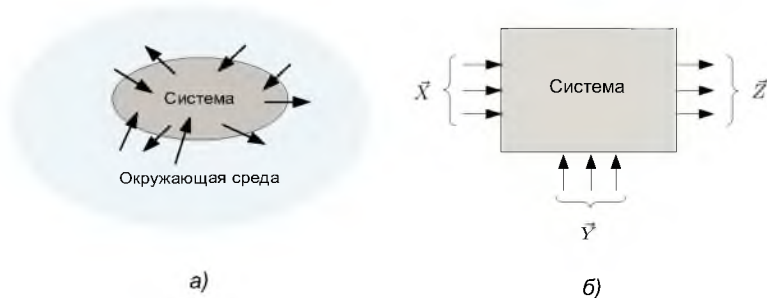


Рис. 1.3. Выделение системы из окружающей среды

управляемые (контролируемые) \vec{X} и неуправляемые (неконтролируемые) \vec{Y} .

Контролируемые факторы находятся в руках *оперирующей стороны*, то есть субъекта, осуществляющего целенаправленное управление. Он реализует это управление, принимая *решение* о выборе тех или иных значений контролируемых факторов из *множества возможных значений*. Оперирующей стороной может быть *лицо, принимающее решение (ЛПР)*, если речь идет об индивидуальном управлении, либо группа лиц, когда решения принимаются сообща.

Множество допустимых решений может быть конечным или бесконечным в зависимости от специфики системы. Например, врачу может быть доступен лишь конечный набор методов лечения, а участник дуэли может выбрать момент выстрела на непрерывном отрезке времени.

Неконтролируемые факторы являются неуправляемыми со стороны субъекта, принимающего решения. Более того, он даже может не знать, какие значения они примут в действительности. Такая неопределенность является характерной для сложных систем: ведь система является сложной именно тогда, когда результат управления в значительной мере непредсказуем, и принимать

решения приходится в условиях неопределенности, вызванной незнанием существенных неконтролируемых факторов. Классификацию типов неопределенностей и связанных с ними подходов к выбору решений мы рассмотрим чуть позже, а сейчас обратимся к выходным связям системы.

Поскольку на данном уровне моделирования система представляется «черным ящиком», то состояние системы, ее реакцию на управление можно получить только по наблюдениям выходов \vec{Z} . Зависимость $\vec{Z} = f(\vec{X}, \vec{Y})$ представляет собой модель системы на макроскопическом уровне.

Каким образом может быть найдена эта зависимость? Здесь можно идти двумя путями.

Первый путь — «просветление» черного ящика путем построения математической (аналитической) модели его работы. Подробно исследуется внутреннее содержание изучаемой системы, подбирается подходящий математический аппарат для описания ее функционирования, составляются и решаются уравнения, выводятся математические соотношения, которые в конечном счете дают возможность определить зависимость выходных факторов от входных.

Несмотря на то, что реальных объектов великое множество и все они неповторимы, в исследовании операций сформировался ряд хорошо изученных и проверенных на практике типовых математических моделей систем. Вот только некоторые из них.

- *Распределение ресурсов.* Проблема распределения ресурсов возникает, когда существует ряд видов деятельности, требующих для своего выполнения некоторых материальных или финансовых ресурсов. Типичными примерами таких операций являются управление инвестициями, производственное планирование, перевозка товаров и т. д. Управляемыми факторами являются объемы выделяемых ресурсов по видам деятельности, неуправляемыми — общие объемы доступных

ресурсов и показатели эффективности их использования в видах деятельности, выходными — суммарная прибыль или затраты. Модели распределительных операций строятся с помощью линейного, нелинейного и динамического программирования.

- *Управление проектами.* Под проектом понимается любой комплекс связанных между собой работ. Каждая работа требует определенных ресурсов (материальных, временных, трудовых, финансовых) и может быть начата только после того, как выполнены технологически предшествующие ей работы. Типичные примеры проектов — строительство сооружения, разработка компьютерной программы, создание сложного изделия. Управление проектом заключается в календарном планировании проекта при заданных внешних условиях — длительности и стоимости отдельных работ, доступном объеме ресурсов и т. п., в качестве выходных выступают обычно глобальные характеристики проекта — время исполнения и стоимость. Модели управления проектами основаны на математическом аппарате теории графов, линейного программирования, математической статистики.
- *Массовое обслуживание.* Системы массового обслуживания моделируют ситуации, когда случайный поток объектов (клиентов, требований, вызовов) обрабатывается некоторыми обслуживающими устройствами. При этом возможны очереди и отказы в обслуживании из-за ограниченности числа устройств и их производительности. Классическим примером является обработка вызовов приборами телефонной станции, похожие ситуации возникают при обслуживании потока покупателей в магазинах, потока пассажиров на транспорте, потока прерываний процессором компьютера и т. п. Управляемыми факторами могут являться количество устройств, дисциплина обслуживания, неуправляемыми — интенсивность потока заявок и время обслуживания,

выходными — вероятность отказа в обслуживании, среднее время ожидания в очереди или ее длина и т. п. Модели массового обслуживания в силу случайности исследуемых процессов широко используют математический аппарат теории вероятностей и математической статистики.

- *Управление запасами.* Управление запасами составляет существенную часть деятельности, которая получила название *логистики*. Они являются типичными для любого производстве и торговли. Соответствующие математические модели управления запасами создаются для обоснования решений по определению момента и объема заказа товаров с учетом множества неконтролируемых факторов: интенсивности спроса, задержек с поставкой, стоимости хранения, естественной убыли и порчи при хранении, убытков от дефицита и т. п. В качестве существенных выходных факторов могут выступать издержки, связанные с созданием и хранением запасов.

Второй путь — компьютерное (имитационное) моделирование [4]. Он применяется в тех случаях, когда исследуемая система слишком сложна для аналитического описания. Работа системы воспроизводится с помощью компьютерной программы, при этом нет принципиальных ограничений на детальность моделирования, она ограничивается лишь параметрами используемого компьютера. Типичными примерами имитационных моделей являются разнообразные компьютерные игры, достаточно адекватно описывающие реальные жизненные ситуации. Имитационное моделирование резко расширило практические возможности исследования операций. Имеется множество примеров, когда компьютерными программами моделировались чрезвычайно сложные сценарии операций, таких, как работа системы противовоздушной обороны страны или влияние научно-технического прогресса на развитие мировой экономики. Достигнутые результаты вдохновляют исследователей на еще более сложные модели, требующие

еще более мощных суперкомпьютеров, и этой спирали прогресса не видно предела.

В данный момент нам безразлично, каким способом построена модель, важно лишь то, что зависимость $\vec{Z} = f(\vec{X}, \vec{Y})$ определена, и от модели системы можно перейти к *модели операции*.

Целевой подход предполагает, что для оперирующей стороны состояния выходов неравноценны: одни из них предпочтительнее других. В модели операции это должно отражаться способом задания предпочтений на множестве значений выходных факторов \vec{Z} . В исследовании операций это одна из самых сложных и трудно формализуемых проблем.

Во-первых, выходные факторы могут быть не только объективно измеряемыми величинами (стоимость, длина очереди, время выполнения проекта и т. п.). Часто существенными выходными факторами являются такие, для которых нет объективных численных показателей, например удобство или красота. Их оценивание можно производить только экспертным методом, путем организации индивидуальной или коллективной *экспертизы*. Задача измерения качественных признаков является пограничной между психологией, социологией и математикой [6]. Существует ряд методов (некоторые из них мы рассмотрим в дальнейшем), которые позволяют приписать качественным значениям таких факторов численные оценки.

Во-вторых, выходных факторов может быть несколько, с точки зрения цели операции они могут даже противоречить друг другу. Например, при управлении проектом время его выполнения и стоимость, как правило, не удастся минимизировать одновременно. Приходится искать компромисс, учитывая влияние всех существенных факторов на цель операции.

В модели операции этот компромисс отображается заданием *целевой функции* (в другой терминологии — *критериальной функции, критерия*) $Q(\vec{Z})$, отображающей область значений выходных факторов \vec{Z} на естественно упорядоченное множество дей-

ствительных чисел: результат операции \vec{Z}_1 предпочтительнее или равноценен \vec{Z}_2 (обозначается $\vec{Z}_1 \lesssim \vec{Z}_2$) тогда и только тогда, когда $Q(\vec{Z}_1) \geq Q(\vec{Z}_2)$.

Правильное задание целевой функции является зачастую самым сложным и важным этапом в построении модели, ибо именно целевая функция формализует цель операции. Остальное — дело математической и вычислительной техники.

Нахождение решения Когда математическая модель операции построена, то задача нахождения наилучшего решения формально ставится как нахождение такого значения управляемых переменных \vec{X} из области допустимых значений D , которое доставляет оптимальное (наибольшее или наименьшее, исходя из смысла операции) значение целевой функции:

$$Q(\vec{Z}) = Q\left(f(\vec{X}, \vec{Y})\right) \rightarrow \max(\min)_{\vec{X} \in D}.$$

Эта задача выглядит как обычная задача оптимизации (методы ее решения мы пока обсуждать не будем, практические алгоритмы оптимизации нам еще предстоит изучить), если бы не факторы \vec{Y} . Именно наличие неконтролируемых факторов явилось тем осложняющим обстоятельством, которое во второй половине XX века вызвало разработку специфических методов принятия решений при неопределенности. Этому вопросу мы специально посвятим следующий параграф, а пока пройдем дальше по алгоритму операционного исследования (см. рис. 1.2).

Проверка, корректировка, реализация решения Итак, с помощью тех или иных математических приемов сделан выбор решения. Значит ли это, что лицо, ответственное за принятие решения, должно следовать полученным рекомендациям? По-видимому, нет. Практика показывает, что иногда решения, обоснованные безупречными математическими методами, оказываются совершенно непригодными с точки зрения опыта и здравого смысла. Это происходит потому,

что математическая модель, положенная в основу выбора решения, оказалась неадекватной истинной цели операции (вспомним историю с лифтами). Никакая самая строгая теория не может быть точнее тех приближенных предпосылок, на которых она основана, и если при выборе предпосылок для построения модели допущен просчет (это скорее правило, чем исключение, потому что реальный объект управления бывает очень сложным), то это неизбежно отразится на качестве рекомендуемого теорией решения. Поэтому процесс выбора решения по управлению сложной системой должен быть циклическим. Если при проверке решения на «разумность» результат получается отрицательным, то исходная модель или даже постановка задачи подвергается корректировке, математическими методами ищется новое оптимальное решение, и так до тех пор, пока не будет получен приемлемый для оперирующей стороны результат. Только после этого решение может быть реализовано в виде управляющих воздействий на систему.

1.4. Проблемы нахождения оптимальных решений

Задачи и математические методы нахождения оптимальных решений можно классифицировать по виду неконтролируемых факторов \vec{Y} (рис. 1.4).



Рис. 1.4. Классификация методов нахождения оптимальных решений

В наиболее простом, вырожденном, случае неконтролируемые факторы известны оперирующей стороне, с математической точки зрения это равносильно тому, что их нет вообще. В этом случае мы имеем задачу принятия оптимальных решений *при определенности*.

Один критерий Обычно предполагается, что целевая функция (критериальная функция, критерий) единственна, поэтому формально задача нахождения оптимального решения сводится к поиску экстремума функции нескольких переменных (по числу контролируемых факторов) $\vec{X} = (x_1, \dots, x_n)$ в области допустимых значений D :

$$Q(\vec{X}) \rightarrow \max_{\vec{X} \in D}(\min).$$

В прикладной математике, изучающей различные методы оптимизации, эта задача называется задачей *математического программирования*⁵.

Несмотря на простоту постановки, задачи математического программирования решаются с большим трудом. Все зависит от вида целевой функции и конфигурации области ограничений. Если целевая функция линейна, т.е. $Q(\vec{X}) = c_1x_1 + \dots + c_nx_n$, а область допустимых значений D представляет собой выпуклый многогранник в многомерном пространстве, то мы имеем задачу *линейного программирования*.

Линейное программирование играет огромную роль в исследовании операций. Множество оптимизационных задач, особенно в области экономики, либо изначально ставится в линейном виде, либо приводится к нему. Мощные методы, разработанные во второй половине XX века, прежде всего *симплексный метод*, который мы будем изучать в гл. 4, способны решить любую зада-

⁵Термин «математическое программирование» не имеет ничего общего с составлением программ для компьютеров. Он возник в послевоенные годы как обобщение понятия «линейное программирование», предложенного американским математиком Джорджем Данцигом (см. исторические замечания к гл. 2).

чу линейного программирования с большим числом переменных (десятки и сотни тысяч). Они реализованы на всех современных компьютерах и входят в золотой фонд прикладной математики. Изучению этих методов мы посвятим всю первую часть нашего курса.

К сожалению, далеко не все задачи оптимизации функции нескольких переменных имеют линейную структуру. Если целевая функция и/или ограничения нелинейны, то это — задача *нелинейного программирования*. Число различных методов нелинейного программирования огромно, это верный признак того, что задача в общем виде неразрешима и построить универсальный эффективный метод невозможно. Тем не менее для важных частных случаев *выпуклого программирования* такие методы изобретены, с рядом из них мы познакомимся во второй части курса.

Говоря об оптимальном управлении системой, мы пока ограничились случаем, когда число контролируемых переменных конечно. Однако на практике важную роль играют *бесконечномерные* задачи. Типичный пример — управление динамическим объектом (автомобилем, самолетом или космическим кораблем), движущимся из одной точки пространства в другую. Управляющие воздействия и выходы системы в этом случае представляют собой функции времени $\vec{X}(t)$, $\vec{Z}(t)$, а цель операции управления описывается *функционалом* $F[\vec{Z}(t)]$, который может представлять, например, общее количество израсходованного топлива или время в пути. Задача состоит в нахождении такого управления $\vec{X}(t)$, которое доставляет экстремум целевому функционалу при ограничениях, диктуемых законами механики. Подобные задачи изучаются в классическом *вариационном исчислении* и современной *теории оптимального управления*, удивительные успехи которой наглядно продемонстрированы в космической навигации.

Поскольку теория оптимального управления основывается на специфическом и весьма продвинутом математическом аппарате, мы оставляем решение соответствующих задач за границами нашего курса.

Несколько критериев Совершенно необычные задачи выбора решений возникают, когда цель операции невозможно описать одной целевой функцией и приходится учитывать одновременно несколько критериев:

$$Q_1(\vec{Z}), \dots, Q_n(\vec{Z}).$$

Такая ситуация имеет место, когда разные выходные факторы описывают совершенно различные стороны операции, и на этапе построения модели их затруднительно «привести к общему знаменателю». Например, если при моделировании операции покупки подержанного автомобиля в качестве существенных выходных факторов выбраны пробег, комфорт и цена, то оценки каждого из этих факторов можно понимать как самостоятельные целевые функции.

Поскольку одновременная оптимизация по нескольким, зачастую противоречивым, критериям невозможна, то выбор разумного решения в этих условиях требует специфических подходов, не сводящихся напрямую к решению задач математического программирования. Некоторые из этих подходов мы также рассмотрим в третьей части нашего курса.

Принятие решений при неопределенности В случае, когда неконтролируемые факторы оперирующей стороне неизвестны, возникает задача выбора решения *при неопределенности*. Дальнейшая классификация подходов определяется видом неопределенности. Различают три варианта:

- риск;
- неопределенность, вызванная незнанием;
- неопределенность, вызванная организованным противодействием.

Под *риском* в теории решений понимается ситуация, когда неконтролируемые факторы являются *случайными*, то есть в каждом конкретном случае выбора решения неизвестными, но распределение вероятностей которых, исходя из прошлого опыта, *заранее известно*.

Классическим примером принятия решений при риске являются разнообразные лотереи и азартные игры, давшие толчок созданию *теории полезности*, которую мы рассмотрим в соответствующем разделе курса. Более полезной с практической точки зрения является теория страхования, также основанная на теории полезности.

Второй вид представляет *неопределенность, вызванная незнанием*. Она типична для операций, в которых роль неконтролируемых факторов играют природные условия. Например, успех весеннего сева зависит, с одной стороны, от момента начала работ, который выбирается лицом, принимающим решения, а с другой стороны, от погодных условий, которые, увы, пока нами не контролируются. Такие операции иногда называют *играми с природой*.

Наконец, неопределенность может возникать в результате *организованного противодействия*. Этот случай имеет место тогда, когда неконтролируемые факторы находятся в руках разумного противника, интересы которого не совпадают с интересами оперирующей стороны. Возникает *конфликтная ситуация*. Типичными примерами конфликтов являются столкновения интересов в салонных играх, войнах, рыночной конкуренции. Математическая модель любой конфликтной ситуации называется *игрой*, а теория, описывающая правила поведения в конфликтных ситуациях, — *теорией игр*.

Несмотря на то, что проблема принятия решений при неопределенности чрезвычайно сложна, она поддается определенной формализации. Существуют некоторые общие принципы «разумного» поведения, знакомство с ними дает возможность лицу, принимающему решение, обосновать свой выбор, основываясь не на интуиции, а на точном расчете.

Глава 2

Примеры и каноническая форма задачи линейного программирования

Изучение методов оптимизации начнем с линейного программирования (ЛИ). В данной главе мы познакомимся с проблемой линейной многомерной оптимизации. Сначала рассмотрим простейшие примеры, затем сформулируем задачу в общем виде.

2.1. Задача о производственном плане

На складе фирмы имеется 50 ед. (погонных метров, квадратных футов или иных подходящих единиц измерения) дерева и 54 ед. (возможно, других) металла. Из этого сырья можно производить дачную мебель двух видов: столы и стулья. Пусть для изготовления одного стола требуется 10 ед. дерева и 6 ед. металла, а для изготовления одного стула — 5 ед. дерева и 9 ед. металла. Продажная цена одного стола составляет 8 условных денежных единиц, а одного стула — 6 у. е.

Спрашивается, какое количество столов и стульев следует из-

готовить, чтобы максимизировать доход, уложившись при этом в имеющиеся ресурсы?

Если обозначить через x_1 количество производимых столов, а через x_2 — количество стульев, то целевая функция задачи об оптимальном производственном плане ¹ будет иметь вид

$$L(x_1, x_2) = 8x_1 + 6x_2 \rightarrow \max, \quad (2.1)$$

а ограничения на объем потребляемых ресурсов запишутся двумя неравенствами

$$\begin{aligned} 10x_1 + 5x_2 &\leq 50, \\ 6x_1 + 9x_2 &\leq 54. \end{aligned} \quad (2.2)$$

Таким образом, экономическая задача об оптимальном производственном плане свелась к математической задаче нахождения максимума линейной целевой функции (2.1) двух переменных с двумя линейными же ограничениями — неравенствами (2.2).

Замечание. По смыслу данной задачи x_1 и x_2 должны быть целочисленными, однако пока мы абстрагируемся от этого требования; исходные данные в примере подобраны так, что оптимальное решение получится заведомо целочисленным. В противном случае возникает намного более трудная задача *целочисленного линейного программирования*, которой мы будем заниматься в гл. 8.

Разумеется, наш пример с двумя видами ресурсов и двумя типами продукции является сильно стилизованным. На практике номенклатура изделий может быть очень широкой, а в ограничениях приходится учитывать как сырьевые, так и другие виды ресурсов (энергию, загрузку оборудования, рабочее время исполнителей и др.). Однако суть задачи от этого не меняется, проблема составления оптимального производственного плана (произ-

¹С точки зрения ортодоксальной экономики правильнее называть ее задачей об оптимальном плане производства, поскольку понятие «производственный план» более широкое, оно включает планирование финансов, трудовых ресурсов и т. п.

водственной программы) является одной из самых актуальных в математической экономике. Отсюда, кстати, пошло название «линейное программирование».

2.2. Задача о диете

Другая классическая задача линейного программирования связана с проблемой подбора оптимального набора пищевых продуктов для составления диеты.

Для того, чтобы жить, человек должен ежедневно получать в необходимых количествах белки, жиры, углеводы, витамины, микроэлементы и прочие питательные вещества. Эти вещества попадают в организм с разнообразными пищевыми продуктами. Химический состав продуктов известен, соответствующие таблицы опубликованы в справочниках по кулинарии и диетике. Задача состоит в выборе такого набора продуктов, который, с одной стороны, обеспечивал бы потребности организма, а с другой — имел минимальную стоимость.

В качестве упрощенного, но реального примера возьмем расчет диеты по трем базовым веществам — белкам, жирам и углеводам. Их суточная потребность равна соответственно 0,1, 0,1 и 0,4 кг. Из всего многообразия пищевых продуктов выберем для примера шесть: рыбу (нежирную), растительное масло, сахар, мясо, молоко и хлеб. Содержание базовых питательных веществ в килограммах на 1 кг указанных продуктов приведено ниже. Остальное — вода, клетчатка и прочие не учитываемые в данной модели ингредиенты.

	Рыба	Масло	Сахар	Мясо	Молоко	Хлеб
Белки	0,12	0	0	0,13	0,03	0,07
Жиры	0	1	0	0,03	0,03	0
Углеводы	0	0	1	0	0,05	0,5

Для определения стоимости диеты нужно знать цены продуктов, которые, к слову сказать, постоянно меняются в сторону повышения. Здесь мы приводим округленные средние цены соседнего магазина, взятые в начале июля 2008 г. (в рублях за 1 кг).

	Рыба	Масло	Сахар	Мясо	Молоко	Хлеб
Цена	60	70	20	150	25	20

Если обозначить количество рыбы в суточной диете через x_1 , масла — x_2 и т. д., хлеба — x_6 , то задача оптимизации диеты сводится к минимизации линейной целевой функции

$$L(x_1, \dots, x_6) = 60x_1 + 70x_2 + 20x_3 + 150x_4 + 25x_5 + 20x_6 \rightarrow \min \quad (2.3)$$

при линейных ограничениях-равенствах

$$\begin{aligned} 0,12x_1 & & +0,13x_4 & +0,03x_5 & +0,07x_6 & = 0,1, \\ & +x_2 & +0,03x_4 & +0,03x_5 & & = 0,1, \\ & & +x_3 & +0,05x_5 & +0,5x_6 & = 0,4 \end{aligned} \quad (2.4)$$

и дополнительных ограничениях на неотрицательность

$$x_1 \geq 0, \dots, x_6 \geq 0. \quad (2.5)$$

Решив эту задачу в менее стилизованном виде, взяв широкий состав продуктов и учитывая остальные необходимые вещества — витамины, микроэлементы и т. д., можно ответить на жизненно важный вопрос о биологическом прожиточном минимуме при данном уровне цен.

2.3. Каноническая форма

Как видно из приведенных примеров, задача линейного программирования может быть поставлена в различных вариантах —

на максимум (2.1) или на минимум (2.3), с ограничениями в виде равенств (2.4) или нестрогих неравенств (2.2), с ограничениями на неотрицательность (2.5) или без них. Для того, чтобы не изобретать для каждого варианта собственный метод решения, договорились приводить все задачи к единообразной, так называемой *канонической*, форме.

Развернутая запись В канонической форме задача линейного программирования есть задача *минимизации* линейной целевой функции n переменных

$$L(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n \rightarrow \min \quad (2.6)$$

при m ограничениях-равенствах

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m, \end{aligned} \quad (2.7)$$

где обязательно $b_1, \dots, b_m \geq 0$, а также ограничениях на неотрицательность переменных

$$x_1 \geq 0, \dots, x_n \geq 0. \quad (2.8)$$

При использовании символов суммирования развернутая запись представляется в виде

$$L(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (2.9)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (2.10)$$

$$x_j \geq 0, \quad j = 1, \dots, n. \quad (2.11)$$

Таким образом, с алгебраической точки зрения задача линейного программирования в канонической форме есть задача нахождения такого *неотрицательного частного решения системы*

линейных уравнений (2.7), которое доставляет минимум целевой функции (2.6).

Замечание. Для того, чтобы система линейных уравнений (2.7) имела множество решений, из которых нужно выбрать оптимальное, необходимо, чтобы она была недоопределена, т. е. число линейно независимых уравнений было меньше числа переменных n . Если предположить, что все уравнения в ограничениях линейно независимы, то должно выполняться условие $m < n$.

Матричная запись Если воспользоваться матричными обозначениями, задачу линейного программирования можно записать очень компактно.

Будем считать векторы столбцами и обозначим

$\vec{X} = (x_1, \dots, x_n)^T$ — вектор переменных,

$\vec{c} = (c_1, \dots, c_n)^T$ — вектор стоимостей,

$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$ — матрица условий,

$\vec{b} = (b_1, \dots, b_m)^T$ — вектор ограничений.

Тогда задача (2.6) – (2.8) запишется в виде

$$L(\vec{X}) = \vec{c}^T \vec{X} \rightarrow \min, \quad (2.6')$$

$$A\vec{X} = \vec{b}, \quad (2.7')$$

$$\vec{X} \geq 0. \quad (2.8')$$

Векторная запись Еще один способ представления задачи линейного программирования получится, если матрицу условий A размерности $m \times n$ представить в виде строки длины n , состоящей из m -мерных векторов-столбцов:

$$A = (\vec{P}_1, \dots, \vec{P}_n),$$

где

$$\vec{P}_j = \begin{pmatrix} a_{1j} \\ \dots \\ a_{mj} \end{pmatrix}.$$

Если для единообразия вектор ограничений обозначить

$$\vec{P}_0 = \vec{b} = \begin{pmatrix} b_1 \\ \dots \\ b_m \end{pmatrix},$$

то задача (2.6') – (2.8') переписется в виде

$$L(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (2.6'')$$

$$\sum_{j=1}^n x_j \vec{P}_j = \vec{P}_0, \quad (2.7'')$$

$$x_j \geq 0, \quad j = 1, \dots, n. \quad (2.8'')$$

Приведение к канонической форме

Легко показать, что любую задачу линейного программирования можно привести к канонической форме.

Принципиально возможны следующие отклонения от канонической формы:

1) целевая функция максимизируется. В этом случае следует поменять знаки у стоимостных коэффициентов c_j ;

2) свободный член b_i в некотором i -м уравнении отрицателен. Умножаем это уравнение на -1 ;

3) i -е ограничение задано неравенством вида

$$a_{i1}x_1 + \dots + a_{in}x_n \leq b_i.$$

Для приведения данного неравенства к равенству следует ввести дополнительную неотрицательную переменную x_{n+1} , входящую в целевую функцию с нулевым стоимостным коэффициентом, т. е. $c_{n+1} = 0$. Эта переменная, прибавленная к левой части, превращает неравенство в равенство:

$$a_{i1}x_1 + \dots + a_{in}x_n + x_{n+1} = b_i.$$

Для каждого неравенства следует вводить свою дополнительную переменную;

4) i -е ограничение задано неравенством вида

$$a_{i1}x_1 + \dots + a_{in}x_n \geq b_i.$$

Для приведения его к равенству также вводим дополнительную неотрицательную переменную, которую вычитаем из левой части неравенства:

$$a_{i1}x_1 + \dots + a_{in}x_n - x_{n+1} = b_i;$$

5) некоторая переменная x_j неограничена по знаку:

$$-\infty < x_j < \infty.$$

Для приведения к канонической форме, в которой все переменные неотрицательны, представляем эту переменную в виде разности двух неотрицательных переменных:

$$x_j = x'_j - x''_j, \quad (x'_j \geq 0, x''_j \geq 0).$$

Пример. Привести к канонической форме задачу

$$\begin{aligned} L(x_1, x_2) = 2x_1 - 3x_2 &\rightarrow \max, \\ 5x_1 + 7x_2 &\geq 5, \\ x_1 - x_2 &\leq -7, \end{aligned}$$

$$x_1 \geq 0, x_2 \text{ — любое.}$$

Представляем x_2 в виде разности $x_2 - x_3$ (то есть считаем $x'_2 = x_2$, $x''_2 = x_3$, чтобы нумерация переменных шла по порядку), вводим дополнительные переменные x_4 и x_5 для устранения неравенств, меняем направление оптимизации и умножаем второе уравнение на -1 . В результате получаем

$$\begin{aligned} L(x_1, \dots, x_5) = & -2x_1 + 3x_2 - 3x_3 + 0x_4 + 0x_5 \rightarrow \min, \\ & 5x_1 + 7x_2 - 7x_3 - x_4 = 5, \\ & -x_1 + x_2 - x_3 - x_5 = 7, \\ & x_1, \dots, x_5 \geq 0. \end{aligned}$$

2.4. Исторические замечания

Задачи линейного программирования, подобные приведенным в примерах, решить обычными методами математического анализа не удастся. Производная от линейной целевой функции равна константе и к нулю ее не приравняешь. Поэтому методы линейного программирования были разработаны только в середине XX века, намного позже, чем классические приемы нахождения экстремума, опирающиеся на работы Ферма (Fermat, Pierre; 1601–1665), Лагранжа (Lagrange, Joseph Louis; 1736–1813) и других великих математиков прошлых столетий. К тому же практическая потребность решать задачи, приводящие к линейному программированию, появилась лишь тогда, когда экономика столкнулась с проблемой планирования крупномасштабных производств.

Первые работы по линейной оптимизации принадлежат выдающемуся советскому математику Леониду Витальевичу Канторовичу (1912–1986). В 1938 г. он (рис. 2.1) консультировал фанерный трест по проблеме эффективного использования лущильных станков. Канторович понял, что проблема сводится к максимизации линейной функции многих переменных при наличии ограничений в форме линейных равенств и неравенств.

Он модифицировал метод множителей Лагранжа для ее решения и осознал, что к такого рода задачам сводится множество проблем экономики. В 1939 г. опубликовал работу «Математические методы организации и планирования производства» [9], в которой описал задачи экономики, поддающиеся открытому им математическому методу. К сожалению, в то время эти пионерские результаты не были должным образом оценены из-за косности официальной советской экономической науки, которая не терпела вторжения «механистической» математики в ее святая святых — марксистско-ленинское политэкономическое учение.



Рис. 2.1.
Л. В. Канторович

Признание в своей стране пришло намного позже, в 1960-е годы, а в 1975 г. Канторович совместно с американским экономистом голландского происхождения Тьяллингом Купмансом (Koormans, Tjalling Charles; 1910–1985) был удостоен Нобелевской премия по экономике «за вклад в теорию оптимального распределения ресурсов».

На Западе отцом-основателем линейного программирования считают американского математика Джорджа Данцига (Dantzig, George Bernard; 1914–2005). В годы Второй мировой войны он (см. рис. 2.2) работал в качестве советника по математике в командовании Военно-воздушных сил и занимался программированием поставок военной техники. Слово «программирование» тогда было специфическим военным термином, оно означало составление планов и расписаний для военных учений, поставок вооружений или развертывания войсковых операций, а для компьютерного программирования использовался термин «кодирование». Данциг предложил использовать линейную модель для оптимизации планирования и изобрел универсальный численный метод решения — симплексный метод, который мы будем рассматривать в гл. 4.

До окончания войны эти результаты считались секретными и были опубликованы только в 1947 г., вызвав взрыв интереса к новой области науки, которая с подачи того же Данцига получила название «линейное программирование».



Рис. 2.2.

Джордж Данциг

На протяжении следующих десятилетий линейное программирование бурно развивалось. Во-первых, появились новые направления [14], обобщающие классическую задачу: параметрическое линейное программирование, стохастическое линейное программирование, целочисленное линейное программирование, булево линейное программирование, блочное программирование и т. д. Во-вторых, для отдельных задач (транспортная задача, задача о назначениях, задача о наибольшем потоке в сети и др.) были разработаны специализированные и чрезвычайно эффективные методы решения. Наконец, для всех поколений и типов компьютеров были созданы пакеты прикладных программ, реализующие различные методы линейного программирования.

Достижения теории и компьютерные программы дали возможность широко использовать методы линейного программирования на практике. В 1980 г. венгерский ученый Ласло Ловаш заметил: «Если бы кто-нибудь собрал статистику использования наибольшего количества машинного времени во всем мире для решения математических проблем (не считая сортировку и поиск в базах данных), то вероятно это было бы линейное программирование». По-видимому, к настоящему времени не осталось ни одной отрасли промышленности, транспорта или связи, где бы не удалось получить значительный экономический эффект от использования линейных оптимизационных моделей.

Глава 3

Повторение линейной алгебры

Цель этой главы – напомнить основные сведения из линейной алгебры, необходимые для последующего изложения теории линейного программирования.

3.1. Преобразование Жордана

Решение системы линейных уравнений

Пусть дана «квадратная» система m линейных уравнений с m неизвестными

$$\begin{array}{ccccccc} a_{11}x_1 + & \dots & + a_{1m}x_m & = & b_1, \\ & \dots & & & \dots \\ a_{m1}x_1 + & \dots & + a_{mm}x_m & = & b_m. \end{array}$$

Суть метода Гаусса – Жордана (метода полного исключения) состоит в том, чтобы с помощью равносильных преобразований (то есть таких, которые сохраняют неизменными решение) привести данную систему к диагональному виду

$$\begin{array}{ccc} x_1 & & = b'_1, \\ & \dots & \\ & & x_m = b'_m. \end{array}$$

К равносильным преобразованиям относятся:

- умножение уравнения на константу;
- сложение (вычитание) двух уравнений;
- перестановка уравнений.

Метод полного исключения состоит из m шагов, каждый из которых называется *преобразованием Жордана* или *жордановым исключением*. С помощью однократного преобразования Жордана некоторая выбранная переменная x_k (*ведущая переменная*) исключается из всех уравнений, кроме одного (*ведущего уравнения*), при этом коэффициент при ведущей переменной в ведущем уравнении становится равным единице.

Для демонстрации исключения ведущей переменной x_k выделим в системе два уравнения: некоторое текущее i -е и ведущее l -е. Коэффициент a_{lk} при ведущей переменной в ведущем уравнении отмечен особо, он играет важную роль в преобразовании Жордана и называется *ведущим коэффициентом*:

$$\begin{array}{ccccccc} a_{i1}x_1 & \cdots & +a_{ij}x_j & \cdots & +a_{ik}x_k & \cdots & = b_i & \text{— текущее уравнение,} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ a_{l1}x_1 & \cdots & +a_{lj}x_j & \cdots & +\boxed{a_{lk}}x_k & \cdots & = b_l & \text{— ведущее уравнение.} \end{array}$$

Чтобы исключить ведущую переменную x_k из текущего i -го уравнения, нужно вычесть из него ведущее, умноженное на $\frac{a_{ik}}{a_{lk}}$. При этом коэффициенты текущего уравнения преобразуются по формуле

$$a'_{ij} = a_{ij} - a_{ij} \frac{a_{ik}}{a_{lk}}, \quad j = 1, \dots, m. \quad (3.1)$$

Такое исключение нужно сделать для каждого i -го уравнения, где $i = 1, \dots, l-1, l+1, \dots, m$, а для того, чтобы коэффициент при ведущей переменной в ведущем l -м уравнении стал равным единице, ведущее уравнение нужно разделить на ведущий коэффи-

циент $\boxed{a_{lk}}$. При этом коэффициенты ведущего уравнения преобразуются по формуле

$$a'_{lj} = \frac{a_{lj}}{a_{lk}}, \quad j = 1, \dots, m. \quad (3.2)$$

Правые части уравнений преобразуются по аналогичным формулам:

$$b'_l = \frac{b_l}{a_{lk}}, \quad (3.3)$$

$$b'_i = b_i - b_l \frac{a_{ik}}{a_{lk}}, \quad i \neq l. \quad (3.4)$$

Замечание 1. При практическом решении систем уравнений методом Гаусса — Жордана сами уравнения записывать не нужно, все преобразования можно производить над числовой матрицей коэффициентов.

Замечание 2. Если правую часть системы уравнений считать дополнительным столбцом коэффициентов — нулевым или $(m + 1)$ -м, как кому нравится, — то формулы преобразования Жордана для основной матрицы и свободных членов (3.2) — (3.3) сведутся в одну. Например, если положить $a_{0i} = b_i$, $i = 1, \dots, m$, то

$$a'_{ij} = \begin{cases} a_{ij} - a_{lj} \frac{a_{ik}}{a_{lk}}, & i \neq l, \\ \frac{a_{ij}}{a_{lk}}, & i = l, \end{cases} \quad (3.5)$$

$$i = 1, \dots, m \quad j = 0, \dots, m.$$

Замечание 3. Для запоминания формул преобразования Жордана (3.5) можно воспользоваться следующим mnemonicским *правилом прямоугольника*. Если в матрице коэффициентов мысленно построить прямоугольник, одна диагональ которого соединяет преобразуемый текущий элемент $\boxed{a_{ij}}$ с ведущим $\boxed{a_{lk}}$,

то новый текущий элемент равен старому минус произведение других диагональных элементов, деленное на ведущий (рис. 3.1).

$$\begin{array}{ccccccc}
 & & & j & & k & \\
 & & & \vdots & & \vdots & \\
 i & \cdots & \boxed{a_{ij}} & \cdots & a_{ik} & \cdots & \\
 & & & \vdots & & \vdots & \\
 l & \cdots & a_{lj} & \cdots & \boxed{a_{lk}} & \cdots & \\
 & & & \vdots & & \vdots &
 \end{array}$$

Рис. 3.1. Правило прямоугольника

Замечание 4. Порядок, в котором выбираются ведущие строка и столбец при преобразовании Жордана для решения системы уравнений, не имеет значения. Единственное условие — ведущий элемент $\boxed{a_{lk}}$ не должен быть равным нулю. Естественно, каждая строка и каждый столбец должны фигурировать в списке ведущих только по одному разу.

Пример. Решим систему трех уравнений с тремя неизвестными:

$$\begin{aligned}
 2x_1 + x_2 + 3x_3 &= 13, \\
 2x_1 \quad \quad \quad x_3 &= 5, \\
 x_1 + 2x_2 + 3x_3 &= 14.
 \end{aligned}$$

Ниже приведены последовательные преобразования Жордана матрицы коэффициентов с выделенными ведущими элементами. Обработанные строки и столбцы отмечены плюсиками. Мы специально отошли от последовательного порядка выбора ведущих элементов, поэтому результирующая матрица отличается от диагональной перестановкой строк:

$$\begin{array}{ccc|c}
 2 & \boxed{1} & 3 & 13 \\
 2 & 0 & 1 & 5 \\
 1 & 2 & 3 & 14
 \end{array} \rightarrow \begin{array}{ccc|c}
 2 & 1 & 3 & 13 \\
 2 & 0 & \boxed{1} & 5 \\
 3 & 0 & -3 & -12
 \end{array} \rightarrow \begin{array}{cccc|ccc|c}
 & & & & -2 & & 0 & 1 & 0 & 2 \\
 + & -4 & 1 & 0 & 5 & & 0 & 0 & 1 & 3 \\
 \rightarrow & + & \boxed{3} & 0 & 0 & 3 & 1 & 0 & 0 & 1
 \end{array}$$

Если интерпретировать последнюю матрицу как систему уравнений, получим решение:

$$\begin{aligned}
 x_2 &= 2, \\
 x_3 &= 3, \\
 x_1 &= 1.
 \end{aligned}$$

Замечание 5 (правило Крамера). При ручном решении систем уравнений небольшой размерности (2 или 3) целесообразно не применять общий аппарат преобразований Жордана, а воспользоваться *правилом Крамера*. Согласно этому правилу, решение $X = (x_1, \dots, x_n)^T$ системы $A\vec{X} = \vec{b}$ получается в виде

$$x_i = \frac{\Delta_i}{\Delta}, \quad i = 1, \dots, n,$$

где $\Delta = \|A\|$ — определитель матрицы системы, а Δ_i — определитель матрицы, получающейся, если в матрице A i -й столбец заменить столбцом свободных членов \vec{b} .

Пример. Решим систему из предыдущего примера с помощью правила Крамера. Вычисляя соответствующие определители третьего порядка, получаем

$$\Delta = \begin{vmatrix} 2 & 1 & 3 \\ 2 & 0 & 1 \\ 1 & 2 & 3 \end{vmatrix} = 3, \quad \Delta_1 = \begin{vmatrix} 13 & 1 & 3 \\ 5 & 0 & 1 \\ 14 & 2 & 3 \end{vmatrix} = 3, \quad \Delta_2 = \begin{vmatrix} 2 & 13 & 3 \\ 2 & 5 & 1 \\ 1 & 14 & 3 \end{vmatrix} = 6,$$

$$\Delta_3 = \begin{vmatrix} 2 & 1 & 13 \\ 2 & 0 & 5 \\ 1 & 2 & 14 \end{vmatrix} = 9, \quad x_1 = \frac{\Delta_1}{\Delta} = 1, \quad x_2 = \frac{\Delta_2}{\Delta} = 2, \quad x_3 = \frac{\Delta_3}{\Delta} = 3.$$

Другие задачи линейной алгебры

Легко показать, что все элементарные равносильные преобразования матрицы $A = (a_{ij})$ эквивалентны умножению ее слева на невырожденную матрицу T (матрицу преобразования). При этом преобразуемая матрица не обязательно должна быть квадратной. Если она имеет размерность $m \times n$, то T должна быть квадратной матрицей размерности m :

$$T_{m \times m} A_{m \times n} = A'_{m \times n}.$$

Например, умножение i -й строки матрицы A на скаляр c соответствует матрице преобразования

$$T_M = \begin{pmatrix} & (i) & & & & \\ & 1 & & & & \\ & \dots & & & & \\ & & 1 & & & \\ & & & c & & \\ & & & & 1 & \\ 0 & & & & & \dots \\ & & & & & & 1 \end{pmatrix}.$$

Преобразование Жордана, как результат нескольких последовательных элементарных преобразований, задается матрицей, представляющей произведение матриц, соответствующих этим элементарным преобразованиям. Точно так же цепочка преобразований Жордана определяется матрицей, представляющей произведение матриц однократных преобразований.

Преобразование Жордана является одним из самых универсальных методов линейной алгебры. Для практических целей удобным оказывается «калькулятор Жордана» (рис. 3.2), который представляет собой программу с окном, содержащим прямоугольную таблицу необходимой размерности. В таблицу вводит-

ся преобразуемая матрица, для выбора ведущего элемента курсор устанавливается на некоторую ячейку, после чего нажатием функциональной клавиши совершается пересчет элементов матрицы по формулам преобразования Жордана (3.5).

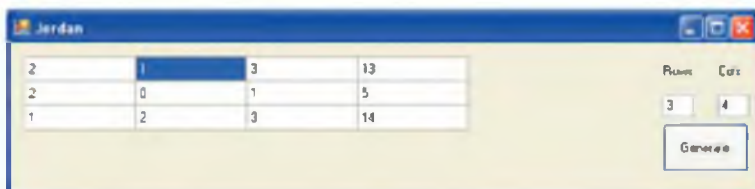


Рис. 3.2. Калькулятор Жордана. Лабораторная работа Петра Тураева

Ниже приводятся еще несколько задач линейной алгебры, которые можно решать на калькуляторе Жордана.

Обращение матриц. Для того чтобы обратить квадратную матрицу A , к ней справа приписывается единичная матрица I такой же размерности:

$$(A|I).$$

Далее с помощью серии преобразований Жордана над объединенной матрицей она приводится к виду, когда на месте матрицы A получается единичная подматрица. Для этого ведущие элементы выбираются последовательно по диагонали. В результате на том месте, где была единичная матрица, образуется A^{-1} . Действительно, если $TA = I$, то $TI = A^{-1}$. В том случае, когда строго диагональный порядок выбора ведущих элементов соблюдается не удается из-за того, что на очередном месте оказывается нуль, возможно отступление от порядка, но тогда для получения единичной подматрицы в результирующей таблице придется переставить строки.

Определение ранга матрицы. Исследуемая матрица A загружается в калькулятор и обрабатывается по диагонали до тех пор, пока это возможно, т. е. пока найдется хотя бы один ненулевой элемент среди необработанных столбцов и строк. Таким образом, результирующая матрица имеет следующий вид (символами

X обозначены несущественные элементы):

$$\left(\begin{array}{cc|ccc} 1 & 0 & X & X & X \\ & \ddots & & & \\ 0 & 1 & X & X & X \\ \hline 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{array} \right).$$

Порядок единичной матрицы, образовавшейся в верхнем левом углу, есть искомый ранг $\text{Rank}(A)$ исследуемой матрицы.

3.2. Системы линейных уравнений общего вида

Частные решения

Пусть имеется неоднородная система линейных уравнений общего вида, имеющая m уравнений и n переменных (неоднородность означает, что среди свободных членов b_i есть ненулевые):

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots & \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m. \end{aligned}$$

Любое решение x_1, \dots, x_n этой системы называется *частным*.

Первое, что необходимо сделать, — исследовать, имеет ли данная система решения (совместна ли она), и если да, то сколько частных решений она имеет.

Как известно, на вопрос о разрешимости отвечает теорема Кронекера — Капелли: *система совместна, если ранг матрицы коэффициентов системы совпадает с рангом расширенной матрицы (с добавлением вектора-столбца свободных членов)*:

$$\text{Rank}(A) = \text{Rank}(\bar{A}).$$

Первокурсники помнят, что строгое доказательство этой теоремы не очень простое. Пользуясь калькулятором Жордана, можно дать ее элементарную и наглядную интерпретацию.

Загрузим расширенную матрицу системы в калькулятор и обработаем по диагонали матрицу коэффициентов системы до тех пор, пока это возможно, то есть пока можно выбирать ненулевые ведущие элементы. Тогда итоговая матрица имеет следующий вид:

$$\left(\begin{array}{cc|ccc} 1 & & 0 & & XXX & X \\ & & \ddots & & XXX & X \\ & & & & XXX & X \\ \hline 0 & & 1 & & 0 \dots 0 & Y \\ 0 & \dots & 0 & & 0 \dots 0 & Y \\ \dots & \dots & \dots & & \dots & \dots \\ 0 & \dots & 0 & & 0 \dots 0 & Y \end{array} \right).$$

Самый правый столбец представляет преобразованный вектор свободных членов, и нас сейчас интересует именно он, вернее, его нижняя часть, обозначенная символами Y .

Если все Y равны нулю, то ранг расширенной матрицы совпадает с рангом матрицы коэффициентов, так как выполнить очередное преобразование Жордана, поместив курсор в последний столбец, невозможно. При этом нулевые строки в преобразованной матрице, если их прочесть как уравнения, имеют вид

$$0x_1 + \dots + 0x_n = 0,$$

и их можно попросту отбросить. При этом предыдущие уравнения оказываются разрешенными относительно первых $\text{Rank}(A)$ неизвестных. Следовательно, система совместна.

Если же среди элементов Y найдется хотя бы один ненулевой, скажем Y_i , то, поместив в него курсор, можно выполнить еще одно преобразование Жордана, увеличив ранг расширенной матрицы на единицу. Соответствующая строка, интерпретируемая как

уравнение, имеет вид

$$0x_1 + \dots + 0x_n = Y_i \neq 0,$$

что невозможно при любых x_1, \dots, x_n . То есть система уравнений является несовместной.

Если система совместна, то она имеет единственное решение при $\text{Rank}(A) = n$ (тогда она называется определенной) и бесчисленное множество решений при $\text{Rank}(A) < n$ (тогда она называется недоопределенной).

Базисные решения Пусть задана совместная недоопределенная система m линейных уравнений с n неизвестными, имеющая, как мы только что сказали, ранг матрицы коэффициентов $r = \text{Rank}(A) < n$.

Среди бесконечного множества частных решений этой системы особое место занимает **к о н е ч н о е** множество так называемых *базисных* решений.

Определение. *Базисным называется такое частное решение системы линейных уравнений, при котором векторы столбцы матрицы коэффициентов, соответствующие ненулевым компонентам решения, являются линейно независимыми.*

Из определения следует, что для того, чтобы получить некоторое базисное решение, нужно

- выбрать в матрице коэффициентов r линейно независимых столбцов — *базисных столбцов*. Переменные, соответствующие базисным столбцам, называются *базисными переменными*, остальные — *свободными переменными*;
- все свободные переменные (их число равно $n - r$) положить равными нулю. Это значит, что в системе остаются только базисные столбцы;

- разрешить систему относительно базисных переменных. Она будет иметь единственное решение, так как число неизвестных равно рангу матрицы коэффициентов.

Замечание 1. Если все базисные переменные x_1, \dots, x_r отличны от нуля, то такое базисное решение называется *невырожденным*.

Замечание 2. Если в системе уравнений нет лишних, то есть таких, которые получаются из остальных путем линейных комбинаций, то ранг такой системы $\text{Rank}(A)$ равен числу уравнений m , и все невырожденные базисные решения будут иметь ровно m ненулевых переменных. Поскольку в практических приложениях чаще всего происходит именно так, впредь, если нет особых указаний, мы будем полагать $\text{Rank}(A) = m$.

Замечание 3. Попробуем оценить число базисных решений. Поскольку нужно некоторым образом выбрать m базисных столбцов (в условиях предыдущего замечания) из общего числа n , то количество таких вариантов оценивается числом сочетаний

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{n(n-1)(n-2)\cdots(n-m+1)}{1 \cdot 2 \cdots m}.$$

Очевидно, это число является оценкой сверху, так как не все случайно выбранные m столбцов являются линейно независимыми.

Пример. Дана система двух уравнений с четырьмя неизвестными

$$\begin{array}{rcccc} 10x_1 & +5x_2 & +x_3 & & = 50, \\ 6x_1 & +9x_2 & & +x_4 & = 54. \end{array}$$

Здесь $m = 2$, $n = 4$, следовательно, в принципе возможны $C_4^2 = \frac{4 \cdot 3}{1 \cdot 2} = 6$ базисных решений.

На самом деле так и есть, сочетания из двух столбцов являются линейно независимыми. В таблице приведены все базисные решения этой систем; они, кстати, оказались невырожденными.

Вариант	Базисные столбцы	x_1	x_2	x_3	x_4
а	1, 2	3	4	0	0
б	1, 3	9	0	-40	0
в	1, 4	5	0	0	24
г	2, 3	0	6	20	0
д	2, 4	0	10	0	-36
е	3, 4	0	0	50	54

Для практического построения базисных решений удобно использовать калькулятор Жордана. Загрузив в него расширенную матрицу системы, нужно произвести m преобразований Жордана, определив в качестве ведущих элементов числа, стоящие в выбранных базисных столбцах. Значения базисных переменных при этом получаются в столбце свободных членов, а свободные переменные полагаются равными нулю. Например, вариант «б» из предыдущего примера получается в результате такой цепочки преобразований (обработанные строки и столбцы, как и прежде, помечены плюсиками):

$$\begin{array}{c}
 \boxed{10} \quad 5 \quad 1 \quad 0 \quad \left| \quad 50 \quad \rightarrow \quad + \quad 1 \quad 0,5 \quad 0,1 \quad 0 \quad \left| \quad 5 \\
 6 \quad 9 \quad 0 \quad 1 \quad \left| \quad 54 \quad \rightarrow \quad 0 \quad 6 \quad \boxed{-0,6} \quad 1 \quad \left| \quad 24 \quad \rightarrow \\
 \\
 \rightarrow \quad + \quad 1 \quad 1,5 \quad 0 \quad 0,167 \quad \left| \quad 9 \\
 + \quad 0 \quad -10 \quad 1 \quad -1,67 \quad \left| \quad -40
 \end{array}$$

Важно заметить, что калькулятором Жордана мы можем не просто строить, но и перебирать базисные решения. Например,

если мы хотим перейти от варианта «б» к варианту «г», отличающемуся подменой базисного столбца 3 на столбец 2, достаточно сделать еще одно преобразование заключительной таблицы предыдущего примера, выбрав в качестве ведущего элемента число, находящееся во вводимом столбце:

$$\begin{array}{cccc|cccc|c} 1 & \boxed{1,5} & 0 & 0,167 & 9 & \rightarrow & 0,67 & 1 & 0 & 0,11 & 6 \\ 0 & -10 & 1 & -1,67 & -40 & & 6,67 & 0 & 1 & -0,56 & 20 \end{array}$$

3.3. Линейные (векторные) пространства

Свойства линейного пространства

Алгебраическая модель линейного пространства используется в том случае, когда имеются некоторые объекты, которые можно смешивать друг с другом в определенных пропорциях и замещать этой смесью другие объекты.

Множество таких объектов обозначим через L , а сами объекты будем называть *векторами* и изображать буквами со стрелкой: $\vec{P} \in L$.

Математическим аналогом реальной смеси является понятие *линейной комбинации* векторов $\vec{P}_1, \dots, \vec{P}_n$:

$$\alpha_1 \vec{P}_1 + \dots + \alpha_n \vec{P}_n.$$

Таким образом, для того, чтобы строго определить линейную комбинацию, необходимо ввести две операции:

- операцию масштабирования (умножения на число), которая каждому $\vec{P} \in L$ и каждому α из некоторого внешнего множества коэффициентов K ставит в соответствие произведение $\alpha \vec{P} \in L$;
- внутреннюю операцию сложения, которая любым двум элементам $\vec{P}_1, \vec{P}_2 \in L$ ставит в соответствие их сумму $\vec{P}_1 + \vec{P}_2 \in L$.

Множество L называется линейным (или векторным) пространством, если введенные операции обладают следующими свойствами.

1. Операция масштабирования ассоциативна:

$$\alpha(\beta\vec{P}) = (\alpha\beta)\vec{P}.$$

2. Числовая единица является нейтральным элементом относительно операции масштабирования:

$$\forall \vec{P} : 1 \cdot \vec{P} = \vec{P}.$$

3. Операция сложения ассоциативна:

$$\vec{P}_1 + (\vec{P}_2 + \vec{P}_3) = (\vec{P}_1 + \vec{P}_2) + \vec{P}_3.$$

4. Операция сложения коммутативна:

$$\vec{P}_1 + \vec{P}_2 = \vec{P}_2 + \vec{P}_1.$$

5. Существует нулевой вектор $\vec{0}$ относительно сложения:

$$\forall \vec{P} : \vec{P} + \vec{0} = \vec{0} + \vec{P} = \vec{P}.$$

6. Для каждого вектора существует противоположный:

$$\forall \vec{P} \exists (-\vec{P}) : \vec{P} + (-\vec{P}) = \vec{0}.$$

7. Операции масштабирования и сложения связаны законами дистрибутивности:

$$\begin{aligned} (\alpha + \beta)\vec{P} &= \alpha\vec{P} + \beta\vec{P}, \\ \alpha(\vec{P}_1 + \vec{P}_2) &= \alpha\vec{P}_1 + \alpha\vec{P}_2. \end{aligned}$$

Простейшим, но вполне достаточным для изучения линейного программирования примером линейного пространства является *арифметическое пространство* \mathbf{R}^m , элементами которого служат векторы-столбцы действительных чисел высотой m , их линейные комбинации осуществляются путем поэлементного умножения на скаляр и сложения. Легко видеть, что все свойства линейного пространства при этом выполняются.

Линейная зависимость и независимость

Важнейшим свойством линейного пространства является возможность представления одних элементов линейной комбинацией других.

Пусть имеется совокупность векторов. Эти векторы являются *линейно зависимыми*, если какой-нибудь из них может быть представлен в виде линейной комбинации других. В противном случае данные векторы называются *линейно независимыми*.

Более формально линейную независимость можно определить следующим образом. Векторы $\vec{P}_1, \dots, \vec{P}_n$ являются *линейно независимыми*, если из них нельзя нетривиальным образом скомбинировать нулевой вектор, то есть если равенство

$$\alpha_1 \vec{P}_1 + \dots + \alpha_n \vec{P}_n = \vec{0}$$

возможно только тогда, когда одновременно все $\alpha_1, \dots, \alpha_n$ равны нулю.

Базис

Центральным понятием в теории линейных пространств является понятие базиса. Система векторов $\vec{P}_1, \dots, \vec{P}_m$ называется *базисом* линейного пространства L , если:

- она линейно независима;
- добавление любого вектора из L делает ее линейно зависимой.

Пусть $B = \{\vec{P}_1, \dots, \vec{P}_m\}$ — базис линейного пространства L . Возьмем произвольный вектор \vec{P} и добавим его к базису. Тогда

совокупность векторов $\{\vec{P}_1, \dots, \vec{P}_m, \vec{P}\}$ становится линейно зависимой и возможно нетривиальное (не все коэффициенты равны нулю) равенство

$$\alpha_1 \vec{P}_1 + \dots + \alpha_m \vec{P}_m + \alpha \vec{P} = \vec{0},$$

причем $\alpha \neq 0$, так как в противном случае нулевой вектор можно было бы скомбинировать из векторов базиса.

Тогда, поделив это выражение на α и обозначив $x_i = \frac{\alpha_i}{\alpha}$, получаем

$$x_1 \vec{P}_1 + \dots + x_m \vec{P}_m = \vec{P}. \quad (3.6)$$

Это значит, что любой вектор из L может быть представлен в виде линейной комбинации векторов базиса, или, как говорят, *разложен* по базису. Числа x_1, \dots, x_m называются *коэффициентами разложения* вектора \vec{P} по базису B . Разложение по базису единственно.

Рассмотрим m -мерное арифметическое пространство, в котором векторы представляют собой столбцы чисел высотой m . Для того, чтобы проверить систему векторов $\vec{P}_1, \dots, \vec{P}_n$ на линейную независимость, нужно составить матрицу из соответствующих столбцов и определить ее ранг. Если он равен числу векторов, то они являются линейно независимыми. Отсюда следует, что при $n > m$ совокупность любых векторов является линейно зависимой.

Для того, чтобы разложить вектор \vec{P} по базису B , нужно решить систему линейных уравнений (3.6) относительно коэффициентов разложения x_1, \dots, x_m . Если собрать эти коэффициенты в вектор \vec{X} , а базисные векторы — в матрицу B , то система уравнений (3.6) записывается в матричном виде

$$\vec{P} = B\vec{X}.$$

Пример. В качестве содержательного примера возьмем арифметическое трехмерное пространство продуктов питания в задаче о диете (см. с. 34).

Легко убедиться, что три продукта

$$\vec{P}_1 (\text{Рыба}) = \begin{pmatrix} 0,12 \\ 0 \\ 0 \end{pmatrix}, \quad \vec{P}_2 (\text{Масло}) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{P}_3 (\text{Сахар}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

образуют базис, так как определитель матрицы

$$\begin{pmatrix} 0,12 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

отличен от нуля, следовательно, ранг матрицы равен трем.

Любой другой продукт, рассматриваемый с точки зрения содержания белков, жиров и углеводов, может быть замещен некоторой смесью этих базисных продуктов. Например,

$$\vec{P}_6 = \begin{pmatrix} 0,07 \\ 0 \\ 0,5 \end{pmatrix} = 0,58\vec{P}_1 + 0 \cdot \vec{P}_2 + 0,5\vec{P}_3,$$

то есть 1 кг хлеба эквивалентен смеси 0,58 кг рыбы и 0,5 кг сахара (за вкусовые качества этой смеси математика не отвечает).

В m -мерном пространстве может быть выбрано сколько угодно базисов, причем все они содержат по m векторов. Среди них особую роль играет *единичный базис*, составленный из единичных векторов (*ортов*):

$$I = (\vec{e}_1, \dots, \vec{e}_m) = \left(\begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \right).$$

Разложение любого вектора по единичному базису тривиально:

коэффициенты разложения равны его компонентам:

$$\vec{P} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = a_1 \vec{e}_1 + \cdots + a_m \vec{e}_m.$$

Преобразование базиса Пусть в векторном пространстве задан базис $\{\vec{P}_1, \dots, \vec{P}_m\}$ и некоторый вектор \vec{P} , который, будучи разложен по данному базису, имеет коэффициенты разложения $\vec{X} = (x_1, \dots, x_m)^T$.

То есть $\vec{P} = \sum x_i \vec{P}_i$, или, если собрать векторы \vec{P}_i в матрицу $B = (\vec{P}_1, \dots, \vec{P}_m)$:

$$\vec{P} = B\vec{X}. \quad (3.7)$$

Если имеется другой базис $B' = \{\vec{P}'_1, \dots, \vec{P}'_m\}$, то тот же вектор \vec{P} можно разложить по новому базису, при этом коэффициенты разложения $\vec{X}' = (x'_1, \dots, x'_m)^T$ будут другие:

$$\vec{P} = B'\vec{X}'. \quad (3.8)$$

Формулу преобразования коэффициентов разложения можно получить, приравнявая (3.7) и (3.8):

$$B'\vec{X}' = B\vec{X}.$$

Отсюда, умножая обе части равенства слева на $(B')^{-1}$, получаем общую формулу:

$$\vec{X}' = (B')^{-1}B\vec{X}. \quad (3.9)$$

Рассмотрим два частных случая.

1. Переход от единичного базиса. Если первоначальный базис был единичным, т. е. $B = I$, то формула (3.9) упрощается:

$$\vec{X}' = (B')^{-1}I\vec{X} = (B')^{-1}\vec{X}. \quad (3.10)$$

2. Замещение одного вектора. Пусть исходный базис состоял из m векторов:

$$B = \{\vec{P}_1, \dots, \vec{P}_{l-1}, \vec{P}_l, \vec{P}_{l+1}, \dots, \vec{P}_m\}.$$

Преобразование базиса состоит в том, что один вектор \vec{P}_l замещается другим вектором \vec{P}_k :

$$B = \{\vec{P}_1, \dots, \vec{P}_{l-1}, \vec{P}_l, \vec{P}_{l+1}, \dots, \vec{P}_m\} \rightarrow$$

$$\begin{array}{l} +\vec{P}_k \\ \rightarrow \{\vec{P}_1, \dots, \vec{P}_{l-1}, \vec{P}_k, \vec{P}_{l+1}, \dots, \vec{P}_m\} = B'. \\ -\vec{P}_l \end{array}$$

Возьмем некоторый вектор \vec{P}_j и запишем его разложение по старому базису, обозначив коэффициенты разложения через x_{ij} :

$$\vec{P}_j = \sum_{i \neq l} x_{ij} \vec{P}_i + x_{lj} \vec{P}_l. \quad (3.11)$$

Вводимый вектор \vec{P}_k также разложим по старому базису:

$$\vec{P}_k = \sum_{i \neq l} x_{ik} \vec{P}_i + x_{lk} \vec{P}_l. \quad (3.12)$$

Запишем разложение \vec{P}_j по новому базису:

$$\vec{P}_j = \sum_{i \neq l} x'_{ij} \vec{P}_i + x'_{kj} \vec{P}_k. \quad (3.13)$$

Выразим \vec{P}_l из (3.12):

$$\vec{P}_l = - \sum_{i \neq l} \frac{x_{ik}}{x_{lk}} \vec{P}_i + \frac{1}{x_{lk}} \vec{P}_k. \quad (3.14)$$

Подставив (3.14) в (3.11), получим

$$\vec{P}_j = \sum_{i \neq l} \left(x_{ij} - \frac{x_{lj}x_{ik}}{x_{lk}} \right) \vec{P}_i + \frac{x_{lj}}{x_{lk}} \vec{P}_k. \quad (3.15)$$

Сравнивая (3.15) с (3.13), получаем

$$\begin{aligned} x'_{ij} &= x_{ij} - x_{lj} \frac{x_{ik}}{x_{lk}}, \quad i \neq l, \\ x'_{kj} &= \frac{x_{lj}}{x_{lk}}. \end{aligned} \quad (3.16)$$

Что напоминает эта формула? Совершенно верно — преобразование Жордана (3.5). Таким образом, существует тесная связь между решением систем линейных уравнений и преобразованиями базиса. Рассмотрим эту связь более подробно.

3.4. Решение системы линейных уравнений как процесс последовательного замещения векторов в базисе

Пусть дана квадратная система линейных уравнений

$$\begin{array}{ccccccc} a_{11}x_1 + & \dots & + a_{1m}x_m & = & b_1, \\ & \dots & \dots & & \dots \\ a_{m1}x_1 + & \dots & + a_{mm}x_m & = & b_m. \end{array}$$

Если обозначить j -й столбец матрицы коэффициентов через \vec{P}_j , а столбец свободных членов через \vec{P}_0 , то эту систему можно представить в векторном виде:

$$x_1 \vec{P}_1 + \dots + x_m \vec{P}_m = \vec{P}_0.$$

Векторы-столбцы матрицы коэффициентов линейно независимы (в противном случае система не имела бы решения) и образуют базис $B = \{\vec{P}_1, \dots, \vec{P}_m\}$.

Решить систему — значит найти коэффициенты разложения вектора \vec{P}_0 по базису B . Мы их не знаем, но зато имеем разложение вектора \vec{P}_0 по единичному базису $I = \{\vec{e}_1, \dots, \vec{e}_m\}$ — это сами компоненты вектора \vec{P}_0 :

$$b_1 \vec{e}_1 + \dots + b_m \vec{e}_m = \vec{P}_0.$$

Следовательно, перед нами стоит задача перейти от единичного базиса к базису B . Эту процедуру мы произведем за m шагов, каждый раз замещая один вектор:

$$\begin{aligned} I = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_m\} &\xrightarrow{+\vec{P}_1} \{\vec{P}_1, \vec{e}_2, \dots, \vec{e}_m\} \xrightarrow{-\vec{e}_1} \\ &\xrightarrow{+\vec{P}_2} \{\vec{P}_1, \vec{P}_2, \dots, \vec{e}_m\} \xrightarrow{-\vec{e}_2} \dots \xrightarrow{+\vec{P}_m} \{\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m\} = B. \end{aligned}$$

Таким образом, решение системы линейных уравнений методом полного исключения можно трактовать как процесс перехода от единичного базиса к базису B путем последовательного замещения единичных векторов.

Замечание. Для применения формул преобразования Жордана (3.5) нужно знать коэффициенты разложения x_{ik} включаемого вектора \vec{P}_k по текущему базису. Поскольку очередь до конкретного вектора может прийти нескоро, приходится хранить коэффициенты разложения всех векторов x_{ik} по текущему базису и преобразовывать их при переходе к новому базису.

Пример. Возьмем систему уравнений из примера на с. 46:

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &= 13, \\ 2x_1 \quad \quad \quad x_3 &= 5, \\ x_1 + 2x_2 + 3x_3 &= 14. \end{aligned}$$

Будем решать ее, делая те же самые преобразования Жордана, как и в прошлый раз, однако теперь мы интерпретируем преобразуемую матрицу как таблицу коэффициентов разложения по текущему базису. Столбцы ее соответствуют разлагаемым векторам (вектор свободных членов \vec{P}_0 мы поставили здесь на первое место), а строки — векторам текущего базиса.

На каждой итерации ведущий столбец будет соответствовать вектору, включаемому в базис, а ведущая строка — вектору, исключаемому из базиса.

Базис	\vec{P}_0	\vec{P}_1	\vec{P}_2	\vec{P}_3	→
\vec{e}_1	13	2	1	3	
\vec{e}_2	5	2	0	1	
\vec{e}_3	14	1	2	3	

$+\vec{P}_2$	Базис	\vec{P}_0	\vec{P}_1	\vec{P}_2	\vec{P}_3	→
	\vec{P}_2	13	2	1	3	
\rightarrow	\vec{e}_2	5	2	0	1	→
$-\vec{e}_1$	\vec{e}_3	-12	-3	0	-3	

$+\vec{P}_3$	Базис	\vec{P}_0	\vec{P}_1	\vec{P}_2	\vec{P}_3	→
	\vec{P}_2	-2	-4	1	0	
\rightarrow	\vec{P}_3	5	2	0	1	→
$-\vec{e}_2$	\vec{e}_3	3	3	0	0	

$+\vec{P}_1$	Базис	\vec{P}_0	\vec{P}_1	\vec{P}_2	\vec{P}_3
	\vec{P}_2	2	0	1	0
\rightarrow	\vec{P}_3	3	0	0	1
$-\vec{e}_3$	\vec{P}_1	1	1	0	0

Таким образом, разложение вектора свободных членов по заключительном базису имеет вид

$$\vec{P}_0 = 1\vec{P}_1 + 2\vec{P}_2 + 3\vec{P}_3,$$

следовательно, решение системы есть $x_1 = 1$, $x_2 = 2$, $x_3 = 3$.

3.5. Выпуклые множества в линейных пространствах

Введенное выше понятие линейного пространства является очень общим, далеко выходящим за рамки обычных геометрических представлений. Однако многие важные свойства линейных пространств становятся понятными и наглядными, если вектор $\vec{X} = (x_1, \dots, x_n)^T$ интерпретировать как геометрическую точку в пространстве L соответствующей размерности. Если размерность равна единице, то мы имеем дело с прямой, двум — с плоскостью, трем — с трехмерным геометрическим пространством, а при $n > 3$ речь идет о *гиперпространстве*.

Выпуклые комбинации и множества

Выпуклой комбинацией точек $\vec{X}_1, \dots, \vec{X}_n$ называется линейная комбинация

$$\vec{X} = \sum_{i=1}^n \alpha_i \vec{X}_i, \quad \text{где } \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i = 1.$$

В частности, когда имеются две точки \vec{X}_1 и \vec{X}_2 , то их выпуклая комбинация

$$(1 - \lambda)\vec{X}_1 + \lambda\vec{X}_2, \quad (\lambda \in [0, 1])$$

представляет собой точку на отрезке, соединяющем эти точки. При $\lambda = 0$ эта точка совпадает с \vec{X}_1 , а при $\lambda = 1$ — с \vec{X}_2 .

Выпуклой оболочкой точек называется множество всевозможных выпуклых комбинаций этих точек.

Множество $D \subset L$ называется *выпуклым*, если оно, наряду с любыми двумя своими точками, содержит их произвольную выпуклую комбинацию:

$$\forall \vec{X}_1, \vec{X}_2 \in D, \quad \forall \lambda \in [0, 1] \quad (1 - \lambda)\vec{X}_1 + \lambda\vec{X}_2 \in D.$$

Другими словами, выпуклое множество содержит весь отрезок, соединяющий любые две его точки. Выпуклое множество совпадает со своей выпуклой оболочкой.

Точка \vec{X} выпуклого множества D называется *крайней точкой*, если она не может быть представлена в виде выпуклой комбинации двух других различных точек этого множества.

Пример. Введенные выше понятия можно наглядно проиллюстрировать в двумерном пространстве, то есть на плоскости (рис. 3.3). Отрезок прямой является выпуклым множеством, у него есть две крайние точки — концы отрезка (*а*). У треугольника имеется три крайних точки — его вершины (*б*). Число крайних точек у круга бесконечно, они лежат на окружности (*в*). Фигура на рисунке (*г*) невыпукла.

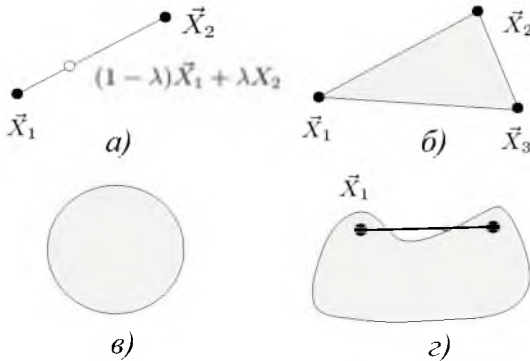


Рис. 3.3. Выпуклые и невыпуклые множества

Крайние точки выпуклых множеств обладают очень важным и полезным свойством, которое формулируется теоремой, приводимой здесь без доказательства.

Теорема (о представлении). *Всякое выпуклое замкнутое ограниченное множество есть выпуклая оболочка своих крайних точек.*

Другими словами, каждая точка выпуклого замкнутого ограниченного множества может быть представлена в виде выпуклой комбинации его крайних (некоторые авторы предпочитают термин *угловых*) точек.

Выпуклые многогранники

Среди разнообразных выпуклых множеств для нас особый интерес будут представлять геометрические фигуры, ограниченные линейными объектами (прямыми линиями на плоскости, плоскостями в трехмерном пространстве или, в общем случае, гиперплоскостями в пространстве произвольной размерности).

Гиперплоскостью называется геометрическое место (множество) точек $\vec{X} = (x_1, \dots, x_n)^T$, координаты которых удовлетворяют линейному уравнению $a_1x_1 + \dots + a_nx_n = b$ или, в векторном виде, $\vec{a}^T \vec{X} = b$.

Одно линейное неравенство вида $a_1x_1 + \dots + a_nx_n \leq b$ или $a_1x_1 + \dots + a_nx_n \geq b$ (второй вариант приводится к первому умножением на -1) определяет *полупространство*. Заметим, что это неравенство обязательно нестрогое, то есть ограничивающая гиперплоскость принадлежит полупространству.

Совокупность ограничивающих неравенств

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &\leq b_1, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &\leq b_m \end{aligned}$$

определит либо пустое множество, если неравенства противоречивы, либо замкнутый (содержащий границу) многогранный объект.

В случае ограниченности, то есть тогда, когда расстояние между любыми его двумя точками конечно, этот объект представляет

собой выпуклый многогранник. В противном случае — это выпуклая неограниченная многогранная область.

Крайними точками выпуклого многогранника являются его вершины.

Определение выпуклого многогранника системой неравенств является интуитивно понятным и наглядным. В двумерном случае его можно даже изобразить на рисунке. Однако многогранное множество можно задать и альтернативным способом, с помощью системы линейных уравнений в пространстве большей размерности с дополнительными ограничениями на неотрицательность некоторых переменных.

Действительно, одно неравенство $a_1x_1 + \dots + a_nx_n \leq b$ в пространстве размерности n эквивалентно двум соотношениям в $(n + 1)$ -мерном пространстве:

$$\begin{aligned} a_1x_1 + \dots + a_nx_n + x_{n+1} &= b, \\ x_{n+1} &\geq 0. \end{aligned}$$

Именно этот прием мы применяли при приведении задачи линейного программирования к канонической форме.

Очень полезной иногда оказывается и обратная процедура приведения ограничений-равенств к системе неравенств.

Пусть имеется набор уравнений:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m, \end{aligned}$$

причем m переменных заведомо неотрицательные (без ограничения общности будем считать, что это последние переменные $x_{n-m+1}, \dots, x_n \geq 0$).

Если $m < n$, то с помощью калькулятора Жордана данную

систему можно привести к виду

$$\begin{array}{rccccccc} a'_{11}x_1 + & \dots & + a'_{1,n-m}x_{n-m} & + x_{n-m+1} & & & = b'_1, \\ a'_{11}x_1 + & \dots & + a'_{1,n-m}x_{n-m} & & + x_{n-m+2} & & = b'_2, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a'_{m1}x_1 + & \dots & + a'_{m,n-m}x_{n-m} & & & + x_n & = b'_m. \end{array}$$

Эти уравнения эквивалентны следующим неравенствам в пространстве размерности $n - m$:

$$\begin{array}{rccccccc} a'_{11}x_1 + & \dots & + a'_{1,n-m}x_{n-m} & \leq b'_1, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a'_{m1}x_1 + & \dots & + a'_{m,n-m}x_{n-m} & \leq b'_m. \end{array}$$

В частности при $n = m + 2$ таким способом можно привести систему многомерных ограничений к виду, когда ее можно изобразить на плоскости.

Пример. Имеется совокупность ограничений

$$\begin{array}{rcccccc} 16x_1 & +14x_2 & +x_3 & +x_4 & = & 104, \\ 4x_1 & -4x_2 & +x_3 & -x_4 & = & -4, \\ & & & & & x_1, \dots, x_4 \geq 0. \end{array}$$

После обработки матрицы системы уравнений калькулятором Жордана получаем эквивалентную систему ограничений

$$\begin{array}{rcccccc} 10x_1 & +5x_2 & +x_3 & & = & 50, \\ 6x_1 & +9x_2 & & +x_4 & = & 54, \\ & & & & & x_1, \dots, x_4 \geq 0. \end{array}$$

Отбрасывая неотрицательные переменные x_3, x_4 , приходим к неравенствам

$$\begin{array}{rcccccc} 10x_1 & +5x_2 & \leq & 50, \\ 6x_1 & +9x_2 & \leq & 54, \\ & & & & & x_1, x_2 \geq 0. \end{array}$$

Для визуализации каждого из этих ограничений следует построить соответствующую отсекающую прямую и выбрать ту полуплоскость, которая определяется неравенством. Здесь можно дать следующие практические рекомендации.

Во-первых, для построения прямой удобно привести ее уравнение к виду уравнения в отрезках:

$$\frac{x_1}{a} + \frac{x_2}{b} = 1.$$

Эта прямая отсекает на координатных осях соответствующие отрезки a и b (рис. 3.4).

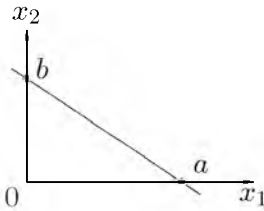


Рис. 3.4. Уравнение прямой в отрезках

Во-вторых, для выбора полуплоскости проще всего подставить в уравнение $x_1 = x_2 = 0$ и посмотреть, выполняется ли соответствующее неравенство. Если да, то выбирается полуплоскость, содержащая начало координат, если нет, — то альтернативная.

Для нашего примера уравнения отсекающих прямых в отрезках имеют вид

$$\frac{x_1}{5} + \frac{x_2}{10} = 1,$$

$$\frac{x_1}{9} + \frac{x_2}{6} = 1.$$

Выбирая соответствующие полуплоскости и учитывая ограничения на неотрицательность переменных, видим, что совокупность ограничений определяет выпуклый четырехугольник, выделенный на рис. 3.5 жирными линиями. Заметим, что именно этот набор условий присутствовал в задаче о производственном плане (см. с. 32).

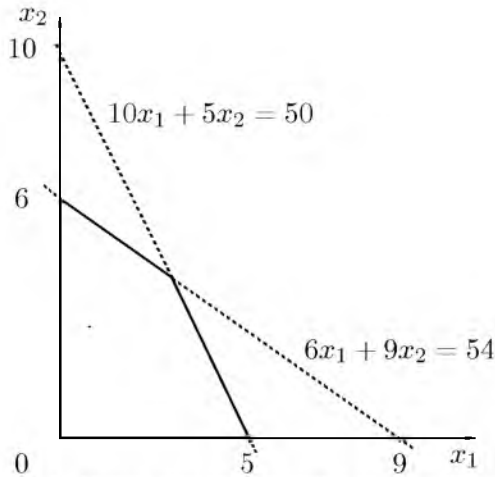


Рис. 3.5. Визуализация линейных ограничений

Отделимость выпуклых множеств

Выпуклые множества обладают еще одним фундаментальным свойством, которое кажется интуитивно очевидным (см. рис. 3.6), но строгое доказательство которого для произвольных линейных пространств представляет непростую математическую задачу.

Теорема (о разделяющей гиперплоскости). Если в линейном пространстве заданы два непересекающихся выпуклых множества K_1 и K_2 , то существует разделяющая их гиперплоскость, т. е.

$$\exists \vec{a} \quad \forall \vec{X}_1 \in K_1, \vec{X}_2 \in K_2 \\ \vec{a}^T \vec{X}_1 < \vec{a}^T \vec{X}_2.$$

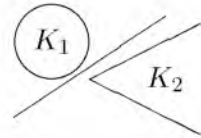


Рис. 3.6. Выпуклые множества на плоскости разделяются прямой линией

Для случая, когда множества K_1 и K_2 касаются друг друга, имеется вариант этой теоремы, называемый *теоремой об опорной гиперплоскости*.

3.6. Исторические замечания

Классический метод решения систем линейных уравнений, описанный в данной главе, носит имя двух великих математиков прошлого — Карла Гаусса и Камилла Жордана (рис. 3.7 и 3.8).



Рис. 3.7. Карл Гаусс



Рис. 3.8. Камилл Жордан

Карл Гаусс (Gauss, Johann Carl Friedrich; 1777–1855) — выдающийся немецкий математик, астроном и физик, иностранный член Петербургской академии наук. Воспитывался в небогатой семье. С его именем связаны фундаментальные исследования почти во всех основных областях математики: алгебре, дифференциальной и неевклидовой геометрии, в математическом анализе, теории функций комплексного переменного, теории вероятностей, а также в астрономии, геодезии и механике. Гаусса называли «королем математиков» XIX века. Несколько учеников Гаусса стали выдающимися математиками, например Риман, Дедекин, Бессель, Мёбиус.

Камилл Жордан (Jordan, Marie Ennemond Camille; 1838–1922) — французский математик, по образованию был инженером. Родился в Лионе, учился в Политехнической школе, потом там преподавал. Основные научные результаты: жорданова нормальная форма в линейной алгебре; мера Жордана в математическом анализе; теорема Жордана — Гёльдера в теории групп.

Глава 4

Симплексный метод

Вооружившись математическим аппаратом линейной алгебры, мы готовы приступить к изучению методов решения задач линейного программирования.

Мы уже говорили, что исторически первым, самым универсальным и мощным вычислительным методом является симплексный метод, который был разработан в 1947 г. американским математиком Джоржем Данцигом. Его название происходит от слова «симплекс», которое обозначает геометрическую фигуру, являющуюся n -мерным обобщением треугольника. Симплекс — это выпуклая оболочка $n + 1$ точек, не лежащих в n -мерной гиперплоскости. Таким образом, одномерный симплекс представляет собой отрезок, двумерный — треугольник, трехмерный — тетраэдр и т. д.

Как все великое, симплексный метод очень прост. Удивительно, почему его не придумали сто лет назад. Впрочем, вы сможете убедиться в этом сами, читая эту главу.

4.1. Графическая интерпретация задачи линейного программирования

Начнем с наглядных геометрических построений. В качестве простейшего примера рассмотрим задачу о производственном

плане с двумя переменными и двумя ограничениями, описанную в гл. 2 (см. с. 32):

$$\begin{aligned} L(x_1, x_2) &= 8x_1 + 6x_2 \rightarrow \max, \\ 10x_1 + 5x_2 &\leq 50, \\ 6x_1 + 9x_2 &\leq 54, \\ x_1, x_2 &\geq 0. \end{aligned}$$

Для графического решения задачи возьмем координатную плоскость (рис.4.1) и определим сначала множество точек (x_1, x_2) , удовлетворяющих *ограничениям* — *constraints* задачи. Каждая такая точка называется *допустимым решением* — *feasible decision* или *планом*. Термин «план» возник как раз применительно к задаче о производственном плане, но впоследствии был распространен и на все другие задачи линейного программирования.

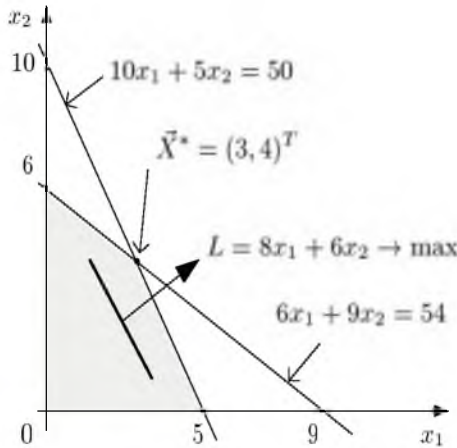


Рис. 4.1. Графическая интерпретация задачи о производственном плане

Как мы видели в примере на с. 71, множество планов — *feasible region* в нашей задаче представляет собой часть плоскости, ограниченную выпуклым четырехугольником. Задача состоит в нахождении такого плана, который обеспечивает оптимум (в нашем случае максимум) *целевой функции* — *objective function* $L(x_1, x_2)$.

Функцию, заданную на точках плоскости, обычно отображают набором изолиний. У нас это будут линии равных значений целевой функции — изоцелевые линии. Так как целевая функция линейна, то изоцелевые линии, определяемые уравнением

$$L(x_1, x_2) = c_1 x_1 + c_2 x_2 = \text{const},$$

представляют собой семейство параллельных прямых, угол наклона которых определяется отношением коэффициентов c_1 и c_2 . С увеличением const прямая смещается вправо, что показано стрелкой.

Таким образом, для графического решения на максимум нужно построить одну из изоцелевых прямых и перемещать ее параллельно самой себе вправо до тех пор, пока она пересекается с областью планов. Точка последнего касания даст *оптимальный план*.

В нашем примере оптимальный план соответствует крайней точке четырехугольника, образованной пересечением двух прямых

$$\begin{aligned} 10x_1 + 5x_2 &= 50, \\ 6x_1 + 9x_2 &= 54. \end{aligned}$$

Решая эту систему уравнений, находим $x_1^* = 3$, $x_2^* = 4$.

Приведенный нами пример является чрезвычайно простым, однако, основываясь на нем, можно сделать ряд фундаментальных наблюдений.

1. Множество планов задачи линейного программирования является выпуклым многогранником с конечным числом крайних точек, либо выпуклой неограниченной многогранной областью.

2. Оптимальный план, если он существует, находится в крайней точке множества планов. Если изоцелевая гиперплоскость совпадает с одной из граней многогранника, то все крайние точки, лежащие на этой грани, имеют равное значение целевой функции.

3. Задача может не иметь решения только в двух случаях:

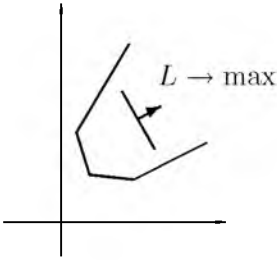


Рис. 4.2. Отсутствие решения по причине неограниченности целевой функции

общего случая их следует строго доказать, чем мы сейчас и займемся.

а) когда множество планов пусто, т.е. условия задачи противоречивы;

б) когда множество планов неограничено в сторону оптимизации (рис. 4.2), поэтому целевая функция также неограничена сверху или снизу.

Разумеется, приведенные рассуждения являются только наблюдениями для частного случая двух переменных. Для

4.2. Свойства планов задачи линейного программирования

Пусть имеется задача линейного программирования в канонической форме в матричной или векторной записи:

$$\begin{aligned}
 L(\vec{X}) = \vec{c}^T \vec{X} \rightarrow \min, & & L(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j \rightarrow \min, \\
 A\vec{X} = \vec{b}, & & \sum_{j=1}^n x_j \vec{P}_j = \vec{P}_0, \\
 \vec{X} \geq 0, & & x_j \geq 0.
 \end{aligned}$$

Свойство 1. *Множество планов задачи линейного программирования выпукло.*

Доказательство. Пусть \vec{X}_1 и \vec{X}_2 — планы. Это значит, что

$$\begin{aligned} A\vec{X}_1 &= \vec{b}, \vec{X}_1 \geq 0, \\ A\vec{X}_2 &= \vec{b}, \vec{X}_2 \geq 0. \end{aligned}$$

Возьмем выпуклую комбинацию планов \vec{X}_1, \vec{X}_2

$$\vec{X} = (1 - \lambda)\vec{X}_1 + \lambda\vec{X}_2, \quad 0 \leq \lambda \leq 1,$$

и проверим ее на удовлетворение условиям задачи:

$$\begin{aligned} A\vec{X} &= A[(1 - \lambda)\vec{X}_1 + \lambda\vec{X}_2] = \\ &= (1 - \lambda)A\vec{X}_1 + \lambda A\vec{X}_2 = (1 - \lambda)\vec{b} + \lambda\vec{b} = \vec{b}, \\ \vec{X} &= \underbrace{(1 - \lambda)}_{\geq 0} \underbrace{\vec{X}_1}_{\geq 0} + \underbrace{\lambda}_{\geq 0} \underbrace{\vec{X}_2}_{\geq 0} \geq 0. \end{aligned}$$

Таким образом, \vec{X} является планом. □

Свойство 2. *Оптимальный план находится в крайней точке множества планов.*

Доказательство. Допустим противное, т. е. предположим, что оптимальный план \vec{X}^* — не крайняя точка. Тогда по теореме о представлении она является выпуклой комбинацией крайних точек \vec{X}_k множества планов D :

$$\vec{X}^* = \sum \alpha_k \vec{X}_k, \quad \text{где } \alpha_k \geq 0, \sum \alpha_k = 1.$$

Тогда

$$\begin{aligned} L(\vec{X}^*) &= \vec{c}^T \vec{X}^* = \vec{c}^T \sum \alpha_k \vec{X}_k = \\ &= \sum \alpha_k \vec{c}^T \vec{X}_k = \sum \alpha_k L(\vec{X}_k) \geq \sum \alpha_k \min_k L(\vec{X}_k) = \\ &= \min_k L(\vec{X}_k) \sum \alpha_k = \min_k L(\vec{X}_k). \end{aligned}$$

Мы получили, что значение целевой функции в оптимальной точке не лучше, чем в одной из крайних точек. Полученное противоречие доказывает утверждение. \square

Итак, мы установили, что и в многомерном случае оптимальный план нужно искать среди крайних точек множества планов. На плоской картинке их видно сразу, но как найти крайние точки у многогранника, заданного системой уравнений и ограничений на неотрицательность? Геометрические соображения здесь не помогут, необходимо найти чисто алгебраический способ.

Подсказать идею нам поможет условная диаграмма (рис. 4.3). На ней схематически изображено бесконечное множество всех

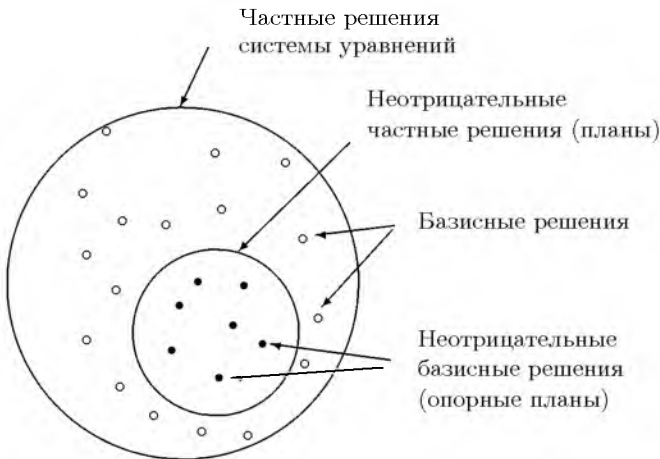


Рис. 4.3. Классификация частных решений системы уравнений частных решений \vec{X} системы уравнений $A\vec{X} = b$, в котором выделено также бесконечное подмножество неотрицательных частных решений, то есть планов.

Среди частных решений имеется к о н е ч н о е множество базисных решений (они отмечены кружками). Те базисные решения,

которые попадают в неотрицательное подмножество, называются *опорными планами*; именно они представляют для нас особый интерес.

Вспоминая понятие базисного решения (с. 52), дадим следующее определение опорного плана:

Определение. План $\vec{X} = (x_1, \dots, x_n)^T$ задачи линейного программирования называется *опорным*, если векторы-столбцы матрицы A , соответствующие ненулевым компонентам вектора \vec{X} , являются линейно независимыми.

Из определения следует, что в опорном плане имеется не более r ненулевых компонент, где r равно рангу матрицы A . Если считать, что в условиях задачи нет линейно зависимых уравнений, то ранг равен числу уравнений m , что мы и будем предполагать в дальнейшем. Тогда если в опорном плане имеется ровно m ненулевых компонент, то он является невырожденным, в противном случае — вырожденным.

Пример. Среди шести базисных решений системы линейных уравнений для задачи о производственном плане (см. пример на с. 53) имеется четыре неотрицательных:

Вариант	Базисные столбцы	x_1	x_2	x_3	x_4
а	1, 2	3	4	0	0
в	1, 4	5	0	0	24
г	2, 3	0	6	20	0
е	3, 4	0	0	50	54

Интересно посмотреть, каким точкам выпуклого многогранника соответствуют эти четыре опорных плана. Если привести систему уравнений к неравенствам и отбросить две последние переменные, то можно воочию убедиться, что опорные планы являются крайними точками множества планов (см. рис.4.1).

Это частное наблюдение является чрезвычайно важным. Проиллюстрированное для пространства двух переменных, оно за-

служивает корректного доказательства для общего многомерного случая.

Свойство 3. План \vec{X} является крайней точкой множества планов тогда и только тогда, когда он опорный.

Доказательство необходимости. Пусть имеется опорный план $\vec{X} = (x_1, \dots, x_n)^T$. Докажем, что он является крайней точкой.

Рассуждаем от противного. Допустим, что он не является крайней точкой. Тогда найдутся два такие различные векторы \vec{X}_1 и \vec{X}_2 , что

$$\vec{X} = (1 - \lambda)\vec{X}_1 + \lambda\vec{X}_2, \quad \text{где } 0 \leq \lambda \leq 1. \quad (4.1)$$

Так как план \vec{X} опорный, то в нем имеется не более m ненулевых компонент. Без ограничения общности можно считать их стоящими на первых местах, то есть план имеет следующую структуру:

$$\vec{X} = (x_1, \dots, x_m, 0, \dots, 0)^T,$$

причем по определению опорного плана векторы $\vec{P}_1, \dots, \vec{P}_m$ линейно независимы.

Из (4.1) следует, что \vec{X}_1 и \vec{X}_2 имеют такую же структуру:

$$\vec{X}_1 = (x_1^{(1)}, \dots, x_m^{(1)}, 0, \dots, 0),$$

$$\vec{X}_2 = (x_1^{(2)}, \dots, x_m^{(2)}, 0, \dots, 0).$$

Поскольку \vec{X}_1 и \vec{X}_2 — планы, то

$$\sum_{i=1}^m x_i^{(1)} \vec{P}_i = \vec{P}_0,$$

$$\sum_{i=1}^m x_i^{(2)} \vec{P}_i = \vec{P}_0.$$

Вычтем из первого уравнения второе:

$$\sum_{i=1}^m [x_i^{(1)} - x_i^{(2)}] \vec{P}_i = \vec{0}.$$

Поскольку векторы $\vec{P}_1, \dots, \vec{P}_m$ линейно независимы, такое равенство возможно, если выражения в квадратных скобках одновременно равны нулю. Следовательно, $x_i^{(1)} = x_i^{(2)}$ и предположение о том, что векторы \vec{X}_1 и \vec{X}_2 различны, ложно. Полученное противоречие доказывает утверждение. \square

Доказательство достаточности. Пусть $\vec{X} = (x_1, \dots, x_n)^T$ — крайняя точка множества планов D . Покажем, что она является опорным планом, то есть векторы, соответствующие ненулевым x_i , линейно независимы.

Пусть ненулевыми являются первые k компонент вектора

$$\vec{X} = (x_1, \dots, x_k, 0, \dots, 0)^T.$$

Тогда, поскольку он является планом,

$$\sum_{i=1}^k x_i \vec{P}_i = \vec{P}_0. \quad (4.2)$$

Будем рассуждать от противного. Предположим, что векторы $\vec{P}_1, \dots, \vec{P}_k$ линейно зависимы. Тогда возможно нетривиальное равенство

$$\sum_{i=1}^k \alpha_i \vec{P}_i = \vec{0}, \quad (4.3)$$

в котором хотя бы один из коэффициентов α_i не равен нулю.

Зададимся некоторым $d > 0$ и умножим на него обе части

равенства (4.3). Прибавляя и вычитая результат из (4.2), имеем

$$\sum_{i=1}^k (x_i - d) \vec{P}_i = \vec{P}_0,$$

$$\sum_{i=1}^k (x_i + d) \vec{P}_i = \vec{P}_0.$$

Поскольку $x_i > 0$, то число d можно выбрать настолько малым, что все выражения в скобках будут положительными.

Таким образом, два различных вектора

$$\vec{X}_1 = (x_1 - d, \dots, x_k - d, 0, \dots, 0)^T,$$

$$\vec{X}_2 = (x_1 + d, \dots, x_k + d, 0, \dots, 0)^T$$

являются планами, при этом план \vec{X} является их выпуклой комбинацией:

$$\vec{X} = \frac{1}{2} \vec{X}_1 + \frac{1}{2} \vec{X}_2.$$

Это противоречит допущению о том, что \vec{X} — крайняя точка. Следовательно, допущение о линейной зависимости векторов $\vec{P}_1, \dots, \vec{P}_k$ оказалось ложным, и план \vec{X} является опорным. Свойство доказано полностью. \square

Итак, мы установили взаимно однозначное соответствие между крайними точками и опорными планами. Этот фундаментальный вывод открывает следующий путь решения задачи линейного программирования. Следуя ему, для нахождения оптимального плана нужно:

- найти все базисные решения соответствующей системы линейных уравнений;
- выбрать из них неотрицательные;
- подставляя выбранные решения в целевую функцию, найти наилучшее.

Пример. Продолжим предыдущий пример (с. 79) и применим этот подход к задаче о производственном плане. Подставляя четыре найденных неотрицательных базисных решения плана в целевую функцию, видим, что оптимальным является вариант «а».

Вариант	x_1	x_2	x_3	x_4	L
а	3	4	0	0	48
в	5	0	0	24	40
г	0	6	20	0	36
е	0	0	50	54	0

К сожалению, для реальных задач этот путь тупиковый. Дело в том, что на практике размерность матрицы условий бывает очень большая – сотни и тысячи строк и столбцов. Соответственно число опорных планов, оцениваемое числом сочетаний C_n^m , астрономически велико, уже при $n = 100$ и $n = 20$ $C_n^m \approx 10^{20}$. Поэтому сплошной перебор крайних точек невозможен, необходимо искать пути сокращения перебора.

4.3. Теория симплексного метода

В данном разделе мы рассмотрим математические основы симплекс-метода. Будем предполагать, что задача линейного программирования приведена к канонической форме и записана в векторном виде:

$$\begin{aligned}
 L(x_1, \dots, x_n) &= \sum_{j=1}^n c_j x_j \rightarrow \min, \\
 \sum_{j=1}^n x_j \vec{P}_j &= \vec{P}_0, \\
 x_j &\geq 0.
 \end{aligned}
 \tag{4.4}$$

Общая идея

Большинство методов оптимизации сводится к последовательному перебору некоторого числа вариантов. Чтобы его сократить, перебор нужно каким-то образом упорядочить. В различных конкретных методах способы упорядочения перебора свои, однако все они в той или иной степени основываются на трех общих приемах.

- *Ограничение перебора.* Состоит в том, что следующий анализируемый вариант выбирается не где попало, а в некоторой связи с предыдущим.
- *Направленность перебора.* При переборе целевая функция должна все время улучшаться. Таким образом, каждый раз отсекается множество заведомо неперспективных вариантов.
- *Критерий оптимальности.* Неоценимую роль для любого метода играет критерий оптимальности, который позволяет распознать в текущем варианте оптимальный без перебора всех оставшихся.

В симплексном методе задействованы все три приема, что делает его необычайно эффективным.

Ограничение перебора

Применительно к линейному программированию этот прием реализуется следующим образом.

Мы знаем, что каждой крайней точке соответствует опорный план – неотрицательное базисное решение системы уравнений; таким образом, перебор крайних точек сводится к перебору базисов.

В симплексном методе ограничение перебора состоит в том, что новый базис получается из текущего замещением всего одного вектора.

Предположим, что включаемый вектор \vec{P}_k нам известен. Спрашивается, какой вектор следует исключить из базиса? Любой или конкретный \vec{P}_l ? Исследуем этот вопрос подробнее.

Итак, пусть имеется текущий опорный план. Для упрощения рассуждений мы полагаем план невырожденным, а ненулевые (положительные) компоненты считаем стоящими на первом месте:

$$\vec{X} = (x_1, \dots, x_l, \dots, x_m, 0, \dots, 0)^T.$$

Векторы, соответствующие этим положительным компонентам, образуют текущий базис $B = \{\vec{P}_1, \dots, \vec{P}_m\}$. Поскольку \vec{X} – план, то

$$x_1 \vec{P}_1 + \dots + x_m \vec{P}_m = \vec{P}_0. \quad (4.5)$$

Попытаемся ввести вектор \vec{P}_k в базис. Для этого разложим его по текущему базису (коэффициенты разложения будут нумероваться двойными индексами: первый – индекс базисного вектора, второй – разлагаемого вектора):

$$\vec{P}_k = x_{1k} \vec{P}_1 + \dots + x_{mk} \vec{P}_m.$$

Это выражение можно переписать в виде

$$-x_{1k} \vec{P}_1 - \dots - x_{mk} \vec{P}_m + \vec{P}_k = \vec{0}. \quad (4.6)$$

Возьмем некоторое число $\theta > 0$, умножим на него равенство (4.6) и прибавим к (4.5):

$$(x_1 - \theta x_{1k}) \vec{P}_1 + \dots + (x_l - \theta x_{lk}) \vec{P}_l + \dots + (x_m - \theta x_{mk}) \vec{P}_m + \theta \vec{P}_k = \vec{P}_0. \quad (4.7)$$

Выражение (4.7) представляет собой разложение \vec{P}_0 по системе векторов $\vec{P}_1, \dots, \vec{P}_m, \vec{P}_k$. Если число θ очень мало, то все коэффициенты разложения будут положительными, следовательно, они образуют план

$$\vec{X}' = ((x_1 - \theta x_{1k}), \dots, (x_l - \theta x_{lk}), \dots, (x_m - \theta x_{mk}), \theta, 0, \dots, 0)^T. \quad (4.8)$$

План \vec{X}' не является опорным, так как содержит $m + 1$ ненулевую компоненту (система из $m + 1$ вектора не может быть линейно независимой).

Будем постепенно увеличивать значение θ и посмотрим, что произойдет с компонентами плана \vec{X}' . Те скобки, где $x_{ik} \leq 0$, останутся положительными при любых θ , а те, где $x_{ik} > 0$, будут уменьшаться. Наконец наступит момент, когда одна из скобок, скажем l -я, обратится в нуль. Это произойдет при $x_l - \theta x_{lk} = 0$, откуда

$$\theta = \theta_0 = \frac{x_l}{x_{lk}} = \min_{x_{ik} > 0} \frac{x_i}{x_{ik}}. \quad (4.9)$$

Таким образом, при $\theta = \theta_0$ образуется новый опорный план

$$\vec{X}' = (x'_1, \dots, x'_{l-1}, 0, x'_{l+1}, \dots, x'_m, x'_k, 0, \dots, 0)^T,$$

который будет содержать ровно m ненулевых компонент. Согласно (4.8), при $\theta = \theta_0$

$$\begin{aligned} x'_i &= x_i - \frac{x_l}{x_{lk}} x_{ik}, \quad i \neq k, \\ x'_k &= \frac{x_l}{x_{lk}}. \end{aligned}$$

Легко заметить, что эти формулы являются хорошо известными формулами преобразования Жордана.

Таким образом, при известном векторе, включаемом в базис, выбор вектора, исключаемого из базиса, predetermined правилом (4.9).

Замечание 1. Если все $x_{ik} \leq 0$, то число θ можно увеличивать неограниченно, и ни одна из переменных в новом плане не обнуляется. Это значит, что можно построить план \vec{X}' со сколь угодно большими компонентами, то есть множество планов является неограниченной выпуклой областью. В дальнейшем мы покажем, что это свидетельствует о неразрешимости задачи по причине неограниченности целевой функции.

Замечание 2. Преобразование базиса путем замещения одного вектора имеет наглядную геометрическую интерпретацию. Оно соответствует перемещению от вершины выпуклого многогранника планов к соседней вершине по соединяющему их ребру.

Пример. Продемонстрируем переход от одного опорного плана к соседнему на примере задачи о производственном плане. Совокупность ограничений для нее в канонической форме имеет вид

$$\begin{aligned} 10x_1 + 5x_2 + x_3 &= 50, \\ 6x_1 + 9x_2 + x_4 &= 54. \end{aligned}$$

Возьмем исходный опорный план $\vec{X} = (0, 0, 50, 54)^T$ из предыдущего примера на с. 83 (вариант «е»). Ему соответствует базис $B = \{\vec{P}_3, \vec{P}_4\}$.

Для того, чтобы ввести некоторый вектор, скажем \vec{P}_1 , в базис, разложим его по текущему базису. Это сделать очень просто, поскольку базис B единичный:

$$\vec{P}_1 = \begin{pmatrix} 10 \\ 6 \end{pmatrix} = x_{31}\vec{P}_3 + x_{41}\vec{P}_4 = 10\vec{P}_3 + 6\vec{P}_4.$$

Чтобы определить, какой вектор следует исключить из базиса, воспользуемся правилом (4.9), имея в виду, что оба коэффициента разложения x_{31} и x_{41} положительны:

$$\frac{x_l}{x_{l1}} = \frac{x_3}{x_{31}} = \min_{i \in \{3,4\}} \frac{x_i}{x_{i1}} = \min \left(\frac{x_3}{x_{31}}, \frac{x_4}{x_{41}} \right) = \min \left(\frac{50}{10}, \frac{54}{6} \right) = 5.$$

Как видим, минимум достигается при $l = 3$, следовательно, из текущего базиса нужно исключить вектор \vec{P}_3 . Тогда новый базис будет $B' = \{\vec{P}_1, \vec{P}_4\}$. Ему соответствует новый опорный план $\vec{X}' = (x'_1, 0, 0, x'_4)^T$, ненулевые компоненты которого можно

вычислить по формулам преобразования Жордана (4.5) и (4.6):

$$x'_1 = \frac{x_l}{x_{lk}} = 5,$$

$$x'_4 = x_4 - x_{4l} \frac{x_l}{x_{lk}} = 54 - 6 \cdot 5 = 24.$$

Таким образом, новый план есть $\vec{X}' = (5, 0, 0, 24)^T$. Возвращаясь к примеру на с. 83, видим, что мы перешли к опорному плану «в». Рассматривая многоугольник планов на с. 74, убеждаемся, что новая крайняя точка лежит на одном ребре с предыдущей.

Направленность перебора Поскольку выбор вектора, исключаемого из базиса, предопределен предыдущими рассуждениями, остается одна степень свободы – выбор вектора, подлежащего включению в базис. Это нужно сделать так, чтобы целевая функция при переходе от текущего опорного плана \vec{X} к новому опорному плану \vec{X}' улучшилась (уменьшилась).

Напомним формулы преобразования компонентов плана:

$$x'_i = x_i - \frac{x_l}{x_{lk}} x_{ik}, \quad i \neq k, \quad (4.10)$$

$$x'_k = \frac{x_l}{x_{lk}}. \quad (4.11)$$

Значение целевой функции для старого плана

$$L(\vec{X}) = \sum_{i \neq l} c_i x_i + c_l x_l.$$

Запишем целевую функцию для нового опорного плана:

$$L(\vec{X}') = \sum_{i \neq l} c_i x'_i + c_k x'_k.$$

Подставим в нее компоненты плана \vec{X}' из (4.10) – (4.11) и выполним очевидные преобразования:

$$\begin{aligned}
L(\vec{X}') &= \sum_{i \neq l} c_i \left(x_i - x_{ik} \frac{x_l}{x_{lk}} \right) + c_k \frac{x_l}{x_{lk}} = \\
&= \left[\text{раскроем скобки, прибавим и вычтем } c_l x_l = c_l x_l \frac{x_{lk}}{x_{lk}} \right] = \\
&= \sum_{i \neq l} c_i x_i - \frac{x_l}{x_{lk}} \sum_{i \neq l} c_i x_{ik} + c_k \frac{x_l}{x_{lk}} + c_l x_l - c_l x_l \frac{x_{lk}}{x_{lk}} = \\
&= \underbrace{\sum_{i \neq l} c_i x_i + c_l x_l}_{L(\vec{X})} - \frac{x_l}{x_{lk}} \underbrace{\left(\sum_{i \neq l} c_i x_{ik} + c_l x_{lk} - c_k \right)}_{z_k} = \\
&= L(\vec{X}) - \frac{x_l}{x_{lk}} (z_k - c_k) = L(\vec{X}) - \theta_0 (z_k - c_k), \quad (4.12)
\end{aligned}$$

где введено обозначение (пока чисто формально)

$$z_k = \sum_{i=1}^m c_i x_{ik}. \quad (4.13)$$

Поскольку $\theta_0 > 0$, то возможность улучшения целевой функции полностью зависит от знака величины $(z_k - c_k)$. Для того чтобы в наибольшей степени уменьшить целевую функцию, следует в базис ввести такой вектор \vec{P}_k , для которого выражение $(z_k - c_k)$ не только положительно, но и максимально:

$$(z_k - c_k) = \max_j (z_j - c_j). \quad (4.14)$$

Величина z_j представляет собой значение целевой функции, которое получится, если в нее подставить коэффициенты разложения вектора \vec{P}_j по текущему базису. Хотя она была введена формально, ее можно содержательно интерпретировать. Сделаем это на примере задачи о диете.

Предположим, имеется текущий базисный набор продуктов, например

$$B = \{ \vec{P}_1(\text{Рыба}), \vec{P}_2(\text{Масло}), \vec{P}_3(\text{Сахар}) \}.$$

Тогда любой продукт, представленный вектором \vec{P}_j и не входящий в базис, может быть заменен эквивалентным набором базисных продуктов, например (см. пример на с. 58):

$$\vec{P}_6(\text{Хлеб}) = x_{16}\vec{P}_1 + x_{26}\vec{P}_2 + x_{36}\vec{P}_3 = 0,58\vec{P}_1 + 0 \cdot \vec{P}_2 + 0,5\vec{P}_3,$$

то есть 1 кг хлеба по содержанию белков, жиров и углеводов эквивалентен смеси 0,58 кг рыбы и 0,5 кг сахара. Если подставить соответствующие цены базисных продуктов, мы получим стоимость эквивалентного набора, замещающего 1 кг хлеба:

$$z_6 = c_1x_{16} + c_2x_{26} + c_3x_{36} = 60 \cdot 0,58 + 70 \cdot 0 + 20 \cdot 0,5 = 44,8 \text{ руб.}$$

Эту величину логично назвать эквивалентной стоимостью хлеба в текущем базисе.

В данной интерпретации критерий выбора (4.14) становится совершенно прозрачным: в базисный набор выгодно включить такой продукт P_j , у которого истинная стоимость c_j меньше эквивалентной z_j , и эта разница максимальна.

Критерий оптимальности

Из предыдущих рассуждений напрашивается вывод: если все $(z_j - c_j) \leq 0$, то никакое замещение вектора в текущем базисе не сможет уменьшить значение целевой функции, значит, текущий опорный план является оптимальным. Следующая теорема делает это предположение истинным утверждением.

Теорема. Если для опорного плана

$$\vec{X}^* = (x_1^*, \dots, x_m^*, 0, \dots, 0)^T$$

справедливы неравенства $z_j - c_j \leq 0$, $j = 1, \dots, n$, то этот план является оптимальным.

Доказательство. Обозначим базисные векторы, соответствующие плану \vec{X}^* , через \vec{P}_i , индекс $i = 1, \dots, m$ будет использоваться при суммировании по базису.

Пусть $\vec{X} = (x_1, \dots, x_n)^T$ – произвольный план. Поскольку \vec{X}^* и \vec{X} – планы, то

$$\sum_{i=1}^m x_i^* \vec{P}_i = \vec{P}_0, \quad (4.15)$$

$$\sum_{j=1}^n x_j \vec{P}_j = \vec{P}_0. \quad (4.16)$$

Если разложить каждый из векторов \vec{P}_j по текущему базису

$$\vec{P}_j = \sum_{i=1}^m x_{ij} \vec{P}_i \quad (4.17)$$

и подставить коэффициенты разложения в целевую функцию, получится набор эквивалентных стоимостей векторов \vec{P}_j в текущем базисе

$$z_j = \sum_{i=1}^m c_i x_{ij}, \quad j = 1, \dots, n, \quad (4.18)$$

которые по условию теоремы не более истинных стоимостей:

$$z_j \leq c_j, \quad j = 1, \dots, n. \quad (4.19)$$

Для начала подставим (4.17) в (4.16) и изменим порядок суммирования:

$$\sum_{j=1}^n x_j \sum_{i=1}^m x_{ij} \vec{P}_i = \sum_{i=1}^m \left(\sum_{j=1}^n x_j x_{ij} \right) \vec{P}_i = \vec{P}_0. \quad (4.20)$$

Сравним (4.20) и (4.15). Поскольку разложение по базису единственно, то

$$x_i^* = \sum_{j=1}^n x_j x_{ij}. \quad (4.21)$$

Далее, так как все $z_j \leq c_j$, то для целевой функции произвольного плана

$$L(\vec{X}) = \sum_{j=1}^n c_j x_j \geq \sum_{j=1}^n z_j x_j.$$

Подставляя сюда z_j из (4.18), меняя порядок суммирования и учитывая (4.21), имеем

$$\begin{aligned} L(\vec{X}) &\geq \sum_{j=1}^n z_j x_j = \sum_{j=1}^n \left(\sum_{i=1}^m c_i x_{ij} \right) x_j = \\ &= \sum_{i=1}^m c_i \left(\sum_{j=1}^n x_j x_{ij} \right) = \sum_{i=1}^m c_i x_i^* = L(\vec{X}^*). \end{aligned}$$

То есть план \vec{X}^* действительно является оптимальным. \square

Случай неограниченности целевой функции

Возможен случай, когда при переходе к новому базису ни один старый вектор исключить не удастся, потому что при некотором $(z_k - c_k) > 0$ все $x_{ik} \leq 0$, $i = 1, \dots, m$. Докажем, что тогда задача не имеет решения.

Действительно, как мы показали, в этом случае можно построить план (4.8), содержащий $m + 1$ положительную компоненту:

$$\vec{X}' = ((x_1 - \theta x_{1k}), \dots, (x_l - \theta x_{lk}), \dots, (x_m - \theta x_{mk}), \theta, 0, \dots, 0)^T,$$

где θ — сколь угодно большое положительно число.

Значение целевой функции для этого плана равно

$$\begin{aligned} L(\vec{X}') &= \sum_{i=1}^m c_i (x_i - \theta x_{ik}) + c_k \theta = \sum_{i=1}^m c_i x_i - \theta \underbrace{\sum_{i=1}^m c_i x_{ik}}_{z_k} + c_k \theta = \\ &= L(\vec{X}) - \theta(z_k - c_k). \end{aligned}$$

Поскольку $(z_k - c_k) > 0$, то при $\theta \rightarrow \infty$ $L(\vec{X}') \rightarrow -\infty$, то есть задача не имеет решения по причине неограниченности целевой функции в сторону оптимизации (в канонической форме целевая функция минимизируется).

**Схема
симплексного
метода**

Собирая воедино отдельные шаги процесса перебора крайних точек, получаем следующий алгоритм симплексного метода.

П о д г о т о в и т е л ь н ы й э т а п

Задача приводится к канонической форме.

Выбирается исходный базис $B_0 = \{\vec{P}_1, \dots, \vec{P}_m\}$, строится исходный опорный план $\vec{X}_0 = (x_1, \dots, x_m, 0, \dots, 0)^T$.

О т д е л ь н а я и т е р а ц и я

Шаг 1. Текущий опорный план проверяется на оптимальность. Для этого:

а) все векторы \vec{P}_j разлагаются по базису:

$$\vec{P}_j = \sum_{i=1}^m x_{ij} \vec{P}_i;$$

б) вычисляются их эквивалентные стоимости

$$z_j = \sum_{i=1}^m c_i x_{ij};$$

в) если все $(z_j - c_j) \leq 0$, $j = 1, \dots, n$, то задача решена. Текущий опорный план является оптимальным. Если нет, то переход на шаг 2.

Шаг 2. Определяется вектор \vec{P}_k , подлежащий включению в базис. Для него

$$(z_k - c_k) = \max_j (z_j - c_j).$$

Шаг 3. Определяется вектор \vec{P}_l , подлежащий исключению из базиса. Для него

$$\frac{x_l}{x_{lk}} = \min_{x_{ik} > 0} \frac{x_i}{x_{ik}}, \quad i = 1, \dots, m.$$

Если все $x_{ik} \leq 0$, то задача не имеет решения по причине неограниченности целевой функции.

Шаг 4. Производится преобразование базиса и пересчет опорного плана по формулам преобразования Жордана (4.10) – (4.11) с ведущим элементом x_{lk} . Возврат на шаг 1.

В результате нескольких итераций будет получен оптимальный план.

Замечание 1. Хотя скорость сходимости симплексного метода не гарантирована, и этим он вызывает подозрения у чистых математиков, на практике число итераций получается небольшим, как правило, от m до $2m$.

Замечание 2. Может ли симплексный метод заикнуться? Теоретически может, но этот случай крайне маловероятен, в литературе описано всего несколько таких экзотических примеров. И хотя для борьбы с возможным заикливанием разработаны специальные методы – антициклины, – их практическая ценность не слишком велика, в нашем не слишком математизированном курсе их рассматривать не будем.

4.4. Практический алгоритм симплексного метода

Вычисления по симплексному методу удобно производить с помощью так называемой *симплексной таблицы* (см. табл. 4.1).

Симплексная таблица состоит из собственно табличной части, выделенной двойными линиями, и вспомогательного оформления,

Таблица 4.1. Симплексная таблица

i	Ба- зис	c_i	План \vec{X}	c_1	\dots	c_j	\dots	c_k	\dots	c_n
				\vec{P}_1	\dots	\vec{P}_j	\dots	\vec{P}_k	\dots	\vec{P}_n
1	\vec{P}_1	c_1	x_1	x_{11}	\dots	x_{1j}	\dots	x_{1k}	\dots	x_{1n}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
i	\vec{P}_i	c_i	x_i	x_{i1}	\dots	x_{ij}	\dots	x_{ik}	\dots	x_{in}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
l	\vec{P}_l	c_l	x_l	x_{l1}	\dots	x_{lj}	\dots	x_{lk}	\dots	x_{ln}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
m	\vec{P}_m	c_m	x_m	x_{m1}	\dots	x_{mj}	\dots	x_{mk}	\dots	x_{mn}
$m+$ $+1$			L	$z_1 -$ $-c_1$	\dots	$z_i -$ $-c_i$	\dots	$z_k -$ $-c_k$	\dots	$z_n -$ $-c_n$

которое создает дополнительные удобства при ручных расчетах, в компьютерном варианте оно не используется.

Табличная часть имеет $m + 1$ строку (нумерация от 1 до $m + 1$) и $n + 1$ столбец (нумерация от 0 до n).

Первые m строк являются основными, они соответствуют базисным векторам и содержат коэффициенты разложения x_{ij} векторов \vec{P}_j , $j = 1, \dots, n$ по текущему базису. При этом нулевой столбец хранит базисные компоненты x_1, \dots, x_m текущего плана, которые, как известно, также равны коэффициентам разложения вектора \vec{P}_0 по текущему базису: $x_{i0} = x_i$.

Дополнительная $(m + 1)$ -я строка является управляющей. Ее нулевой элемент содержит текущее значение целевой функции: $x_{m+1,0} = L(\vec{X})$, а остальные соответствуют значениям $z_j - c_j$ для каждого из векторов P_j : $x_{m+1,j} = z_j - c_j$.

П о д г о т о в и т е л ь н ы й э т а п

На данном этапе производится первоначальное заполнение симплексной таблицы. Пока мы будем предполагать, что матрица ограничений задачи в канонической форме содержит единичную подматрицу. Такая ситуация возникает, в частности, когда система неравенств вида \leq приводится к каноническим равенствам.

Тогда исходный базис будет естественным единичным базисом $B_0 = I = \{\vec{e}_1, \dots, \vec{e}_m\}$, и первоначальное заполнение основной части симплексной таблицы сведется к копированию в нее матрицы условий.

После заполнения основной части производится вычисление элементов $m + 1$ строки:

$$x_{m+1,0} = L = \sum_{i=1}^m c_i x_i; \quad x_{m+1,j} = z_j - c_j = \sum_{i=1}^m c_i x_{ij} - c_j, \quad j = 1, \dots, n.$$

О т д е л ь н а я и т е р а ц и я

Шаг 1. Сканируется управляющая $(m + 1)$ -я строка (кроме нулевого элемента). Если все $x_{m+1,j} = z_j - c_j \leq 0$, то текущий опорный план $\vec{X}^* = (x_1^*, \dots, x_m^*, 0, \dots, 0)^T$ является оптимальным. Если нет, то переход на следующий шаг.

Шаг 2. Определяется индекс ведущего столбца k . Ему соответствует максимальный положительный элемент управляющей строки, не считая нулевого ее элемента: $z_k - c_k = \max_j (z_j - c_j)$.

Шаг 3. Сканируется ведущий столбец, не считая $m + 1$ -го элемента. Определяется индекс ведущей строки l . Ей соответствует

$$\frac{x_l}{x_{lk}} = \min_{x_{ik} > 0} \frac{x_i}{x_{ik}}.$$

Если же все $x_{ik} \leq 0$, то задача не имеет решения по причине неограниченности целевой функции.

Шаг 4. Таблица пересчитывается калькулятором Жордана относительно ведущего элемента x_{lk} . При этом, как оказывается, новые значения L' и $(z'_j - c_j)$ в $m + 1$ -й управляющей строке можно не вычислять заново, а пересчитывать по тем же формулам преобразования Жордана.

Сначала докажем это для L' . Как следует из (4.12):

$$L' = L - \theta_0(z_k - c_k) = L - \frac{x_l}{x_{lk}}(z_k - c_k).$$

Если вспомнить правило прямоугольника, то можно увидеть, что $L' = x_{m+1,0}$ действительно пересчитывается по этому правилу.

Для $(z_j - c_j)$ доказательство будет чуть длиннее. По определению

$$\begin{aligned} z_j &= \sum_{i=1}^m c_i x_{ij} = \sum_{i \neq l} c_i x_{ij} + c_l x_{lj}, \\ z'_j &= \sum_{i \neq l} c_i x'_{ij} + c_k x'_{kj}. \end{aligned} \quad (4.22)$$

При переходе к новому базису коэффициенты разложения векторов \vec{P}_j пересчитываются по формулам Жордана (3.5):

$$\begin{aligned} x'_{ij} &= x_{ij} - x_{lj} \frac{x_{ik}}{x_{lk}}, \quad i \neq l, \\ x'_{kj} &= \frac{x_{lj}}{x_{lk}}. \end{aligned}$$

Подставляя их в (4.22), получаем

$$\begin{aligned}
(z'_j - c_j) &= \sum_{i \neq l} c_i x'_{ij} + c_k x'_{kj} - c_j = \\
&= \sum_{i \neq l} c_i \left(x_{ij} - x_{lj} \frac{x_{ik}}{x_{lk}} \right) + c_k \frac{x_{lj}}{x_{lk}} - c_j = \\
&= \sum_{i \neq l} c_i x_{ij} - \frac{x_{lj}}{x_{lk}} \sum_{i \neq l} c_i x_{ik} + c_k \frac{x_{lj}}{x_{lk}} - c_j = \\
&= \left[\text{прибавим и вычтем } c_l x_{lj} = \frac{x_{lj}}{x_{lk}} c_l x_{lk} \right] = \\
&= \underbrace{\sum_{i \neq l} c_i x_{ij} + c_l x_{lj}}_{z_j} - \frac{x_{lj}}{x_{lk}} \underbrace{\left(\sum_{i \neq l} c_i x_{ik} + c_l x_{lk} - c_k \right)}_{z_k} - c_j = \\
&= (z_j - c_j) - \frac{x_{lj}}{x_{lk}} (z_k - c_k).
\end{aligned}$$

Если посмотреть на симплексную таблицу, можно убедиться, что элементы $(z'_j - c_j) = x_{m+1,j}$ также пересчитываются по правилу прямоугольника.

Пример. Решим симплексным методом задачу о производственном плане.

Каноническая форма задачи имеет вид

$$\begin{aligned}
L(x_1, x_2, x_3, x_4) &= -8x_1 - 6x_2 + 0x_3 + 0x_4 \rightarrow \min \\
10x_1 + 5x_2 + x_3 &= 50, \\
6x_1 + 9x_2 + x_4 &= 54, \\
x_1, \dots, x_4 &\geq 0.
\end{aligned}$$

Подготовительный этап. Матрица условий содержит единичную подматрицу, следовательно, имеется естественный единичный базис $B = \{\vec{P}_3, \vec{P}_4\}$, поэтому исходный опорный план $\vec{X}_0 = (0, 0, 50, 54)^T$.

Заполняем начальную симплексную таблицу (табл. 4.2):

Таблица 4.2. Исходная симплексная таблица

i	Ба- зис	c_i	План \vec{X}	-8	-6	0	0
				\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4
1	\vec{P}_3	0	50	10	5	1	0
2	\vec{P}_4	0	54	6	9	0	1
3			0	8	6	0	0

И т е р а ц и я 1. Этот план не оптимальный, так как в управляющей строке есть положительные элементы.

Сканируем эту строку, находим максимальный элемент, равный 8. Он находится в 1-м столбце (нумерация столбцов от 0), следовательно, в базис включаем вектор \vec{P}_1 . Этот столбец становится ведущим.

Сканируем ведущий столбец с целью определения вектора, исключаемого из базиса. Находим

$$\theta_0 = \min \left(\frac{50}{10}, \frac{54}{6} \right) = 5.$$

Выводим из базиса вектор \vec{P}_3 . Пересчитываем таблицу с ведущим элементом x_{31} , вспомогательное оформление упрощаем (табл. 4.3).

Таблица 4.3. Симплексная таблица после первой итерации

	\vec{X}	\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4
\vec{P}_1	5	1	$\frac{1}{2}$	$\frac{1}{10}$	0
\vec{P}_4	24	0	6	$-\frac{3}{5}$	1
	-40	0	2	$-\frac{4}{5}$	0

И т е р а ц и я 2. План опять не оптимальный. Сканируем

управляющую строку. Вводим в базис \vec{P}_2 . Сканируем ведущий столбец:

$$\theta_0 = \min \left(\frac{5}{0,5}, \frac{24}{6} \right) = 4.$$

Выводим из базиса вектор \vec{P}_4 . Пересчитываем таблицу с ведущим элементом x_{42} (табл. 4.4).

Таблица 4.4. Симплексная таблица после второй итерации

	\vec{X}	\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4
\vec{P}_1	3	1	0	$\frac{3}{20}$	$-\frac{1}{12}$
\vec{P}_2	4	0	1	$-\frac{1}{10}$	$\frac{1}{6}$
	-48	0	0	$-\frac{3}{5}$	$-\frac{1}{3}$

Так как в управляющей строке нет положительных элементов, то задача решена. Оптимальный план $\vec{X}^* = (3, 4, 0, 0)^T$, оптимальное значение целевой функции $L^* = -48$. Размер прибыли получился отрицательным, так как при переходе к канонической форме мы сменили знаки у стоимостных коэффициентов. В целом результат тот же, что и при графическом решении.

Замечание 1. Легко видеть, что симплексный метод для решения задачи линейного программирования очень похож на метод Гаусса — Жордана для решения систем линейных уравнений. Все отличие в том, что в симплексном методе на каждой итерации ведущие столбец и строка выбираются не произвольно, а в соответствии с дополнительными расчетами.

4.5. Метод искусственного базиса

Если матрица условий не содержит естественного единичного базиса, то начальный опорный план можно построить методом искусственного базиса. Кроме того, этот метод дает возможность

проверить, не имеет ли задача решения по причине противоречивости условий.

Пусть исходная задача имеет вид

$$\begin{aligned} L = c_1x_1 + \dots + c_nx_n &\rightarrow \min, \\ a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots &\dots \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m, \\ x_1 \geq 0, \dots, x_n &\geq 0. \end{aligned}$$

Образуюм расширенную задачу, введя дополнительно m неотрицательных *искусственных* переменных:

$$\begin{aligned} \tilde{L} = c_1x_1 + \dots + c_nx_n + \$x_{n+1} + \dots + \$x_{n+m} &\rightarrow \min, \\ a_{11}x_1 + \dots + a_{1n}x_n + x_{n+1} &= b_1, \\ \dots &\dots \\ a_{m1}x_1 + \dots + a_{mn}x_n + x_{n+m} &= b_m, \\ x_j \geq 0, \quad j = 1, \dots, m+n. \end{aligned}$$

Стоимостные коэффициенты при искусственных переменных, которые мы обозначили символом $\$$, мы пока не определяем, но полагаем, что они представляют собой достаточно большие положительные числа, заведомо большие, чем c_j .

Легко видеть, что если $\vec{X} = (x_1, \dots, x_n)^T$ — план исходной задачи, то $\vec{\tilde{X}} = (x_1, \dots, x_n, 0, \dots, 0)^T$, в котором все искусственные переменные равны нулю, является планом расширенной задачи. Если же исходная задача не имеет планов, то любой план расширенной задачи будет содержать хотя бы одну ненулевую искусственную переменную.

Расширенная задача имеет искусственный единичный базис $B_0 = \{\vec{P}_{n+1}, \dots, \vec{P}_{n+m}\}$, которому соответствует начальный опорный план $(0, \dots, 0, b_1, \dots, b_m)^T$.

Для того чтобы избежать фиксирования значения $\$,$ воспользуемся такой интерпретацией. Стоимостные коэффициенты, стоящие при основных и искусственных переменных, можно считать имеющими разную природу. Например, c_1, \dots, c_n трактовать как обычные цены, измеряемые, скажем, в рублях, а $\$$ — цена в один доллар, причем курс доллара предполагается очень большим и заранее не фиксированным. Тогда целевая функция будет иметь рублевую и долларовую составляющие: $L = L_1 + \$L_2$. Точно так же устроены величины $z_j - c_j = (z_j - c_j)_1 + \$(z_j - c_j)_2$, поскольку z_j представляет собой результат подстановки коэффициентов разложения вектора \vec{P}_j в целевую функцию.

Так как в симплексной таблице L и $(z_j - c_j)$ хранятся в $(m + 1)$ -й строке, то эту строку мы расщепляем на две: $(m + 1)$ -ю и $(m + 2)$ -ю, причем в $(m + 1)$ -й строке будут содержаться рублевые, а в $(m + 2)$ -й — долларовые составляющие этих величин:

$$\begin{aligned} x_{m+1,0} &= L_1, \quad x_{m+2,0} = L_2, \\ x_{m+1,j} &= (z_j - c_j)_1, \quad x_{m+2,j} = (z_j - c_j)_2, \quad j = 1, \dots, n. \end{aligned}$$

Практический алгоритм симплексного метода с искусственным базисом состоит из подготовительного и двух основных этапов.

На подготовительном этапе составляется и заполняется симплексная таблица, содержащая $n + m + 1$ столбец и $m + 2$ строки. Последние m столбцов содержат искусственный базис. Рублевые и долларовые составляющие стоимостей L и $z_j - c_j$ заносятся соответственно в $(m + 1)$ -ю и $(m + 2)$ -ю строки.

На первом этапе производятся обычные итерации симплексного метода, причем в качестве управляющей используется «долларовая» $(m + 2)$ -я строка. Признаком окончания этапа является выполнение соотношений $x_{m+2,j} \leq 0, j = 1, \dots, n + m$.

Первый этап может завершиться двояким образом. Нормальное окончание предполагает, что все искусственные векторы выведены из базиса, полученный оптимальный план \vec{X}^* есть план

исходной задачи, он является начальным планом для следующего этапа.

При аварийном окончании некоторые искусственные векторы не удается вывести из базиса, это означает, что исходная задача не имеет решения по причине отсутствия планов из-за противоречивости условий.

Чтобы узнать, чем завершился этап, проще всего посмотреть долларовую составляющую целевой функции $L_2 = x_{m+2,0}$. Если она не равно нулю, то это значит, что в заключительном плане остались искусственные переменные, задача не имеет решения. При $x_{m+2,0} = 0$ осуществляется переход ко второму этапу.

На втором этапе симплексная таблица обрезается: отбрасываются $(m+2)$ -я строка и все столбцы искусственного базиса $\vec{P}_{n+1}, \dots, \vec{P}_{m+n}$. Затем итерации продолжаются, но в качестве управляющей теперь используется «рублевая» $(m+1)$ -я строка.

Пример. Дана задача линейного программирования:

$$\begin{aligned} L(\vec{X}) &= -x_1 - 4x_2 - x_3 \rightarrow \min, \\ x_1 - x_2 + x_3 &= 3, \\ 2x_1 - 5x_2 - x_3 &= 0, \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Поскольку в матрице условий нет единичной подматрицы, применяем метод искусственного базиса. Расширенная задача имеет вид

$$\begin{aligned} L(\vec{X}) &= -x_1 - 4x_2 - x_3 + \$x_4 + \$x_5 \rightarrow \min, \\ x_1 - x_2 + x_3 + x_4 &= 3, \\ 2x_1 - 5x_2 - x_3 + x_5 &= 0, \\ x_1, \dots, x_5 &\geq 0. \end{aligned}$$

Заполняем симплексную таблицу, содержащую искусственный базис (табл. 4.5).

Таблица 4.5. Исходная симплексная таблица

i	Ба- зис	c_i	План \vec{X}	-1	-4	-1	\$	\$
				\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4	\vec{P}_5
1	\vec{P}_4	\$	3	1	-1	1	1	0
2	\vec{P}_5	\$	0	2	-5	-1	0	1
3	руб.		0	1	4	1	0	0
4	\$		3	3	-6	0	0	0

На первом этапе в качестве управляющей используем 4-ю – «долларовую» – строку. Она содержит положительные элементы, следовательно, текущий план не оптимальный. Находим наибольший элемент, он выделен в табл. 4.5 жирным шрифтом. Соответствующий ему вектор \vec{P}_1 вводим в базис. Определяем вектор, исключаемый из базиса:

$$\min \left(\frac{3}{1}, \frac{0}{2} \right) = 0$$

(0 не должен нас смущать, так как он не в знаменателе, а в числителе). Выводим из базиса \vec{P}_5 . Совершаем преобразование Жордана относительно отмеченного ведущего элемента. Получаем таблицу (табл. 4.6):

Таблица 4.6. Симплексная таблица после первой итерации

	\vec{X}	\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4	\vec{P}_5
\vec{P}_4	3	0	1,5	1,5	1	-0,5
\vec{P}_1	0	1	-2,5	-0,5	0	0,5
руб.	0	0	6,5	1,5	0	-0,5
\$	3	0	1,5	1,5	0	-1,5

План опять не оптимальный. Вводим в базис вектор \vec{P}_2 (а можно и \vec{P}_3 , так как максимумов два). Поскольку в ведущем столбце, не считая управляющих строк, имеется единственный положительный элемент, то вариантов нет — выводим \vec{P}_4 . Делаем преобразование Жордана (табл. 4.7):

Таблица 4.7. Симплексная таблица после второй итерации

	\vec{X}	\vec{P}_1	\vec{P}_2	\vec{P}_3	\vec{P}_4	\vec{P}_5
\vec{P}_2	2	0	1	1	0,67	-0,33
\vec{P}_1	5	1	0	2	1,67	-0,33
руб.	-13	0	0	-5	-4,33	1,67
\$	0	0	0	0	-1	-1

В управляющей «долларовой» строке не осталось положительных элементов, следовательно, первый этап завершен. Анализируем долларовую составляющую целевой функции (выделена в табл. 4.7 жирным прифтом). Она равна нулю, следовательно, можно переходить ко второму этапу.

Отбрасываем долларовую управляющую строку и два последних искусственных столбца (табл. 4.8):

Таблица 4.8. Симплексная таблица после завершения первого этапа

	\vec{X}	\vec{P}_1	\vec{P}_2	\vec{P}_3
\vec{P}_2	2	0	1	1
\vec{P}_1	5	1	0	2
руб.	-13	0	0	-5

Тем самым ко второму этапу мы от расширенной вернулись к исходной задаче линейного программирования с единичным базисом и начальным опорным планом $\vec{X} = (5, 2, 0)^T$. Начинаем ите-

рации стандартного симплексного метода, используя в качестве управляющей обычную «рублевую» $(m + 1)$ -ю строку. Анализируя ее, видим, что все элементы этой строки неположительны, следовательно, итерации не понадобятся, данный опорный план сразу является оптимальным. Соответствующее значение целевой функции $L^* = -13$.

Замечание 1. Идея искусственного базиса может быть применена не только для построения исходного опорного плана, но и самостоятельно для решения любой системы линейных неравенств. Для этого исходные ограничения путем введения дополнительных переменных должны быть преобразованы в равенства и приведены к каноническому виду:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m, \\ x_1 \geq 0, \dots, x_n &\geq 0, \end{aligned}$$

а затем добавляются m искусственных переменных x_{n+1}, \dots, x_{n+m} .

Стоимостные коэффициенты при искусственных переменных полагаются равными единице: $c_{n+1} = \dots = c_{n+m} = 1$, а долларовые стоимости здесь не вводятся, так как все коэффициенты при основных переменных $c_1 = \dots = c_n = 0$, значит, вес искусственных переменных бесконечно большой по сравнению с основными. Соответствующая задача линейного программирования имеет вид

$$\begin{aligned} L = & \quad \quad \quad x_{n+1} + \dots + x_{n+m} \rightarrow \min, \\ & a_{11}x_1 + \dots + a_{1n}x_n + x_{n+1} = b_1, \\ & \dots \\ & a_{m1}x_1 + \dots + a_{mn}x_n + x_{n+m} = b_m, \\ & x_j \geq 0, \quad j = 1, \dots, m + n. \end{aligned}$$

Если исходная система неравенств непротиворечива, все искусственные векторы будут выведены из базиса и оптимальное значение целевой функции будет равно нулю.

Замечание 2. Приведенный алгоритм симплексного метода с искусственным базисом является универсальным и вполне работоспособным; он пригоден для реализации на компьютере, и с его помощью можно решить любую задачу линейного программирования (если абстрагироваться от маловероятного заикливания). Однако с точки зрения трудоемкости вычислений он не всегда оптимален, так как на каждой итерации пересчитывается вся симплексная таблица. Поэтому существуют различные модификации базового симплексного метода (модифицированный симплексный метод, двойственный симплексный метод), менее трудоемкие при больших размерах задачи. А для супербольших задач, насчитывающих десятки тысяч переменных, разработан так называемый *блочный симплексный метод*, позволяющий декомпозировать их на задачи меньшей размерности.

Вместе с тем для некоторых специфических задач линейного программирования можно предложить значительно более эффективные алгоритмы, удачно использующие особенности задачи. К ним, в частности, относятся транспортная задача и задача о назначениях, которые мы рассмотрим в гл. 6 и 7.

4.6. Исторические замечания

Современники отмечают, что Джорж Данциг отличался необыкновенной скромностью и постоянно подчеркивал роль, которую сыграл в линейном программировании Джон фон Нейман (Neumann, John von; 1903–1957). Вот как Данциг описывает¹ первую встречу с этим замечательным человеком (см. рис. 4.4):

¹ *Dantzig G. B. Reminiscences About the Origins of Linear Programming // Mathematical Programming: The State of the Art / Ed. by A. Bachem, M. Groetschel and B. Korte. — Berlin: Springer Verlag, 1983. — P. 79–86.*

«... прошел примерно год, прежде чем мы в 1948 г. осознали, насколько в действительности силен симплекс-метод. Тем временем я решил проконсультироваться у "великого" Джонни фон Неймана, чтобы посмотреть, что он скажет по поводу метода решения. Многими он считался ведущим математиком в мире. 3 октября 1947 г. я посетил его в первый раз в Институте перспективных исследований в Принстоне. Помню, я старался описать ему, как обычному смертному, задачу ВВС. Я начал с формулировки модели линейного программирования в терминах работ и ресурсов и т. п.



Рис. 4.4.
Джон фон Нейман

Фон Нейман сделал нечто, я думаю, нехарактерное для него. "Ближе к делу", — раздраженно сказал он. Имея в то время несколько более толстую кожу, я сказал себе: "О'кей, если он хочет побыстрее, он это получит". Менее чем за минуту я набросал геометрическую и алгебраическую версии задачи на доске. Фон Нейман встал и сказал: "А, это!..". После чего в течение следующих полутора часов он прочитал мне лекцию по математической теории линейных программ.

В какой-то момент заметив, что я сижу с открытым ртом и хлопаю глазами (перед этим проштудировав литературу и ничего не найдя), фон Нейман сказал: "Я не хочу, чтобы Вы думали, что я вытаскиваю все это из моего рукава экспромтом, как волшебник. Просто я недавно закончил книгу с Оскаром Моргенштерном по теории игр. Я полагаю, что эти обе задачи эквивалентны. Теория, которую я набросал для Вашей задачи, аналогичная той, которую мы разработали для игр»».

Данциг сам удивлялся удивительной эффективности симплексного метода, ведь время работы алгоритма должно быть экспоненциальной функцией от числа переменных, и гипотетически можно построить соответствующие экзотические примеры.

К счастью, на практике они не встречаются, на современных компьютерах успешно решаются задачи с сотнями тысяч и даже миллионами переменных.

Тем не менее исследователи продолжали поиск альтернативных методов, имеющих гарантированную трудоемкость. В 1979 г. Леонид Генрихович Хачиян (1952–2005) — российский математик, эмигрировавший в 1989 г. в США — опубликовал так называемый *эллипсоидный* метод, число шагов в котором является полиномиальной функцией от размера матрицы условий. Таким образом, на особо трудных задачах эллипсоидный метод должен быть быстрее, чем симплексный. С точки зрения теории метод действительно интересный, но практическое его значение оказалось ничтожным.

В 1984 г. индийский математик Кармахар (Karmakhar, Narendra; р. 1957), работавший в это время в Bell Laboratories, предложил *метод внутренней точки*, сочетающий теоретические возможности эллипсоидного метода с практическими достижениями симплексного. Это сообщение вызвало большой интерес у специалистов и привело к бурному развитию аналогичных алгоритмов. Методы внутренней точки основаны не на переходе от одной крайней точки области планов к соседней, а на движении внутри области. Хотя их работу оказалось довольно сложно теоретически исследовать, в настоящее время они рассматриваются как конкуренты симплексному методу и реализованы в некоторых продвинутых пакетах линейного программирования.

Глава 5

Теория двойственности

Теория двойственности — dual problem, к изучению которой мы приступаем, занимает центральное место в линейном программировании. Она целиком подтверждает известный тезис о том, что «нет ничего более практичного, чем хорошая теория».

С каждой задачей линейного программирования можно связать тесно связанную с ней так называемую *двойственную* задачу. Двойственная задача является в некотором смысле зеркальным отражением прямой. Прямая и двойственные задачи образуют *двойственную пару*. Какую из задач двойственной пары считать прямой, а какую двойственной, не имеет значения.

5.1. Симметричные двойственные задачи

Зеркальность двойственной пары нагляднее всего проявляется в случае, когда прямая задача сформулирована с ограничениями в виде неравенств, типичных для задачи о производственном плане.

Прямая задача имеет n переменных и m ограничений:

$$\vec{X} = (x_1, \dots, x_n)^T,$$

$$\begin{aligned}
L(\vec{X}) = & \quad c_1x_1 + \dots + c_nx_n \rightarrow \max, \\
& a_{11}x_1 + \dots + a_{1n}x_n \leq b_1, \\
& \quad \dots \quad \dots \quad \dots \quad \dots \\
& a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m, \\
& x_1, \dots, x_n \geq 0.
\end{aligned} \tag{5.1}$$

Двойственную задачу, содержащую, наоборот, m переменных и n ограничений, определим следующим образом:

$$\begin{aligned}
\vec{Y} = & (y_1, \dots, y_m)^T, \\
M(\vec{Y}) = & \quad b_1y_1 + \dots + b_my_m \rightarrow \min, \\
& a_{11}y_1 + \dots + a_{m1}y_m \geq c_1, \\
& \quad \dots \quad \dots \quad \dots \quad \dots \\
& a_{1n}y_1 + \dots + a_{mn}y_m \geq c_n, \\
& y_1, \dots, y_m \geq 0.
\end{aligned} \tag{5.2}$$

Сравнивая задачи (5.1) и (5.2), можно заметить четыре зеркальных соотношения между ними:

- 1) число переменных прямой задачи равно числу ограничений двойственной и наоборот, матрица условий транспонирована;
- 2) вектор стоимостей прямой задачи равен вектору ограничений двойственной и наоборот;
- 3) направления оптимизации противоположны;
- 4) знаки неравенств в ограничениях противоположны.

Зеркальность проявляется еще более ярко, если воспользоваться компактной матричной записью:

Прямая задача:	Двойственная задача:
$L(\vec{X}) = \vec{c}^T \vec{X} \rightarrow \max,$	$M(\vec{Y}) = \vec{b}^T \vec{Y} \rightarrow \min,$
$A\vec{X} \leq \vec{b},$	$A^T \vec{Y} \geq \vec{c},$
$\vec{X} \geq \vec{0}.$	$\vec{Y} \geq \vec{0}.$

Двойственные условия

Если собрать все условия прямой задачи, считая ограничения на неотрицательность, то их наберется $m + n$. Аналогично, в двойственной задаче также будет $n + m$ условий. Таким образом, весь набор ограничений двойственных задач можно разбить на $m + n$ пар двойственных условий:

$$\begin{array}{l}
 m \text{ пар: } A\vec{X} \leq \vec{b} \quad \leftrightarrow \quad \vec{Y} \geq \vec{0}, \\
 n \text{ пар: } \vec{X} \geq \vec{0} \quad \leftrightarrow \quad A^T \vec{Y} \geq \vec{c}.
 \end{array}$$

Пример. Задача о производственном плане с полным набором условий:

$$\begin{array}{rcl}
 L(x_1, x_2) = & 8x_1 & +6x_2 \rightarrow \max, \\
 & 10x_1 & +5x_2 \leq 50, \\
 & 6x_1 & +9x_2 \leq 54, \\
 & x_1 & \geq 0, \\
 & & x_2 \geq 0.
 \end{array}$$

Двойственная ей задача также будет иметь две переменные и полный набор четырех ограничений:

$$\begin{array}{rcl}
 M(y_1, y_2) = & 50y_1 & +54y_2 \rightarrow \min, \\
 & 10y_1 & +6y_2 \geq 8, \\
 & 5y_1 & +9y_2 \geq 6, \\
 & y_1 & \geq 0, \\
 & & y_2 \geq 0.
 \end{array}$$

Следовательно, имеется четыре пары двойственных условий:

$$\begin{aligned} 10x_1 + 5x_2 \leq 50 &\leftrightarrow y_1 \geq 0, \\ 6x_1 + 9x_2 \leq 54 &\leftrightarrow y_2 \geq 0, \\ x_1 \geq 0 &\leftrightarrow 10y_1 + 6y_2 \geq 8, \\ x_2 \geq 0 &\leftrightarrow 5y_1 + 9y_2 \geq 6. \end{aligned}$$

Двойственную задачу к задаче о производственном плане мы пока записали чисто формально, но, как мы увидим в дальнейшем, оно имеет очень наглядную и важную для экономической теории интерпретацию.

5.2. Несимметричные двойственные задачи

Если прямая задача записана в канонической форме, то есть с ограничениями в виде матричного равенства:

$$\begin{aligned} L(\vec{X}) = \vec{c}^T \vec{X} \rightarrow \min, \\ A\vec{X} = \vec{b}, \\ \vec{X} \geq \vec{0}, \end{aligned} \tag{5.3}$$

то двойственная задача будет выглядеть не столь зеркально. Для того чтобы найти ее вид, основываясь на определении симметричной пары, заменим равенство двумя неравенствами:

$$\begin{aligned} A\vec{X} &\geq \vec{b}, \\ -A\vec{X} &\geq -\vec{b}. \end{aligned}$$

Если воспользоваться символикой блочных матриц, эти неравенства можно объединить в одно:

$$\begin{pmatrix} A \\ -A \end{pmatrix} \vec{X} \geq \begin{pmatrix} \vec{b} \\ -\vec{b} \end{pmatrix}.$$

Теперь прямая задача имеет $2m$ ограничений-неравенств, поэтому двойственная задача будет иметь $2m$ переменных, которые мы обозначим с помощью блочного вектора $\begin{pmatrix} \vec{Y}_1 \\ \vec{Y}_2 \end{pmatrix}$, где векторы \vec{Y}_1 и \vec{Y}_2 имеют размерность m .

Записываем двойственную задачу, учитывая все четыре зеркальные соотношения и пользуясь тем, что с блочными матрицами можно работать как с обычными:

$$M = \begin{pmatrix} \vec{b} \\ -\vec{b} \end{pmatrix}^T \begin{pmatrix} \vec{Y}_1 \\ \vec{Y}_2 \end{pmatrix} = \vec{b}^T \vec{Y}_1 - \vec{b}^T \vec{Y}_2 = \vec{b}^T (\vec{Y}_1 - \vec{Y}_2) \rightarrow \max;$$

$$\begin{pmatrix} A \\ -A \end{pmatrix}^T \begin{pmatrix} \vec{Y}_1 \\ \vec{Y}_2 \end{pmatrix} = A^T \vec{Y}_1 - A^T \vec{Y}_2 = A^T (\vec{Y}_1 - \vec{Y}_2) \leq \vec{c},$$

$$\vec{Y}_1, \vec{Y}_2 \geq \vec{0}.$$

Поскольку неотрицательные переменные \vec{Y}_1 и \vec{Y}_2 входят во все выражения не индивидуально, а только в виде разности, эту разность можно заменить одним вектором $\vec{Y} = \vec{Y}_1 - \vec{Y}_2$ размерности m , но теперь уже не ограниченным по знаку.

С учетом сказанного двойственная задача приобретает следующий вид:

$$M(\vec{Y}) = \vec{b}^T \vec{Y} \rightarrow \max,$$

$$A^T \vec{Y} \leq \vec{c},$$

$$\vec{Y} - \text{любое.} \quad (5.4)$$

Заметим, что в паре двойственных задач (5.3) и (5.4) имеется только n пар двойственных условий. В матричных обозначениях они имеют вид

$$\vec{X} \geq 0 \leftrightarrow A^T \vec{Y} \leq \vec{c}, \quad (5.5)$$

или, если каждую компоненту векторов расписать отдельно:

$$x_j \geq 0 \leftrightarrow \sum_{i=1}^m a_{ij} y_i \leq c_j, \quad j = 1, \dots, n. \quad (5.6)$$

Пример. Если прямую задачу о производственном плане записать в канонической форме:

$$\begin{aligned} L(x_1, x_2, x_3, x_4) = & -8x_1 - 6x_2 + 0x_3 + 0x_4 \rightarrow \min, \\ & 10x_1 + 5x_2 + x_3 = 50, \\ & 6x_1 + 9x_2 + x_4 = 54, \\ & x_1, \dots, x_4 \geq 0, \end{aligned}$$

то двойственная ей несимметричная задача будет иметь 2 переменных и 4 ограничения-неравенства:

$$\begin{aligned} M(y_1, y_2) = & 50y_1 + 54y_2 \rightarrow \max, \\ & 10y_1 + 6y_2 \leq -8, \\ & 5y_1 + 9y_2 \leq -6, \\ & y_1 \leq 0, \\ & y_2 \leq 0, \\ & y_1, y_2 - \text{любые.} \end{aligned}$$

Двойственные пары условий:

$$\begin{aligned} x_1 \geq 0 & \leftrightarrow 10y_1 + 6y_2 \leq -8, \\ x_2 \geq 0 & \leftrightarrow 5y_1 + 9y_2 \leq -6, \\ x_3 \geq 0 & \leftrightarrow y_1 \leq 0, \\ x_4 \geq 0 & \leftrightarrow y_2 \leq 0. \end{aligned}$$

5.3. Первая теорема двойственности

Смысл теории двойственности раскрывается с помощью двух теорем, из которых первая является основной, а вторая получается как простое следствие из первой.

Теорема (Первая теорема двойственности).

I. Если одна из задач двойственной пары имеет решение, то и другая имеет решение, причем экстремальные значения целевых функций совпадают: $L(\vec{X}^) = M(\vec{Y}^*)$.*

II. Если одна из задач не имеет решения из-за неограниченности целевой функции, то другая также не имеет решения из-за противоречивости условий, и наоборот.

Доказательство (конструктивное).

Пусть имеется двойственная пара задач линейного программирования, причем прямая задача задана в канонической форме:

$$\begin{aligned} L(\vec{X}) = \vec{c}^T \vec{X} \rightarrow \min, & & M(\vec{Y}) = \vec{b}^T \vec{Y} \rightarrow \max, \\ A\vec{X} = \vec{b}, & & A^T \vec{Y} \leq \vec{c}, \\ \vec{X} \geq \vec{0}. & & \vec{Y} \text{ — любое.} \end{aligned}$$

Предположим, что прямая задача имеет решение, и мы нашли это решение симплексным методом. Пусть заключительный базис, соответствующий оптимальному плану, есть $B = (\vec{P}_1, \dots, \vec{P}_m)$. Тогда заключительная симплексная таблица имеет следующую структуру (сверху от двойной линии помещены элементы оформления):

План	\vec{P}_1	\vec{P}_2	\dots	\vec{P}_n
\vec{X}^*	\vec{P}'_1	\vec{P}'_2	\dots	\vec{P}'_n
L^*	$z_1 - c_1$	$z_2 - c_2$	\dots	$z_n - c_n$

Здесь

$$\begin{aligned} \vec{X}^* &\text{— урезанный до } m \text{ базисных компонент оптимальный план;} \\ \vec{c}_0 &\text{— соответственно урезанный вектор стоимостей;} \\ L^* = \vec{c}_0^T \vec{X}^* &\text{— оптимальное значение целевой функции;} \quad (5.7) \\ \vec{P}'_j &\text{— преобразованный вектор } \vec{P}_j. \end{aligned}$$

С алгебраической точки зрения симплексный метод эквивалентен переходу от начального единичного базиса

$I = (\vec{e}_1, \dots, \vec{e}_m)$ к заключительному оптимальному базису $B = (\vec{P}_1, \dots, \vec{P}_m)$.

При этом по формулам преобразования базиса (с. 61)

$$\vec{X}^* = B^{-1} \vec{b}, \quad (5.8)$$

$$\vec{P}'_j = B^{-1} \vec{P}_j. \quad (5.9)$$

Тогда эквивалентная стоимость вектора \vec{P}'_j в оптимальном базисе z_j равна

$$z_j = \vec{c}_0^T \vec{P}'_j = \vec{c}_0^T B^{-1} \vec{P}_j. \quad (5.10)$$

Так как матрица условий $A = (\vec{P}_1, \dots, \vec{P}_n)$ может быть интерпретирована как строка, состоящая из векторов-столбцов, то, собирая z_j в вектор-строку \vec{Z}^T , получаем

$$\vec{Z}^T = (z_1, \dots, z_n) = \vec{c}_0^T B^{-1} (\vec{P}_1, \dots, \vec{P}_n) = \vec{c}_0^T B^{-1} A. \quad (5.11)$$

Поскольку заключительный план оптимальный, то элементы управляющей строки $z_j - c_j \leq 0$, т. е. $\vec{Z}^T \leq \vec{c}^T$.

Подставляя \vec{Z}^T из (5.11), получаем

$$\vec{c}_0^T B^{-1} A \leq \vec{c}^T. \quad (5.12)$$

Главная и конструктивная часть доказательства состоит в том, что мы предполагаем, что оптимальный план \vec{Y}^* двойственной задачи нам известен:

$$\vec{Y}^* = (B^{-1})^T \vec{c}_0. \quad (5.13)$$

Докажем, что это действительно так и что теорема справедлива, т. е.

- а) \vec{Y}^* — план;
- б) \vec{Y}^* — оптимальный план, т. е. для любого плана \vec{Y} $M(\vec{Y}^*) \geq M(\vec{Y})$;
- в) $M(\vec{Y}^*) = L(\vec{X}^*)$.

Для доказательства «а» подставляем план (5.13) в двойственные условия:

$$\begin{aligned} A^T \vec{Y}^* &= A^T (B^{-1})^T \vec{c}_0 = [[A^T (B^{-1})^T \vec{c}_0]^T]^T = \\ &= [\vec{c}_0^T B^{-1} A]^T = [\text{учитывая (5.12)}] \leq [\vec{c}^T]^T = \vec{c}. \end{aligned}$$

Для доказательства «б» и «в» рассмотрим скалярное выражение $\vec{Y}^T A \vec{X}$.

С одной стороны, если \vec{X} — план прямой задачи, и для него $A \vec{X} = \vec{b}$, то

$$\vec{Y}^T A \vec{X} = \vec{Y}^T (A \vec{X}) = \vec{Y}^T \vec{b} = M(\vec{Y}).$$

С другой стороны, если \vec{Y} — план двойственной задачи и для него $A^T \vec{Y} \leq \vec{c}$, то

$$\vec{Y}^T A \vec{X} = [\text{транспонируем скаляр}] = \vec{X}^T (A^T \vec{Y}) \leq \vec{X}^T \vec{c} = L(\vec{X}).$$

Таким образом, для любых планов \vec{X} и \vec{Y} $L(\vec{X}) \geq M(\vec{Y})$.

Но для плана \vec{Y}^*

$$\begin{aligned} M(\vec{Y}^*) &= \vec{b}^T \vec{Y}^* = [\text{подставляем из (5.13)}] = \vec{b}^T (B^{-1})^T \vec{c}_0 = \\ &= [\text{транспонируем скаляр}] = \vec{c}_0^T B^{-1} \vec{b} = [\text{с учетом (5.8)}] = \\ &= \vec{c}_0^T \vec{X}^* = L(\vec{X}^*) \geq M(\vec{Y}). \end{aligned}$$

То есть \vec{Y}^* — действительно оптимальный план двойственной задачи, первая часть теоремы полностью доказана.

Для доказательства второй части предположим, что прямая задача двойственной пары не имеет решения из-за неограниченности целевой функции, т. е. $L(\vec{X}^*) = -\infty$. Тогда для всех планов двойственной задачи $M(\vec{Y}) \leq -\infty$, что невозможно. \square

Замечание 1. Первая теорема двойственности имеет очевидное обращение: если $L(\vec{X}^*) = M(\vec{Y}^*)$, то планы \vec{X}^* и \vec{Y}^*

оптимальные. Это следует из того, что, как мы только что показали, для любых планов \vec{X} и \vec{Y} $L(\vec{X}) \geq M(\vec{Y})$.

Замечание 2. Первая теорема двойственности дает возможность найти оптимальный план двойственной задачи, не решая ее. Оказывается, этот план спрятан в заключительной симплексной таблице.

Предположим, что исходная симплексная таблица содержала единичную подматрицу

$$A = (\dots | I).$$

Тогда, возвращаясь к формуле (5.11), получаем

$$\vec{Z}^T = \vec{c}_0^T B^{-1} A = \vec{c}_0^T B^{-1} (\dots | I) = (\dots | \vec{c}_0^T B^{-1}) = (\dots | \vec{Y}^{*T}).$$

Таким образом, часть вектора эквивалентных стоимостей \vec{Z} , соответствующая единичным векторам исходного базиса, равна оптимальным значениям двойственных переменных. Этот вектор (в виде разностей $z_j - c_j$) содержится в $m + 1$ -й (управляющей) строке заключительной симплексной таблицы. Если стоимости c_j , относящиеся к единичному исходному базису, равны 0, то соответствующие элементы управляющей строки $z_j - c_j = z_j - 0 = y_j^*$.

Схематически это показано на диаграмме, показывающей преобразование исходной симплексной таблицы в заключительную; символами X изображены несущественные элементы.

1	X ... X	1	→	X ... X	X ... X	
...
m	X ... X	1		X ... X	X ... X	X ... X
m + 1	X ... X	0 ... 0		X ... X	y_1^* ...	y_m^*

Исходная таблица

Заключительная таблица

Пример. Задача, двойственная задаче о производственном плане, имеет вид

$$\begin{aligned} M(y_1, y_2) &= 50y_1 + 54y_2 \rightarrow \min, \\ 10y_1 + 6y_2 &\geq 8, \\ 5y_1 + 9y_2 &\geq 6, \\ y_1, y_2 &\geq 0. \end{aligned}$$

Решая ее графическим методом (рис. 5.1), получаем оптимальный план

$$y_1^* = \frac{3}{5}, y_2^* = \frac{1}{3}, M(y_1^*, y_2^*) = 48.$$

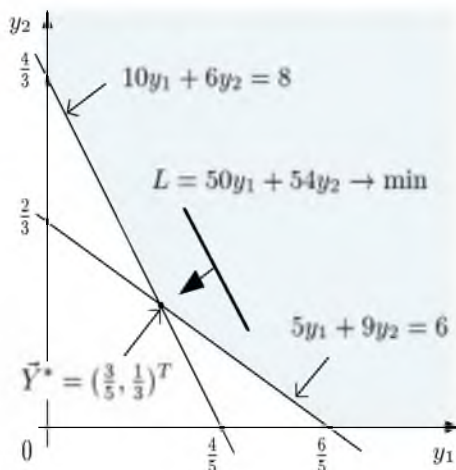


Рис. 5.1. Графическое решение двойственной задачи

Рассматривая заключительную симплексную таблицу для прямой задачи (см. с. 100), видим, что оптимальные значения двойственных переменных действительно находятся в 3-й строке под векторами \vec{P}_3, \vec{P}_4 , только с обратным знаком. Это объясняется тем, что исходная задача о производственном плане ставилась на максимизацию, а при переходе к канонической форме мы сменили знаки у стоимостных коэффициентов.

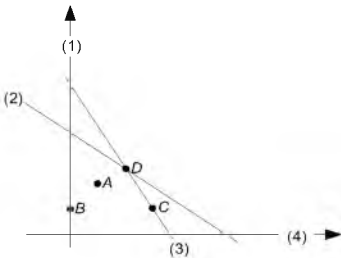
5.4. Вторая теорема двойственности

Принцип дополняющей нежесткости

Вторая теорема двойственности иначе называется *принципом дополняющей нежесткости*. Для ее формулировки нам потребуется ввести классификацию ограничений.

Если взять конкретный план задачи линейного программирования и подставить его в некоторое ограничение, имеющее вид нестрогого неравенства, то оно будет выполняться либо как равенство, либо как строгое неравенство. В случае равенства ограничение для этого плана будет *активным* (в другой терминологии — *жестким, связанным*), а при неравенстве — *неактивным* (*нежестким, свободным*).

Пример. Пусть имеется четыре ограничения (1), (2), (3), (4), показанных на рис. 5.2. Их классификация для планов A, B, C, D приведена здесь же.



План	Активные	Неактивные
A	Нет	(1), (2), (3), (4)
B	(1)	(2), (3), (4)
C	(3)	(1), (2), (4)
D	(2), (3)	(1), (4)

Рис. 5.2. Активные и неактивные ограничения

Теорема (Вторая теорема двойственности). Для оптимальных планов прямой и двойственной задачи в каждой паре двойственных условий только одно условие может быть неактивным (*нежестким, свободным*).

Доказательство. Приведем доказательство для несимметричного случая. Предположим, что прямая задача задана в ка-

ионической форме. Тогда, согласно (5.6), двойственные условия имеют следующий вид:

$$x_j \geq 0 \leftrightarrow \sum_{i=1}^m a_{ij}y_i \leq c_j, \quad j = 1, \dots, n.$$

Пусть \vec{X} и \vec{Y} — оптимальные планы.

Составим выражение:

$$\begin{aligned} \vec{X}^T[\vec{c} - A^T\vec{Y}] &= \vec{X}^T\vec{c} - \vec{X}^T A^T\vec{Y} = [\text{транспонируем скаляр}] = \\ &= \vec{X}^T\vec{c} - \vec{Y}^T A\vec{X} = [\text{т. к. } \vec{X} - \text{план}] = \vec{X}^T\vec{c} - \vec{Y}^T\vec{b} = \\ &= L(\vec{X}) - M(\vec{Y}) = [\text{по Первой теореме двойственности}] = 0. \end{aligned}$$

Расписываем покомпонентно скалярное произведение:

$$\vec{X}^T[\vec{c} - A^T\vec{Y}] = \sum_{j=1}^n \underbrace{x_j}_{\geq 0} \underbrace{\left[c_j - \sum_{i=1}^m a_{ij}y_i \right]}_{\geq 0} = 0, \quad j = 1, \dots, n.$$

Отсюда следует, что в каждой j -й паре двойственных условий

$$\text{либо } x_j = 0, \quad \text{либо } \sum_{i=1}^m a_{ij}y_i = c_j.$$

Доказательство для симметричного случая предлагается провести самостоятельно. \square

Замечание. Если планы прямой и двойственных задач являются невырожденными, то принцип дополняющей нежесткости звучит более категорично: для оптимальных планов прямой и двойственной задач в каждой паре двойственных условий одно является активным, другое — неактивным.

**Обращение
второй теоремы
двойственности**

Ценность второй теоремы двойственности состоит в том, что ее можно обратить и использовать как критерий оптимальности.

Пусть имеется опорный план

$$\vec{X} = (x_1, \dots, x_m, 0, \dots, 0)^T,$$

и мы хотим проверить его на оптимальность.

Предположим, что он невырожден, т. е. $x_1 > 0, \dots, x_m > 0$. Тогда для каждой ненулевой базисной переменной x_j можно записать двойственное условие

$$\sum_{i=1}^m a_{ij}y_i = c_j \quad j = 1, \dots, m.$$

Получилась система m уравнений с m неизвестными. Решая ее, находим значения двойственных переменных y_1, \dots, y_m .

Тогда, если для всех остальных небазисных x_j выполняется неравенство

$$\sum_{i=1}^m a_{ij}y_i \leq c_j, \quad j = m+1, \dots, n,$$

то данный план является оптимальным.

Для доказательства убедимся, что полученный таким образом критерий оптимальности совпадает с критерием $z_j - c_j \leq 0$, который мы использовали в симплексном методе, то есть

$$\sum_{i=1}^m a_{ij}y_i = z_j.$$

Действительно (см. доказательство Первой теоремы двойственности на с. 116),

$$\begin{aligned} z_j &= \vec{c}_0^T \vec{P}'_j = \vec{c}_0^T B^{-1} \vec{P}_j = \\ &= [\text{транспонируем скаляр}] = \vec{P}'_j{}^T (B^{-1})^T \vec{c}_0 = \\ &= \vec{P}_j{}^T \vec{Y}^* = [\text{расписываем покомпонентно}] = \sum_{i=1}^m a_{ij}y_i^*. \end{aligned}$$

Пример. Возьмем задачу о производственном плане и проверим на оптимальность вариант «Г» $X = (0, 6, 20, 0)^T$ из списка опорных планов на с. 83 (мы уже знаем, что он неоптимальный).

Перечень двойственных условий берем из примера на с. 115:

$$\begin{aligned} x_1 \geq 0 &\leftrightarrow 10y_1 + 6y_2 \leq -8, \\ x_2 \geq 0 &\leftrightarrow 5y_1 + 9y_2 \leq -6, \\ x_3 \geq 0 &\leftrightarrow y_1 \leq 0, \\ x_4 \geq 0 &\leftrightarrow y_2 \leq 0. \end{aligned}$$

Выбирая второе и третье условия, соответствующие положительным компонентам плана $X = (0, 6, 20, 0)^T$, составляем систему уравнений относительно двойственных переменных y_1, y_2 :

$$\begin{aligned} 5y_1 + 9y_2 &= -6, \\ y_1 &= 0. \end{aligned}$$

Отсюда $y_1 = 0, y_2 = -\frac{2}{3}$. Подставляя полученные значения в первое и четвертое условия, имеем

$$\begin{aligned} 10y_1 + 6y_2 &= -4 (\leq ?) - 8, \\ y_2 &= -\frac{2}{3} (\leq ?) 0. \end{aligned}$$

Первое условие не выполняется, следовательно, наш план не оптимальный, как и есть на самом деле.

5.5. Экономическая интерпретация двойственности

Двойственные переменные

До сих пор мы относились к двойственной задаче формально, однако для многих экономических задач возможна теоретически и практически значимая интерпретация двойственности.

Рассмотрим задачу линейного программирования типа задачи о производственном плане. Для выявления содержательного смысла двойственных переменных воспользуемся анализом размерности.

Прямая задача

$$L \begin{bmatrix} \text{Выручка,} \\ \text{руб.} \end{bmatrix} = \sum_{j=1}^n c_j \begin{bmatrix} \text{Цена } j\text{-го} \\ \text{изделия,} \\ \text{руб./шт.} \end{bmatrix} x_j \begin{bmatrix} \text{Кол-во } j\text{-х} \\ \text{изделий,} \\ \text{шт.} \end{bmatrix} \rightarrow \max,$$

$$\sum_{j=1}^n a_{ij} \begin{bmatrix} \text{Кол-во ресур-} \\ \text{сов } i\text{-го вида} \\ \text{на } j\text{-е изделие,} \\ \text{ед./шт.} \end{bmatrix} x_j \begin{bmatrix} \text{Кол-во } j\text{-х} \\ \text{изделий,} \\ \text{шт.} \end{bmatrix} \leq b_i \begin{bmatrix} \text{Запас ре-} \\ \text{сурсов } i\text{-го} \\ \text{вида, ед.} \end{bmatrix}.$$

Общие затраты ресурсов i -го вида

Двойственная задача

$$M [?] = \sum_{i=1}^m b_i \begin{bmatrix} \text{Запас ресур-} \\ \text{сов } i\text{-го вида,} \\ \text{ед.} \end{bmatrix} y_i [?] \rightarrow \min,$$

$$\sum_{i=1}^m a_{ij} \begin{bmatrix} \text{Кол-во ресурсов} \\ i\text{-го вида на } j\text{-е} \\ \text{изделие, ед./шт.} \end{bmatrix} y_i [?] \geq c_j \begin{bmatrix} \text{Цена } j\text{-го} \\ \text{изделия,} \\ \text{руб./шт.} \end{bmatrix}.$$

Из последнего соотношения следует, что размерность двойственных переменных y_i есть руб./ед.

Следовательно, для задачи о производственном плане двойственные переменные имеют смысл цен единиц ресурсов. В математической экономике они называются учетными, неявными или фиктивными ценами или, как назвал их Л. В. Канторович, объективно обусловленными оценками (ООО).

Учитывая это, двойственную задачу можно содержательно интерпретировать следующим образом:

$$M \begin{bmatrix} \text{Стоимость} \\ \text{ресурсов,} \\ \text{руб.} \end{bmatrix} = \sum_{i=1}^m b_i \begin{bmatrix} \text{Запас ре-} \\ \text{сурсов } i\text{-го} \\ \text{вида, ед.} \end{bmatrix} y_i \begin{bmatrix} \text{Цена единицы} \\ \text{ресурса } i\text{-го} \\ \text{вида, руб./ед.} \end{bmatrix} \rightarrow \min,$$

$$\underbrace{\sum_{i=1}^m a_{ij} \begin{bmatrix} \text{Кол-во ресурсов} \\ i\text{-го вида на } j\text{-е} \\ \text{изделие, ед./шт.} \end{bmatrix} y_i \begin{bmatrix} \text{Цена единицы} \\ \text{ресурса } i\text{-го} \\ \text{вида, руб./ед.} \end{bmatrix}}_{\text{Себестоимость } i\text{-го изделия}} \geq c_j \begin{bmatrix} \text{Цена } j\text{-го} \\ \text{изделия,} \\ \text{руб./шт.} \end{bmatrix}.$$

Таким образом, двойственная задача ставится как задача нахождения оптимальных цен, минимизирующих затраты на производство при ограничениях на себестоимость.

Двойственные условия

Предположим, что план производства $\vec{X} = (x_1, \dots, x_n)^T$ и учетные цены y_i на ресурсы установлены оптимальными. Основываясь на второй теореме двойственности, выясним содержательный смысл двойственных условий.

Возьмем первую группу двойственных условий:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \leftrightarrow y_i \geq 0, \quad i = 1, \dots, m.$$

Сумма в левой части равна суммарному объему потребляемых ресурсов i -го вида. Принцип дополняющей нежесткости для этой группы условий будет означать, что *если некоторый ресурс расходуется не полностью, то его учетная цена равна нулю*.

Рассмотрим вторую группу двойственных условий:

$$\sum_{i=1}^m a_{ij} y_i \geq c_j \leftrightarrow x_j \geq 0, \quad j = 1, \dots, n.$$

Здесь сумма по ресурсам определяет себестоимость изделия. Невозможность одновременного выполнения двух строгих неравенств говорит о том, что *если себестоимость некоторого изделия превышает его продажную цену, то такое изделие выпускать не нужно*. С точки зрения экономики эти требования выглядят вполне логично.

Чувствительность к ограничениям Двойственные переменные имеют еще одну содержательную интерпретацию, как показатели чувствительности целевой функции к ограничениям.

Предположим, мы решили задачу производственного планирования при заданных объемах ресурсов $\vec{b} = (b_1, \dots, b_m)^T$, нашли оптимальный план производства \vec{X}^* и соответствующее ему максимальное значение выручки от продажи произведенной продукции $L(\vec{X}^*)$. Увеличить выручку больше нельзя, не дадут ограничения на ресурсы.

Для расширения производства необходимо добавить ресурсы, которые можно приобрести, например, на рынке. Если купить Δb_i единиц i -го ресурса и пустить его в производства, за этот счет можно увеличить объем выпуска продукции и получить дополнительную выручку ΔL^* .

Спрашивается, какую предельную цену мы готовы заплатить за единицу i -го ресурса? Очевидно, эта величина равна $\frac{\Delta L^*}{\Delta b_i}$. При $\Delta b_i \rightarrow 0$ отношение превращается в дифференциальный показатель чувствительности оптимальной целевой функции к i -му ограничению.

Для определения этого показателя воспользуемся первой теоремой двойственности, которая утверждает, что $L(\vec{X}^*) = M(\vec{Y}^*)$.

Тогда

$$\frac{\partial L^*}{\partial b_i} = \frac{\partial M^*}{\partial b_i} = \frac{\partial \sum b_i y_i^*}{\partial b_i} = y_i^*.$$

Таким образом, двойственные переменные еще раз подтвердили право называться ценами на ресурсы.

5.6. Исторические замечания

Одним из удивительных исторических совпадений явилось одновременное, но независимое развитие теории линейного программирования и теории игр и обнаружившаяся впоследствии глубокая связь между этими двумя предметами. В связи с этим основателем теории двойственности считается фон Нейман, который пришел к идее двойственности, исходя из теории матричных игр.

Первая формулировка основной теоремы двойственности сохранилась в неопубликованном отчете Данцига, датированном 5 января 1948 г., составленном по итогам его разговора с фон Нейманом. На вопрос, заданный много лет спустя: «Почему вы это не опубликовали?», — Данциг ответил: «Потому, что это не мой результат — это результат фон Неймана. Все, что я сделал, — это записал для внутреннего обращения мое собственное доказательство того, что набросал фон Нейман».

Дальнейшие результаты в области теории двойственности были получены группой математиков из Принстонского университета (США) — профессором Таккером (Tucker, Albert William; 1905–1995) и его учениками Куном (Kuhn, Harold; р. 1925) и Гейлом (Gale, David; 1921–2008). Они дали строгое доказательство теоремы двойственности и обобщили ее на нелинейный случай (*теорема Куна — Таккера*, которую мы будем рассматривать, изучая *выпуклое программирование*).

Очень интересна и драматична отечественная судьба теории двойственности, которая оказалась тесно переплетенной с историей страны. Экономическая интерпретация двойственных переменных, изучаемых Л. В. Канторовичем в довоенные годы, настолько противоречила марксовской теории ценообразования, что он даже не решился назвать их ценами, а употребил идеологически нейтральный термин «объективно обусловленные оценки». Нечего и говорить, что никто из отечественных экономистов его тогда не поддержал.

Второе пришествие теории двойственности в советскую эконо-

мику состоялось в начале 1960-х годов и было связано с новыми веяниями в жизни страны.

Нынешнее поколение молодых людей, живущее в условиях рынка и экономической свободы, уже не помнит, как в то время было организовано социалистическое производство и распределение. Денег практически не было, вернее, формально они были, но на них, кроме скудного ассортимента предметов личного потребления, ничего нельзя было купить. Вместо рынка действовала административно-командная система планирования. На ее верху стоял всемогущий Госплан, который должен был рассчитать, сколько требуется населению, исходя из гипотетических «научно обоснованных» норм, хлеба, бритвенных лезвий или носков, затем пересчитывал это в планы производства зерна, стали, хлопка и электричества и разверстывал планы по государственным предприятиям (негосударственных не было), которые обязаны были их неукоснительно соблюдать. Никакой собственной прибыли предприятия не имели, их работники получали фиксированную зарплату независимо от того, нужна ли потребителю их продукция или нет. Поэтому материальной заинтересованности в выпуске разнообразной и высококачественной продукции не было.

Аналогично действовала система натурального распределения товаров, возглавляемая Госпланом и Госснабом. Каждый объект (станок, компьютер, письменный стол), запланированный к производству, заранее кому-то расписывался, потребители принудительно закреплялись за поставщиками.

Цены на все товары были стабильными и также устанавливались централизованно, за этим следил Государственный комитет по ценам. Розничные цены штамповались прямо на изделиях, и горе было тому продавцу государственного магазина (других не было), который осмелился их нарушить.

Согласно утопической теории научного коммунизма, такая система должна была обеспечить удовлетворение потребностей трудящихся и всеобщее благосостояние, но на практике все оказалось намного сложнее. Например, стоило какому-нибудь чиновни-

ку Госплана, отвечающему за трикотажные изделия, ошибиться в расчетах или соответствующему предприятию сорвать план, как во всей стране исчезали носки (реальный жизненный случай)¹. То тут, то там возникал *дефицит*, ставший неременным атрибутом и проклятием административно-командной системы хозяйства.

К 1960-м годам стало ясно, что централизованно уследить за производством необозримой номенклатуры товаров, необходимых цивилизованному человеку, невозможно, необходимо включать какие-то механизмы саморегуляции. А так как отказаться от коммунистических догм и полностью перейти к рыночной системе было невозможно, возникла идея экономической реформы, основанной на принципах *хозрасчета*.

По ее замыслу предприятиям, работающим в сфере производства предметов потребления, должна быть предоставлена некоторая экономическая свобода, чтобы они сами могли получать прибыль и составлять планы производства, ориентируясь на спрос. При этом цены на ресурсы и готовую продукцию опять-таки должны устанавливаться централизованно.

Глубинный смысл реформы состоял в том, чтобы прямое административное управление производством дополнить косвенным экономическим управлением через цены. Если цены установлены правильно, то предприятия, стремясь увеличить прибыль, самостоятельно, без давления сверху, стараются установить и выполнить планы производства, отвечающие реальным потребностям общества. Весь вопрос в том, каковы должны быть эти «правильные» цены.

Вот здесь и должно было помочь экономико-математическое моделирование, в частности линейное программирование с его двойственными переменными. Разумеется, рассмотренная нами модель производственного плана слишком примитивна для описания реального многоотраслевого хозяйства, поэтому в 1960–1970-е годы в стране, освободившейся к тому времени от догм периода

¹На эту тему есть занимательный рассказ Сергея Довлатова «Финские креповые носки» из цикла «Чемодан» (<http://www.lib.ru/DOWLATOW/chemodan.txt>).

культы личности, развернулись научные исследования по созданию математической теории оптимального функционирования социалистической экономики. Их возглавил один из основоположников экономико-математического направления академик Василий Сергеевич Немчинов (1894–1964), создавший в составе Академии наук СССР Центральный экономико-математический институт (ЦЭМИ). Поскольку решаемые задачи предполагали большой объем расчетов, было организовано производство современной вычислительной техники, началась работа по проектированию Общегосударственной автоматизированной системы управления (ОГАС), включающей автоматизированную систему плановых расчетов (АСПР) и автоматизированную систему обработки информации по ценам (АСОИцен).

Следует признать, что за эти годы в теории были достигнуты значительные успехи, работы активно поощрялись правительством, в 1965 г. Л. В. Канторовичу вместе с В. С. Немчиновым и руководителем ленинградской экономико-математической школы Виктором Валентиновичем Новожиловым (1892–1970) была присуждена Ленинская премия. Однако на практике построить социалистический рай на основе научных методов управления и компьютерных технологий не удалось. Система хозяйства, основанная на государственной собственности и централизованном управлении, оказалась в принципе неререформируемой и вскоре рухнула вместе со всей коммунистической идеологией.

Однако идея применения расчетных цен, полученных на основании решения двойственных задач линейного программирования, жива. Такие цены могут использоваться во внутрифирменном планировании.

Глава 6

Транспортная задача

6.1. Постановка и формы записи транспортной задачи

Постановка задачи

Имеются m пунктов A_1, \dots, A_m производства некоторого однородного продукта. Их производственные мощности равны a_1, \dots, a_m .

Имеются n пунктов B_1, \dots, B_n потребления этого продукта с потребностями b_1, \dots, b_n .

Предполагается, что $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ – условие баланса.

Стоимость перевозки единицы продукта из A_i в B_j задается матрицей $C = (c_{ij})_{m \times n}$.

Требуется определить, сколько продукта откуда куда нужно возить, чтобы:

- из пунктов производства весь продукт был вывезен;
- потребности пунктов потребления были бы удовлетворены;
- транспортные издержки были бы минимальны.

Для формализации задачи введем матрицу плана перевозок $X = (x_{ij})_{m \times n}$, где x_{ij} показывает количество продукта, перевозимого из A_i в B_j .

Тогда суммарные транспортные расходы

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (6.1)$$

а ограничения имеют вид

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = 1, \dots, m, \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, \dots, n, \\ x_{ij} &\geq 0. \end{aligned} \quad (6.2)$$

Отсюда видно, что транспортная задача изначально поставлена в канонической форме (c_{ij} , a_i и b_j мы предполагаем неотрицательными), имеет mn переменных и $m + n$ ограничений.

Пример. Процесс решения транспортной задачи мы будем иллюстрировать следующим простейшим примером.

В городе, имеющем прямоугольную планировку улиц (рис. 6.1) расположены два склада A_1 и A_2 некоторого товара (на рисунке обозначены квадратиками) и три магазина B_1, B_2, B_3 (обозначены кружками).

Запасы товара на складах $a_1 = 5, a_2 = 7$, потребности магазинов $b_1 = 2, b_2 = 4, b_3 = 6$. Задача, как видим, сбалансирована: $\sum a_i = \sum b_j = 12$.

Стоимость перевозки будем измерять длиной пути, измеряемой количеством кварталов, разделяющих склад и магазин (расстояние по Манхэттену).

Матрицы стоимостей и плана имеют размерность 2×3 :

$$C = \begin{pmatrix} 2 & 4 & 2 \\ 3 & 1 & 3 \end{pmatrix}, \quad X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}.$$

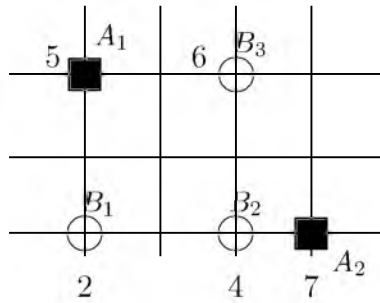


Рис. 6.1. Пример транспортной задачи

Тогда задача в развернутой записи будет иметь следующий вид:

$$\begin{aligned}
 L(X) = & 2x_{11} + 4x_{12} + 2x_{13} + 3x_{21} + x_{22} + 3x_{23} \rightarrow \min, \\
 & x_{11} + x_{12} + x_{13} = 5, \\
 & \phantom{x_{11} + x_{12} + x_{13}} + x_{21} + x_{22} + x_{23} = 7, \\
 & x_{11} \phantom{+ x_{12} + x_{13}} + x_{21} = 2, \\
 & \phantom{x_{11} + x_{12} + x_{13}} x_{12} \phantom{+ x_{21} + x_{22} + x_{23}} + x_{22} = 4, \\
 & \phantom{x_{11} + x_{12} + x_{13}} \phantom{+ x_{21} + x_{22} + x_{23}} x_{13} \phantom{+ x_{21} + x_{22} + x_{23}} + x_{23} = 6, \\
 & x_{11}, \dots, x_{23} \geq 0.
 \end{aligned}$$

Особенности векторной формы записи

Глядя на развернутую запись транспортной задачи, легко заметить, что матрица ее условий довольно специфична: все ее элементы равны либо 0, либо 1, причем в каждом столбце содержится ровно две единицы.

Если представить задачу в векторной записи

$$\begin{aligned}
 L(X) &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \\
 \sum_{i=1}^m \sum_{j=1}^n x_{ij} \vec{P}_{ij} &= \vec{P}_0, \\
 x_{ij} &\geq 0,
 \end{aligned}$$

то размерность входящих в нее векторов будет равна $m+n$, а сами векторы будут иметь следующий вид:

$$\vec{P}_{ij} = \begin{pmatrix} 0 & 1 \\ \dots & \\ 1 & i \\ \dots & \\ 0 & m \\ \hline 0 & 1 \\ \dots & \\ 1 & j \\ \dots & \\ 0 & n \end{pmatrix}, \quad \vec{P}_0 = \begin{pmatrix} a_1 \\ \dots \\ a_m \\ \hline b_1 \\ \dots \\ b_n \end{pmatrix}.$$

Как мы увидим в дальнейшем, именно такая специфическая структура векторов условий дает возможность адаптировать базовый симплексный метод к особенностям транспортной задачи и тем самым повысить его эффективность.

Несбалансированная транспортная задача

До сих пор мы предполагали, что транспортная задача сбалансирована, однако может случиться так, что условие баланса не будет выполняться. В этом случае производится принудительная балансировка транспортной задачи.

1. Если объем производства больше объема потребления, то вводится фиктивный пункт потребления B_{n+1} (нечто вроде свалки), потребность которого равна дисбалансу:

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j,$$

а перевозки туда бесплатны: $c_{i,n+1} = 0$, $i = 1, \dots, m$.

2. Если объем производства меньше объема потребления, то формальная балансировка невозможна, так как производимого продукта все равно не хватит. Можно, конечно, ввести штрафы за недопоставку, но это будет уже другая задача.

Замечание 1. Мы рассматриваем случай, когда перевозится один продукт, такая задача называется *однопродуктовой*. Однако транспортную задачу можно сформулировать и для многопродуктового варианта.

Замечание 2. Решать транспортную задачу можно любым универсальным методом линейного программирования, например симплексным. При этом нужно иметь в виду, что реальные задачи имеют большую размерность. Например, уже при 100 производителях и 100 потребителях симплексная таблица будет иметь размерность $200 \times 10\,000$. Поэтому имеет смысл разрабатывать более эффективные методы, учитывающие специфический вид условий транспортной задачи.

6.2. Свойства транспортной задачи

Специфичность транспортной задачи проявляется в следующих четырех свойствах.

Свойство 1 (разрешимость). *Сбалансированная транспортная задача всегда имеет решение.*

Доказательство. Для разрешимости любой задачи линейного программирования необходимо доказать: а) непротиворечивость

условий; б) ограниченность целевой функции.

а) Для доказательства непротиворечивости условий достаточно указать хотя бы один план.

Пусть $\sum a_i = \sum b_j = D$. Возьмем $x_{ij} = \frac{a_i b_j}{D} \geq 0$. Тогда

$$\sum_j x_{ij} = \sum_j \frac{a_i b_j}{D} = \frac{a_i}{D} \sum b_j = a_i,$$

$$\sum_i x_{ij} = \sum_i \frac{a_i b_j}{D} = \frac{b_j}{D} \sum a_i = b_j;$$

б) Целевая функция ограничена в сторону минимизации, так как

$$L(X) = \sum_i \sum_j c_{ij} x_{ij} \geq 0.$$

Свойство доказано. \square

Свойство 2 (размерность). Ранг матрицы условий равен $m + n - 1$, т. е. опорные планы транспортной задачи могут содержать не более $m + n - 1$ ненулевых компонент.

Доказательство. Для доказательства необходимо показать: а) что все миноры порядка $m + n$ равны 0; б) что существует минор порядка $m + n - 1$, не равный 0.

Доказательство «а». Возьмем любой минор порядка $m + n$. Его матрица имеет $m + n$ строк, причем из структуры векторов условий следует, что в каждом столбце обязательно будет по единице в первой группе из m строк и во второй группе из n строк.

Если мы в первой группе строк прибавим все строки к первой, а во второй — все строки к последней, то получим матрицу, у которой первая и последняя строки состоят из единиц. Определитель такой матрицы равен нулю, поскольку он имеет две одинаковых строки:

Этот результат является вполне предсказуемым, так как не все ограничения являются независимыми из-за условия баланса производства и потребления.

Свойство 3 (унимодулярность). *Любой минор матрицы условий транспортной задачи равен 0 либо ± 1 . Такая матрица называется унимодулярной.*

Доказательство индукцией по размерности k .

При $k = 1$ утверждение очевидно.

Пусть утверждение справедливо для $1, \dots, k - 1$. Докажем его для k .

Возьмем минор k -го порядка. Возможны три варианта.

- Все столбцы его матрицы содержат точно по две единицы. Этот минор равен нулю (см. предыдущее доказательство).
- Найдется столбец, не содержащий ни одной единицы. Такой минор равен 0.
- Найдется столбец, содержащий одну единицу. Раскладываем минор по этому столбцу. В силу индуктивного предположения утверждение справедливо.

□

Свойство 4 (целочисленность). *Если объемы производства и потребления целочисленны или кратны Q , то все опорные планы транспортной задачи, в том числе оптимальный план, также целочисленны или кратны Q .*

Доказательство. Пусть имеется опорный план X транспортной задачи. Для $m + n - 1$ его ненулевых компонент справедливо разложение

$$\sum_{ij} x_{ij} \vec{P}_{ij} = \vec{P}_0,$$

причем векторы \vec{P}_{ij} линейно независимы.

Это — векторная запись системы $m + n$ линейных уравнений с $m + n - 1$ неизвестными. Ее можно решить по правилу Крамера (см. с. 47) и найти компоненты плана $X = (x_{ij})$.

Выберем в матрице условий базисный минор B размерности $m+n-1$ и заменим в нем столбец P_{ij} вектором свободных членов.

Тогда, раскладывая определитель по столбцу свободных членов, получаем

$$x_{ij} = \frac{\begin{vmatrix} 0 & \cdots & a_1 & \cdots & 0 \\ 1 & \cdots & a_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & a_m & \cdots & 1 \\ \hline 0 & \cdots & b_1 & \cdots & 1 \\ 1 & \cdots & b_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & b_n & \cdots & 0 \end{vmatrix}}{\|B\|} = \frac{\sum_i \alpha_i a_i + \sum_j \beta_j b_j}{\pm 1},$$

где

$$\alpha_i, \beta_j \in \{0, +1, -1\}.$$

□

Итак, мы доказали ряд специфических свойств транспортной задачи. Опираясь на них, можно адаптировать симплексный метод таким образом, чтобы эти свойства учитывались в наибольшей степени.

Общая схема метода остается прежней: строится исходный базис и соответствующий ему опорный план, затем проводятся итерации до тех пор, пока не будет найдено оптимальное решение.

На каждой итерации текущий план проверяется на оптимальность, определяются векторы, включаемые и исключаемые из базиса, и производится преобразование плана перевозок по правилу Жордана.

Вместе с тем имеются некоторые отличия от базового симплексного метода. Прежде всего, вместо одномерной индексации переменных в транспортной задаче будет использоваться двойная. Кроме того, некоторые шаги алгоритма, сохраняя принципиальную схожесть, из-за специфики задачи здесь будут реализованы более простым образом.

Начнем с проблемы построения исходного опорного плана.

6.3. Построение исходных опорных планов

Разложение векторов условий

Прежде чем заняться собственно опорным планом, нужно научиться раскладывать векторы условий по совокупности других векторов условий.

Возникает естественный вопрос: зачем это нужно, ведь существуют стандартные процедуры решения данных задач, основанные на методах линейной алгебры.

Дело в том, что векторы, встречающиеся в транспортной задаче, столь специфичны (по две единицы в столбце, остальные нули), что указанные процедуры намного проще реализуются непосредственно, путем комбинаторных построений.

Правило. Для того, чтобы разложить вектор условий \vec{P}_{ij} транспортной задачи по другим векторам условий, нужно построить замкнутый чередующийся маршрут из единиц векторов, замыкающийся на разлагаемый вектор. Коэффициенты разложения равны $+1$ для нечетных векторов в маршруте, -1 для четных векторов и 0 для векторов, не вошедших в маршрут.

Это правило становится очевидным, если его показать на нашем примере ($m = 2, n = 3$). Пусть требуется разложить \vec{P}_{13} по векторам $\{\vec{P}_{11}, \vec{P}_{12}, \vec{P}_{22}, \vec{P}_{23}\}$, то есть представить его в виде суммы: $\vec{P}_{13} = \alpha_{11}\vec{P}_{11} + \alpha_{12}\vec{P}_{12} + \alpha_{22}\vec{P}_{22} + \alpha_{23}\vec{P}_{23}$.

Если расписать все участвующие здесь векторы, то получится картина, изображенная на рис. 6.2. Из нее следует, что верхняя единица в \vec{P}_{13} может быть получена, если положить $\alpha_{12} = 1$. Но тогда появится лишняя нижняя единица, которую можно скомпенсировать, взяв $\alpha_{22} = -1$. Но тогда появится лишняя верхняя -1 , которую можно скомпенсировать, положив $\alpha_{23} = 1$. Вектор \vec{P}_{11} в маршруте не участвует, поэтому $\alpha_{11} = 0$. В итоге мы получаем следующее разложение:

$$\vec{P}_{13} = 0 \cdot \vec{P}_{11} + 1 \cdot \vec{P}_{12} - 1 \cdot \vec{P}_{22} + 1 \cdot \vec{P}_{23}.$$

Очевидно, маршрут из единиц всегда получается циклический и чередующийся (по верхним единицам — по нижним единицам и т. д.).

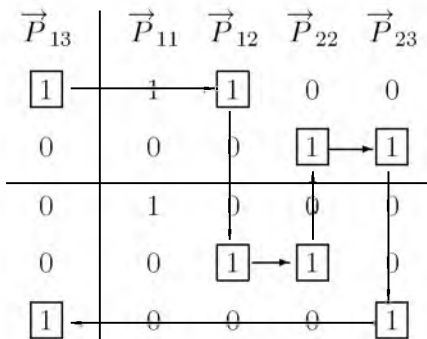


Рис. 6.2. Пример разложения вектора условий в транспортной задаче

Построение маршрута из единиц можно сделать более наглядным, если вспомнить, что каждому вектору \vec{P}_{ij} соответствует элемент x_{ij} плана перевозок. Тогда движение по верхним единицам будет соответствовать перемещению по строке матрицы X (индекс i не меняется), а движение по нижним единицам — перемещению по столбцу (индекс j не меняется).

С учетом этого правило можно модифицировать: для того, чтобы разложить \vec{P}_{ij} по другим векторам, нужно построить чередующийся (по строке — по столбцу — по строке — по столбцу...) замкнутый маршрут из соответствующих перевозок, замыкающийся на перевозку, соответствующую разлагаемому вектору. Коэффициенты разложения равны +1 для нечетных перевозок в маршруте, -1 для четных перевозок и 0 для перевозок, не вошедших в маршрут.

Для демонстрации повторим наш пример разложения \vec{P}_{13} по векторам $\{\vec{P}_{11}, \vec{P}_{12}, \vec{P}_{22}, \vec{P}_{23}\}$, используя матрицу плана перевозок (см. рис. 6.3). Разлагаемый вектор соответствует элементу,

отмеченному звездочкой. Результат получится тот же самый:

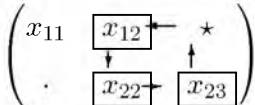
$$\vec{P}_{13} = \vec{P}_{12} - \vec{P}_{22} + \vec{P}_{23}.$$


Рис. 6.3. Разложение вектора условий путем построения замкнутого чередующегося маршрута на матрице плана

Из приведенного правила немедленно следует специфический критерий линейной независимости для векторов условий транспортной задачи.

Критерий линейной независимости. Система векторов условий является линейно независимой, если из соответствующих им перевозок нельзя построить замкнутого маршрута.

В предыдущем параграфе мы показали, что ранг матрицы условий равен $m + n - 1$ (свойство 2). Следовательно, среди векторов условий могут быть только $m + n - 1$ линейно независимых, и всякие $m + n - 1$ линейно независимых векторов образуют базис.

Теперь мы готовы сформулировать критерий опорности плана перевозок в транспортной задаче, вспоминая, что план задачи линейного программирования является опорным, если соответствующие ненулевым компонентам плана векторы линейно независимы.

Критерий опорности. План перевозок $X = (x_{ij})$ является опорным, если из его ненулевых перевозок нельзя составить чередующийся (строка — столбец — строка — столбец...) замкнутый маршрут.

Как мы уже отмечали, в опорном плане не может быть более $m + n - 1$ ненулевых перевозок. Если их ровно $m + n - 1$, то план является невырожденным.

**Метод
северо-
западного
угла**

После теоретического вступления можно приступить к описанию практических способов построения начального опорного плана перевозок.

Простейший из них называется *методом северо-западного угла*. Название объясняется тем, что заполнение матрицы плана начинается с левого верхнего элемента. Продемонстрируем его на нашем примере. Сначала построим пустую матрицу плана перевозок $X = (x_{ij})_{m \times n}$ нужной размерности (в нашем случае 2×3), к которой справа и снизу приписаны объемы производства и потребления. Для любого плана суммы по строкам и столбцам должны быть равны этим числам:

			$a_1 = 5$
			$a_2 = 7$
$b_1 = 2$	$b_2 = 4$	$b_3 = 6$	

Ищем северо-западную незаполненную клетку (сейчас это будет элемент x_{11}) и записываем туда $x_{11} = \min(a_1, b_1) = \min(2, 5) = 2$, после чего заполняем нулями ту строку или тот столбец, которая или который соответствуют минимуму. Сейчас это будет первый столбец. Далее корректируем объемы производства и потребления с учетом запланированной перевозки. Получаем

2			5 3
0			7
2	4	6	
0			

Снова ищем северо-западный незаполненный угол и заполняем клетку $x_{12} = \min(3, 4) = 3$:

2	3	0	5 3 0
0			7
2	4	6	
0	1		

Продолжая эту процедуру до конца, получаем план

$$X = \begin{array}{|c|c|c|} \hline 2 & 3 & 0 \\ \hline 0 & 1 & 6 \\ \hline \end{array}$$

По способу построения этот план будет опорным, так как на каждой итерации оставшаяся от включаемого элемента часть строки или столбца заполняется нулями, следовательно, возможность чередующегося (строка — столбец — строка — столбец...) циклического маршрута по ненулевым элементам плана исключается.

Количество ненулевых элементов в получившемся опорном плане равно $m + n - 1 = 2 + 3 - 1 = 4$, следовательно, он является невырожденным.

Полученный план перевозок можно отобразить графом (рис. 6.4). На нем хорошо видно, что в графе перевозок отсутствуют циклы, то есть он является деревом. Этот факт мы используем в дальнейшем.

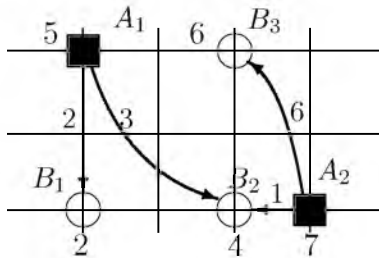


Рис. 6.4. План перевозок, полученный методом северо-западного угла

Если объемы перевозок умножить на соответствующие стоимости и просуммировать, то получим величину транспортных расходов для данного плана:

$$C = \begin{pmatrix} 2 & 4 & 2 \\ 3 & 1 & 3 \end{pmatrix}, \quad L = \sum_i \sum_j c_{ij} x_{ij} = \begin{pmatrix} 2^{(2)} & 3^{(4)} & 0^{(2)} \\ 0^{(3)} & 1^{(1)} & 6^{(3)} \end{pmatrix} = 35.$$

Метод минимального элемента

Метод северо-западного угла имеет тот недостаток, что при заполнении очередной клетки в матрице перевозок никак не учитывается ее стоимость, поэтому построенный опорный план может оказаться далеким от оптимального.

В методе минимального элемента вся схема построения опорного плана сохраняется, только на каждой итерации ищется не северо-западный, а минимальный по стоимости перевозок элемент.

Если применить этот метод к нашему примеру (подробное изложение хода заполнения матрицы опускаем), получится следующий план:

$$X = \begin{array}{|c|c|c|} \hline 2 & 0 & 3 \\ \hline 0 & 4 & 3 \\ \hline \end{array}$$

Транспортные издержки для этого плана:

$$L = \begin{pmatrix} 2^{(2)} & 0^{(4)} & 3^{(2)} \\ 0^{(3)} & 4^{(1)} & 3^{(3)} \end{pmatrix} = 23.$$

Проблема вырожденности

Рассматривая симплексный метод, мы отмечали, что вырожденность опорных планов, когда число ненулевых компонент меньше ранга матрицы условий, является неприятным явлением, способным привести к заикливанию алгоритма.

Мы отмечали, что в общей задаче линейного программирования вырожденность встречается очень редко. В транспортной же задаче, наоборот, вырожденность является скорее правилом, нежели исключением, поэтому обойти стороной этот вопрос мы не можем.

Для того чтобы бороться с явлением, нужно знать его причины. Начнем опять с примера.

Изменим немного постановку нашей задачи, перераспределив объемы запасов на складах: $a_1 = 6, a_2 = 6$ и оставив объемы потребления прежними, так что условие баланса сохраняется. По-

строим для этого случая опорный план методом северо-западного угла.

На первой итерации не замечаем ничего необычного:

2			ϕ 4
0			
2	4	6	
0			

Однако на второй итерации, когда $x_{12} = \min(4, 4) = 4$, мы вынужденно заполняем нулями не только строку, но и столбец:

2	4	0	ϕ 4 0
0	0		
2	4	6	
0	0		

Это приводит к выпаданию одной ненулевой компоненты, в результате построенный план оказывается вырожденным:

$$X = \begin{pmatrix} 2 & 4 & 0 \\ 0 & 0 & 6 \end{pmatrix}.$$

Если нарисовать граф перевозок, соответствующий этому плану, то можно заметить, что в отличие от предыдущего он получился несвязным (см. рис. 6.5).

Несвязность вырожденных планов можно доказать строго, пользуясь понятиями теории графов.

Рассмотрим граф перевозок опорного плана. Он не содержит циклов, поэтому его цикломатическое число $\lambda = M - N + \kappa = 0$, где $M = s$ — число ребер, $N = m + n$ — число вершин, κ — число компонент связности. Отсюда $\kappa = N - M = m + n - s$. Если $s < m + n - 1$, то $\kappa > 1$, т. е. граф перевозок несвязен.

Таким образом, вырожденным планам соответствуют несвязные графы, а это значит, что существуют сепаратные взаимно сбалансированные группы пунктов производства и потребления.

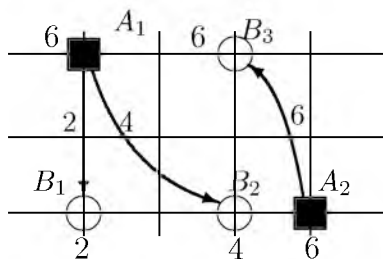


Рис. 6.5. Вырождение плана перевозок

Теперь мы видим, почему в видоизмененной задаче получился вырожденный план. Так как $a_1 = b_1 + b_2$, то это значит, что неполная группа производителей и потребителей $\{A_1, B_1, B_2\}$ внутренне сбалансирована. То же можно сказать о группе $\{A_2, B_3\}$. В противоположность этому в исходной задаче объемы производства и потребления специально были подобраны так, что никакой сбалансированный сепаратный створ между группами пунктов производства и потребления невозможен.

Эпсилон-метод устранения вырожденности

задачи.

Пусть задана транспортная задача с объемами производства и потребления a_1, \dots, a_m и b_1, \dots, b_n соответственно, причем все эти числа целочисленны или кратны некоторой величине Q .

На подготовительном этапе возьмем некоторую малую возмущающую константу

$$0 < \varepsilon < \frac{Q}{2m}$$

(в программе проще всего положить $\varepsilon = Q/(2m + 1)$) и скоррек-

тируем объемы следующим образом:

$$a'_i = a_i + \varepsilon, \quad i = 1, \dots, m;$$

$$b'_j = \begin{cases} b_j, & j = 1, \dots, n-1, \\ b_j + m\varepsilon, & j = n. \end{cases}$$

Глобальное условие баланса при этом сохранится, однако, если в исходной задаче наблюдались условия баланса в отдельных группах производства и потребления, они хоть ненамного, но нарушатся. В результате все опорные планы станут невырожденными, и в таком состоянии транспортную задачу можно дорешать до конца (метод решения мы рассмотрим позже).

После того как будет получен оптимальный план перевозок, его компоненты снова нужно округлить до чисел, кратных Q . Это можно сделать по формуле

$$x_{ij} := \left[\frac{x_{ij}}{Q} + \frac{1}{2} \right] Q,$$

где квадратные скобки означают целую часть числа.

Теперь мы рассмотрели все вопросы, связанные с построением начального опорного плана. Двигаясь по общей схеме симплексного метода, подходим к очередному этапу — проверке плана на оптимальность.

6.4. Критерий оптимальности транспортной задачи

**Задача,
двойственная
транспортной**

Для построения критерия оптимальности плана перевозок воспользуемся обращением второй теоремы двойственности.

Прежде всего построим задачу, двойственную транспортной. Напомним прямую задачу:

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min,$$

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m,$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n,$$

$$x_{ij} \geq 0.$$

Поскольку она записана в канонической форме, имеет mn переменных и $m+n$ ограничений-равенств, то двойственная задача будет иметь $m+n$ переменных и mn ограничений-неравенств, причем двойственные переменные неограничены по знаку.

Переменные двойственной задачи принято разбивать на две группы, соответствующие пунктам производства и потребления; они называются *потенциалами*:

$$\begin{aligned} \vec{Y} &= (y_1, \dots, y_m, y_{m+1}, \dots, y_{m+n})^T = \\ &= (u_1, \dots, u_m; v_1, \dots, v_n)^T. \end{aligned}$$

Целевая функция двойственной задачи равна

$$M(u_1, \dots, u_m; v_1, \dots, v_n) = \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j.$$

Так как матрица условий A прямой задачи имеет ровно две единицы в каждом столбце, то при ее транспонировании при переходе к двойственной задаче образуются по две единицы в каждой строке. Во время матричного умножения $A^T \vec{Y}$ строка из двух единиц, умножаясь на столбец $\vec{Y} = (y_1, \dots, y_m, y_{m+1}, \dots, y_{m+n})^T$,

«вырежет» из вектора \vec{Y} компоненты u_i и v_j . Поэтому система ограничений для двойственной задачи имеет вид

$$\begin{aligned}u_i + v_j &\leq c_{ij}, \\-\infty < u_i < \infty, \\-\infty < v_j < \infty, \\i &= 1, \dots, m; \quad j = 1, \dots, n.\end{aligned}$$

Критерий оптимальности

Рассматривая ограничения прямой и двойственной задач, можно выделить mn пар двойственных условий:

$$x_{ij} \geq 0 \leftrightarrow u_i + v_j \leq c_{ij}, \quad i = 1, \dots, m; \quad j = 1, \dots, n.$$

Применяя обращение второй теоремы двойственности (с. 122), сформулируем следующий критерий оптимальности транспортной задачи.

Опорный план является оптимальным, если существуют такие потенциалы u_i, v_j , находимые из соотношения $u_i + v_j = c_{ij}$ для $x_{ij} > 0$, что $u_i + v_j \leq c_{ij}$.

Таким образом для проверки опорного плана на оптимальность нужно:

- выбрать ненулевые перевозки x_{ij} ;
- для каждой из них составить уравнение для соответствующих потенциалов: $u_i + v_j = c_{ij}$;
- решить полученную систему уравнений и найти потенциалы;
- с полученными потенциалами проверить выполнение неравенств $u_i + v_j \leq c_{ij}$, $i = 1, \dots, m; \quad j = 1, \dots, n$. Если все неравенства справедливы, то данный план оптимальный.

Процедура несложная, но в ней есть одна тонкость. Дело в том, что в опорном плане имеется $m + n - 1$ ненулевых компонент (от вырожденности мы раз и навсегда защитились эpsilon-методом), а число потенциалов равно $m + n$. Система уравнений получается недоопределенная. Но это не должно нас смущать, так как реальная размерность прямой транспортной задачи, как мы установили, как раз равна $m + n - 1$ (одну степень свободы забирает условие баланса). Следовательно, в двойственной задаче имеется $m + n - 1$ независимых переменных, одной из них, скажем u_1 , можно придать произвольное значение, например нуль.

Пример. Проверим на оптимальность план, полученный методом минимального элемента (для удобства составления уравнений справа от каждой компоненты x_{ij} плана указано соответствующее значение стоимостного коэффициента c_{ij}):

$$X = \begin{pmatrix} 2^{(2)} & 0^{(4)} & 3^{(2)} \\ 0^{(3)} & 4^{(1)} & 3^{(3)} \end{pmatrix}, \quad L(X) = 23.$$

Составим систему уравнений для потенциалов:

$$\begin{aligned} x_{11} > 0 & \rightarrow u_1 + v_1 = 2, \\ x_{13} > 0 & \rightarrow u_1 + v_3 = 2, \\ x_{22} > 0 & \rightarrow u_2 + v_2 = 1, \\ x_{23} > 0 & \rightarrow u_2 + v_3 = 3, \\ & u_1 = 0. \end{aligned}$$

Решая ее, находим потенциалы $u_1 = 0$, $u_2 = 1$; $v_1 = 2$, $v_2 = 3$, $v_3 = 2$.

Теперь проверяем неравенства для нулевых x_{ij} :

$$\begin{aligned} x_{12} = 0 & \rightarrow u_1 + v_2 = 0 + 3 \leq 4, \\ x_{21} = 0 & \rightarrow u_2 + v_1 = 1 + 2 \leq 3. \end{aligned}$$

Следовательно, данный план оптимальный.

Практический алгоритм

Система уравнений для нахождения потенциалов довольно специфична, и для ее решения нет нужды использовать универсальные алгоритмы линейной алгебры. Правда, у меня был случай, когда один студент решал эту систему методом Гаусса — Жордана, но это уже слишком...

Для работы с потенциалами удобно ввести в рассмотрение матрицу невязок $S = (s_{ij})$, где $s_{ij} = u_i + v_j - c_{ij}$.

В ручных расчетах ее можно представить в виде таблицы $m \times n$, справа от которой записываются потенциалы u_i , соответствующие строкам, а внизу — потенциалы v_j , соответствующие столбцам. Разумеется, в программной реализации все это организуется в виде отдельных массивов.

Продемонстрируем работу с матрицей невязок на примере опорного плана, полученного методом северо-западного угла. Около элементов x_{ij} и s_{ij} для справки записаны стоимостные коэффициенты c_{ij} .

Сначала матрица невязок заполняется нулями в тех клетках, которые соответствуют ненулевым перевозкам плана, так как для них по определению потенциалов $s_{ij} = u_i + v_j - c_{ij} = 0$.

Потенциал u_1 полагается равным нулю.

$$X = \begin{pmatrix} 2^{(2)} & 3^{(4)} & 0^{(2)} \\ 0^{(3)} & 1^{(1)} & 6^{(3)} \end{pmatrix}, \quad \begin{array}{c} S \\ \begin{array}{|c|c|c|} \hline 0^{(2)} & 0^{(4)} & 0^{(2)} \\ \hline 0^{(3)} & 0^{(1)} & 0^{(3)} \\ \hline \end{array} \\ \begin{array}{c} u_i \\ 0 \end{array} \\ v_j \end{array}$$

Далее матрица S сканируется построчно. Как только попадаете нуль, делается попытка вычислить второй потенциал из соотношения $u_i + v_j = c_{ij}$, если предварительно вычислен первый.

В результате первого сканирования мы можем вычислить потенциалы $v_1 = c_{11} - u_1 = 2 - 0 = 2$ и $v_2 = c_{12} - u_1 = 4 - 0 = 4$, так как априори известен только $u_1 = 0$:

	<i>S</i>			u_i
	0 (2)	0 (4)	(2)	0
	(3)	0 (1)	0 (3)	
v_j	2	4		

При втором сканировании по известному потенциалу v_2 определяется $u_2 = c_{22} - v_2 = 1 - 4 = -3$:

	<i>S</i>			u_i
	0 (2)	0 (4)	(2)	0
	(3)	0 (1)	0 (3)	-3
v_j	2	4		

Наконец, при третьем сканировании вычисляется оставшийся потенциал $v_3 = c_{23} - u_2 = 3 - (-3) = 6$:

	<i>S</i>			u_i
	0 (2)	0 (4)	(2)	0
	(3)	0 (1)	0 (3)	-3
v_j	2	4	6	

Зная все потенциалы, можно вычислить незаполненные элементы матрицы невязок:

	<i>S</i>			u_i
	0 (2)	0 (4)	4 (2)	0
	-4 (3)	0 (1)	0 (3)	-3
v_j	2	4	6	

Поскольку в матрице невязок имеется положительный элемент s_{13} , то проверяемый план неоптимален. Искусные программисты могут попытаться изобрести более изощренный и быстрый алгоритм.

Замечание. Матрица невязок $S = (s_{ij})$ выполняет функцию управляющей $m + 1$ строки в базовом симплексном алгоритме, только в двумерном варианте. Действительно, для заключительной симплексной таблицы выполняется соотношение (5.10) на с. 117:

$$z_j = \vec{c}_0^T B^{-1} \vec{P}_j = \vec{P}_j^T (B^{-1})^T \vec{c}_0 = \vec{P}_j^T \vec{Y}.$$

Модифицируя это соотношение для двойных индексов, получаем:

$$z_{ij} = \vec{P}_{ij}^T \vec{Y} = u_i + v_j,$$

откуда $z_{ij} - c_{ij} = u_i + v_j - c_{ij} = s_{ij}$. Аналогия полная.

6.5. Переход к новому опорному плану

Если текущий план перевозок неоптимален, следует перейти к новому опорному плану с меньшими транспортными издержками.

Согласно канонам симплексного метода, этот процесс совершается в три шага.

1. Определение вектора, включаемого в базис. Соответственно создается новая перевозка, которая в старом плане отсутствовала.
2. Определение вектора, исключаемого из базиса. Из плана при этом некоторая перевозка исключается.
3. Преобразование плана перевозок.

Создание новой перевозки

В базовом симплексном методе для определения вектора, включаемого в базис, анализируется управляющая $m + 1$ -я строка и находится наибольший положительный ее элемент.

У нас роль управляющей строки играет матрица невязок. Поэтому в базис включается вектор \vec{P}_{ij} , а в план — перевозка x_{ij} ,

соответствующая наибольшему положительному элементу матрицы невязок.

Пример. Как мы установили, начальный опорный план

$$X = \begin{pmatrix} 2 & 3 & 0 \\ 0 & 1 & 6 \end{pmatrix},$$

полученный методом северо-западного угла, неоптимален. Матрица невязок, соответствующая этому плану, содержит один положительный (он же наибольший) элемент $s_{13} = 4$. Следовательно, в базис нужно включить вектор \vec{P}_{13} , то есть организовать новую перевозку x_{13} из склада A_1 в магазин B_3 .

**Исключение
перевозки**

В обычном симплексном методе вектор \vec{P}_l исключается из базиса, если

$$\theta_0 = \frac{x_l}{x_{lk}} = \min_{x_{ik} > 0} \frac{x_i}{x_{ik}},$$

где x_{ik} — коэффициенты разложения включаемого вектора \vec{P}_k по текущему базису, они в готовом виде содержатся в ведущем столбце симплексной таблицы.

В транспортной задаче, в силу специфики матрицы ограничений, коэффициенты разложения легко получить *ex tempore* (лат. — в нужный момент, без подготовки, незамедлительно), построив замкнутый маршрут из ненулевых базисных перевозок, замыкающийся на вводимую перевозку. Так как коэффициенты разложения, стоящие в знаменателе, равны $+1$ для нечетных позиций в маршруте, -1 для четных позиций и 0 для перевозок, не вошедших в маршрут, то поиск минимума по положительным коэффициентам разложения превращается в поиск *минимальной нечетной перевозки в маршруте*.

Пример. Продолжим предыдущий пример. Для того чтобы определить, какую перевозку следует исключить, если

создается перевозка x_{13} , построим замкнутый маршрут из ненулевых перевозок опорного плана, замыкающийся на вводимую перевозку (этот пример мы разбирали на с. 143):

$$\vec{X} = \begin{pmatrix} 2 & \boxed{3} \leftarrow \star \\ 0 & \boxed{1} \rightarrow \boxed{6} \end{pmatrix}.$$

В данном маршруте две нечетных перевозки: 1-я $x_{12} = 3$ и 3-я $x_{23} = 6$, минимальная из них 1-я:

$$\theta_0 = \min(x_{12}, x_{23}) = 3.$$

Следовательно, из базиса исключается вектор \vec{P}_{12} , в новом плане продукция с первого склада во второй магазин перевозиться не будет.

Преобразование плана перевозок

Преобразование опорного плана при переходе к новому базису в симплексном методе происходит по формулам

$$\begin{aligned} x'_i &= x_i - \theta_0 x_{ik}, \\ x'_k &= \theta_0, \end{aligned}$$

где $\theta_0 = \frac{x_l}{x_{lk}}$. Адаптируя это преобразование к транспортной задаче, в которой коэффициенты разложения $x_{ik} \in (+1, -1, 0)$, а θ_0 — минимальная нечетная перевозка, получаем следующее правило преобразования плана перевозок:

$$x'_{ij} = \begin{cases} x_{ij} - \theta_0, & \text{если это — нечетная перевозка в маршруте;} \\ x_{ij} + \theta_0, & \text{если это — четная перевозка в маршруте;} \\ x_{ij}, & \text{если эта перевозка не входит в маршрут;} \\ \theta_0, & \text{если это — новая перевозка.} \end{cases}$$

Пример. Определив параметры преобразования, закончим пример с базами и магазинами. Вычитая из нечетных перевозок $\theta_0 = 3$, прибавляя эту величину к четной перевозке $x_{22} = 1$ и введя новую перевозку $x_{13} = \theta_0 = 3$, получаем

$$\vec{X}' = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 4 & 3 \end{pmatrix}.$$

Именно этот план мы получали методом минимального элемента, а впоследствии установили его оптимальность.

Практический алгоритм

Закончив теоретическое рассмотрение отдельных этапов, приведем весь алгоритм решения транспортной задачи методом потенциалов.

Подготовительный этап

Шаг 1. Балансировка (если необходимо).

Шаг 2. Выбор общего множителя Q для объемов производства и потребления и устранение вырожденности эpsilon-методом.

Шаг 3. Построение исходного опорного плана.

Итерация

Шаг 4. Проверка на оптимальность:

- построение матрицы невязок и нахождение потенциалов;
- проверка матрицы невязок на неотрицательность.

Шаг 5. Определение включаемой перевозки. Ей соответствует максимальный (положительный) элемент матрицы невязок.

Шаг 6. Определение исключаемой перевозки:

- построение циклического (строка — столбец — строка. . .) маршрута, замыкающегося на включаемую перевозку;
- определение минимальной нечетной (в данном маршруте) перевозки θ_0 .

Шаг 7. Преобразование плана перевозок:

- нечетные перевозки уменьшаются на минимальную нечетную перевозку θ_0 ;
- четные перевозки увеличиваются на минимальную нечетную перевозку θ_0 ;
- не входящие в маршрут перевозки остаются прежними;
- вводится новая перевозка θ_0 .

З а к л ю ч и т е л ь н ы й э т а п

Шаг 8. Округление полученного оптимального плана с точностью до Q .

6.6. Исторические замечания

Транспортная задача впервые была поставлена и решена Л. В. Канторовичем (совместно с М. К. Гавуриным) в 1940 г. и опубликована в 1942 г., еще до появления симплексного метода¹. Ими же были предложены понятия потенциалов. На Западе автором транспортной задачи обычно считают Хичкока².

Рассмотренная нами постановка задачи предполагала, что из каждого пункта производства существует прямой путь в каждый

¹ *Канторович Л. В.* О перемещении масс // Докл. АН СССР. — 1942. — Т. 37, № 7–8. — С. 227–229.

² *Hitchcock F. L.* Distribution of a product from several sources to numerous localities // J. Math. Phys. — 1941. — V. 20, No. 2. — P. 224–230.

пункт потребления, и пропускная способность этого пути неограничена. Значительно более реальной представляется модель, в которой рассматривается *транспортная сеть*, примером которой является сеть железных дорог. Пункты производства и потребления размещаются в некоторых узлах сети, а пропускная способность отдельных участков лимитирована заданными числами. На транспортной сети могут быть поставлены различные оптимизационные задачи. Наиболее известная и исторически первая — *задача о наибольшем потоке*, для решения которой Фордом и Фалкерсоном [15] предложен весьма эффективный алгоритм.

Глава 7

Задача о назначении

7.1. Постановка и формализация

Еще одна специфическая задача линейного программирования, заслуживающая изобретения особого метода решения, называется *задачей о назначении* — *assignment problem*.

Имеется n должностей и n претендентов на эти должности. Если i -го человека определить на j -ю должность, то полезность его деятельности оценивается числом c_{ij} .

Задача состоит в том, чтобы назначить людей на должности так, чтобы суммарная польза была наибольшей (или, наоборот, вред наименьшим).

На первый взгляд, математической проблемы тут нет, так как задача решается перебором $n!$ перестановок. Все дело в размерности, при больших n перебор всех вариантов становится нереальным (см. в комментариях к этой главе пример, приведенный Данцигом).

Попытаемся свести эту задачу к задаче линейного программирования.

Введем план назначений

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nn} \end{pmatrix},$$

где $x_{ij} = \begin{cases} 1, & \text{если } i\text{-й человек на } j\text{-м месте,} \\ 0 & \text{в противном случае.} \end{cases}$

Тогда наша задача приобретает вид

$$\begin{aligned} L(X) &= \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min(\max), \\ \sum_{j=1}^n x_{ij} &= 1, \quad i = 1, \dots, n, \\ \sum_{i=1}^n x_{ij} &= 1, \quad j = 1, \dots, n, \\ x_{ij} &\in \{0, 1\}. \end{aligned}$$

Это — почти транспортная задача, но с дополнительными булевыми ограничениями, которые делают задачу нелинейной, так как могут быть записаны в виде $x_{ij}(1 - x_{ij}) = 0$.

К счастью, свойство целочисленности транспортной задачи дает возможность заменить эти условия на более слабые: $x_{ij} \geq 0$, поэтому задача о назначении превращается в обычную транспортную задачу, для решения которой можно использовать как стандартный симплексный метод, так и более эффективный по трудоемкости метод потенциалов.

Однако существует еще более эффективный и очень изящный комбинаторный метод, опирающийся на теорию графов. Он получил название *венгерского* (история его создания приведена в комментариях к этой главе). Венгерский метод основан на нескольких специфических свойствах задачи о назначении, которые мы сейчас рассмотрим.

7.2. Свойства задачи о назначении

Эквивалентные преобразования

Эквивалентным преобразованием (в смысле задачи о назначении) матрицы стоимостей $C = (c_{ij})$ называется преобразование

$$c'_{ij} = c_{ij} + \alpha_i + \beta_j, \quad -\infty < \alpha_i, \beta_j < \infty,$$

т. е. к строкам и столбцам матрицы C можно прибавлять (или вычитать из них) произвольные числа.

Теорема 1. Эквивалентные преобразования матрицы стоимостей не меняют решения задачи о назначении.

Доказательство. Действительно, после эквивалентного преобразования целевая функция

$$\begin{aligned} L(X') &= \sum_i \sum_j c'_{ij} x_{ij} = \sum_i \sum_j (c_{ij} + \alpha_i + \beta_j) x_{ij} = \\ &= \sum_i \sum_j c_{ij} x_{ij} + \sum_i \alpha_i \sum_j x_{ij} + \sum_j \beta_j \sum_i x_{ij} = \\ &= L(X) + \sum_i \alpha_i + \sum_j \beta_j \end{aligned}$$

изменилась на постоянную величину, не зависящую от X . \square

Следствие 1. Задача о назначении на минимум может быть сведена к неотрицательной матрице стоимостей, имеющей в каждой строке и в каждом столбце по нулю.

Доказательство. Определим в каждом столбце минимальный элемент и вычтем этот элемент из столбца:

$$\beta_j = \min_i c_{ij}, \quad c'_{ij} = c_{ij} - \beta_j.$$

В получившейся матрице определим в каждой строке минимальный элемент и вычтем его из строки:

$$\alpha_i = \min_j c'_{ij}; \quad c''_{ij} = c'_{ij} - \alpha_i. \quad \square$$

Следствие 2. *Задача о назначении на максимум может быть сведена к задаче на минимум с неотрицательной матрицей стоимостей, имеющей в каждой строке и в каждом столбце по нулю.*

Доказательство. В первом пункте предыдущего доказательства столбец вычитается из максимального элемента. \square

Независимые нули

Система нулей матрицы называется *независимой*, если в каждой строке и в каждом столбце имеется не более одного нуля. Независимые нули мы будем обозначать звездочками: 0^* .

Теорема 2. *Если в результате эквивалентных преобразований матрица стоимостей приведена к виду с неотрицательными элементами, причем в ней можно выделить n независимых нулей, то решение задачи о назначении на минимум соответствует расположению независимых нулей.*

Доказательство. Для любого плана назначения

$$L(X) = \sum_{i,j} c_{ij}x_{ij} \geq 0.$$

Для оптимального плана $L(X^*) = 0$, т. е. $\forall X L(X^*) \leq L(X)$. \square

Из доказанных свойств можно сделать вывод о том, что для решения задачи о назначении нужно уметь:

- находить наибольшее число независимых нулей в заданной матрице;
- если число независимых нулей меньше порядка матрицы, сделать эквивалентное преобразование, увеличивающее число нулей.

Пример. Венгерский метод мы будем демонстрировать на следующем содержательном примере.

Пятеро друзей, назовем их Анна, Борис, Владимир, Галина, Дарья, решили приготовить обед из пяти блюд: антрекот, борщ, винегрет, гарнир, десерт.

Предположим, что кулинарные дарования наших героев оцениваются такой матрицей полезности:

$$C = \begin{array}{c|ccccc} & a & б & в & г & д \\ \hline A & 3 & 4 & 2 & 2 & 1 \\ B & 4 & 5 & 3 & 1 & 3 \\ B & 4 & 3 & 1 & 1 & 1 \\ Г & 3 & 1 & 2 & 2 & 2 \\ Д & 1 & 3 & 1 & 2 & 1 \end{array}$$

Суммарную полезность нужно максимизировать. Для того чтобы свести дело к минимизации, вычитаем все столбцы из максимальных элементов, а затем в каждой строке вычитаем минимальные элементы. Получится матрица стоимостей, эквивалентная исходной с обратным знаком:

$$-C \sim \begin{array}{c|ccccc} & a & б & в & г & д \\ \hline A & 1 & 1 & 1 & 0^* & 2 \\ B & 0^* & 0 & 0 & 1 & 0 \\ B & 0 & 2 & 2 & 1 & 2 \\ Г & 1 & 4 & 1 & 0 & 1 \\ Д & 3 & 2 & 2 & 0 & 2 \end{array}$$

В этой матрице можно найти несколько независимых нулей, например те, которые отмечены звездочками; но больше трех нулей отметить не удастся. Независимые нули легко увидеть на матрице малой размерности, если же она содержит сотни и тысячи строк и столбцов, то нахождение наибольшего числа независимых нулей превращается в самостоятельную проблему. Вот здесь нам и поможет теория графов.

7.3. Независимые нули и паросочетания

В теории графов есть классическая задача о наибольшем паросочетании, сформулированная еще в 1916 г. венгерским математиком Кёнигом как задача о деревенских свадьбах.

Имеется *двудольный* граф (синонимы: бихроматический граф, граф Кёнига), имеющий две группы вершин: левые и правые. Их можно представить как юношей и девушек, стоящих на противоположных сторонах танцплощадки. Ребра, соединяющие некоторые левые вершины с некоторыми правыми, отображают отношение знакомства и приязни между молодыми людьми (это отношение предполагается взаимным).

Всякий набор несмежных ребер графа называется *паросочетанием*. Таким образом, задача подбора наибольшего числа непесекающихся пар из симпатизирующих друг другу участников сводится к нахождению наибольшего паросочетания в двудольном графе.

Интерпретация на графе

Проблема поиска наибольшего числа независимых нулей матрицы естественным образом формулируется на языке теории графов.

Построим двудольный неориентированный граф, имеющий n левых вершин и n правых вершин, и проведем ребра в соответствии с расположением нулей в матрице (табл. 7.1).

Таблица 7.1. Соответствие между матрицей стоимостей и графом

На матрице	На графе
Строки	Левые вершины
Столбцы	Правые вершины
Нули	Ребра
Независимые нули	Несмежные ребра (паросочетание)
Наибольшая система независимых нулей	Наибольшее паросочетание

Пример. Преобразованная матрица стоимостей из предыдущего примера отображается двудольным графом, имеющим пять левых вершин, соответствующих строкам (персонажам) $A, B, B, Г, Д$, и пять правых вершин, соответствующих столбцам (блюдам) $a, б, в, г, д$. Восемь нулей этой матрицы отображаются соответствующими ребрами графа (рис.7.1).

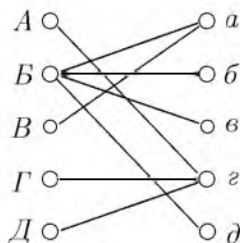


Рис. 7.1. Граф, соответствующий матрице стоимостей

Метод чередующихся цепей

Для нахождения наибольшего паросочетания в двудольном графе венгерские математики Денеш Кёниг и Евгений Эгервари предложили остроумный *метод чередующихся цепей*. В нем используются следующие терминология и обозначения.

Ребра паросочетания называются *сильными* и выделяются на графе жирными линиями. Вершины, инцидентные сильным ребрам (то есть вовлеченные в паросочетание), называются *насыщенными*, на графе они будут отмечаться закрашенными кружками.

Алгоритм метода следующий:

Шаг 1. Строится некоторое исходное паросочетание.

Шаг 2. Ищется *чередующаяся цепь* (слабое ребро — сильное ребро — слабое ребро — и т. д.), соединяющая две ненасыщенные вершины. Следовательно, она начинается со слабого ребра и заканчивается слабым ребром.

Шаг 3. Если такой цепи нет, текущее паросочетание является

наибольшим (этот факт в теории графов устанавливается теоремой Кёнига — Эгервари).

Шаг 4. Если цепь построена, происходит ее инверсия, то есть слабые ребра превращаются в сильные, а сильные — в слабые. Очевидно, несмежность ребер паросочетания при этом сохраняется, но число сильных ребер увеличивается на единицу, так как в чередующейся цепи слабых ребер было больше, чем сильных. Переход на шаг 2.

Пример. Продолжим предыдущий пример. Начнем с исходного паросочетания (см. рис 7.2, а), состоящего из двух ребер: (A, z) и (B, a) , которые соответствуют двум 0^* в матрице стоимостей. Заметим, что данное паросочетание является максимальным, то есть к нему невозможно добавить ни одного ребра, не нарушив их несмежность. Кстати, обратите внимание на тонкость терминологии: определение *максимальный* применительно к множеству используется для указания невозможности добавления к нему новых элементов, а слово *наибольший* характеризует его с точки зрения количества элементов.



Рис. 7.2. Инверсия чередующейся цепи

Далее замечаем, что две ненасыщенные вершины B и d можно соединить чередующейся цепью $B - a - B - d$. Если эту цепь инвертировать: $B - a - B - d$, то получится паросочетание, представленное на рис. 7.2, б. Поскольку в новом графе невозмож-

но построить чередующуюся цепь, соединяющую две ненасыщенные вершины, то данное паросочетание является наибольшим.

7.4. Практический алгоритм венгерского метода

При вычислениях по венгерскому методу сам граф не строят, поиск чередующейся цепи производят непосредственно на матрице стоимостей, для чего некоторые нули помечаются штрихами $0'$.

Для того чтобы последующий алгоритм стал понятным, продолжим таблицу соответствия между матрицей стоимости и графом, ее представляющим (табл. 7.2).

Таблица 7.2. Соответствие между матрицей стоимостей и графом (продолжение)

На матрице	На графе
0^*	Сильное ребро чередующейся цепи
$0'$	Слабое ребро чередующейся цепи
Строка без 0^*	Ненасыщенная левая вершина
Столбец без 0^*	Ненасыщенная правая вершина
Маршрут $0' - \text{ по столбцу } - 0^* - \text{ по строке } - 0' - \text{ по столбцу } - 0^* - \dots - \text{ по строке } - 0'$	Чередующаяся цепь, соединяющая две ненасыщенные вершины

Пример. Чередующуюся цепь $B - a - B - d$, построенную на графе в предыдущем примере, можно проследить на матрице (см. рис. 7.3).

	<i>a</i>	<i>б</i>	<i>в</i>	<i>г</i>	<i>д</i>
<i>A</i>	1	1	1	0*	2
<i>B</i>	0*	0	0	1	0'
<i>B</i>	0'	2	2	1	2
<i>Г</i>	1	4	1	0	1
<i>Д</i>	3	2	2	0	2

Рис. 7.3. Чередующаяся цепь на матрице

После такого подробного объяснения приведем полный алгоритм венгерского метода, решающего задачу о назначении.

П о д г о т о в и т е л ь н ы й э т а п

Шаг 0 для задачи на минимум. Определение минимального элемента в каждом столбце и вычитание его из столбца.

Шаг 0 для задачи на максимум. Определение максимального элемента в каждом столбце и вычитание столбца из него.

Шаг 1. Определение минимального элемента в каждой строке и вычитание его из строки.

Шаг 2. Сканирование строк. Построение начальной системы независимых нулей.

И т е р а ц и я

Шаг 3. В начале итерации производится выделение (пометка знаком +) столбцов, в которых имеются 0^* . Элементы этих столбцов считаются выделенными. Если число помеченных столбцов равно n , работа алгоритма завершается.

Этап 1. Штриховка нулей

Шаг 4. Ищется невыделенный нуль.

Шаг 5. Если такового нет, переход на этап 3 (шаг 12).

Шаг 6. Если есть, то он отмечается штрихом $0'$. Строка, в которой он стоит, отмечается $+$, ее элементы также считаются выделенными.

Шаг 7. Проверяется, есть ли в данной строке 0^* . Если нет, то на этап 2 (шаг 9).

Шаг 8. Если 0^* есть, то снимается выделение с того столбца, где он стоит (плюс обводится квадратиком). Элементы этого столбца вновь считаются невыделенными. Переход на шаг 4.

Этап 2. Построение и инверсия
чередующейся цепи

Шаг 9. От только что построенного $0'$ идем по столбцу к 0^* , затем по строке к $0'$, затем по столбцу к 0^* и т. д. до тех пор, пока чередующаяся цепь не закончится $0'$.

Шаг 10. Инверсия: в построенной цепи $0'$ заменяются на 0^* , а 0^* — на $0'$.

Шаг 11. Стираются все штрихи и знаки $+$. Переход на шаг 3.

Этап 3. Образование нового нуля

Шаг 12. Определяется минимальный невыделенный элемент h .

Шаг 13. От невыделенных строк h вычитается, к выделенным столбцам прибавляется. В результате:

- все невыделенные элементы уменьшатся на h . Образуется по крайней мере один новый невыделенный нуль;

- выделенные элементы останутся без изменения;
- дважды выделенные (перекрестные) элементы увеличатся на h .

Шаг 14. Возврат на этап 1 (шаг 4) с сохранением разметки матрицы (штрихов и $+$).

Пример. Пользуясь описанием венгерского алгоритма, решим наш пример задачи о назначении. Подготовительный этап был выполнен в предыдущих примерах, поэтому начинаем работу с шага 3.

Итерация 1

Шаг 3. Выделяем столбцы 1, 4. Число выделенных столбцов равно 2, что меньше 5 (рис. 7.4, а).

Этап 1

Шаги 4–6–7–8. Сканируем матрицу по строкам, находим первый невыделенный нуль в позиции (22). Штригуем его, выделяем строку 2. В строке 2 есть 0^* в 1-м столбце. Снимаем выделение с 1-го столбца. На шаг 4.

Шаги 4–6–7. Находим невыделенный нуль в позиции (31), так как он снова стал невыделенным. Штригуем его, выделяем строку 3. В строке 3 нет 0^* , переход на этап 2 (шаг 9).

Этап 2

Шаги 9–10–11. От только что заштрихованного $0'$ в позиции (31) движемся по столбцу к 0^* в позиции (21), затем по строке к $0'$ в позиции (22) (рис. 7.4, б).

Инвертируем чередующуюся цепь. Образуются два 0^* в позициях (31) и (22). Стираем штрихи и знаки $+$.

Итерация закончена, возвращаемся на шаг 3.

Итерация 2

Шаг 3. Выделяем столбцы 1, 2, 4. Число выделенных столбцов меньше 5 (рис. 7.4, в).

+					+										
1	1	1	0*	2	1	1	1	0*	2	1	1	1	0*	2	
0*	0	0	1	0	0*	0'	0	1	0	+	0	0*	0	1	0
0	2	2	1	2	0'	2	2	1	2	+	0*	2	2	1	2
1	4	1	0	1	1	4	1	0	1		1	4	1	0	1
3	2	2	0	2	3	2	2	0	2		3	2	2	0	2
	<i>a)</i>				<i>б)</i>						<i>в)</i>				

+	+														
1	0	0	0*	1		1	0'	0	0*	1	+				
1	0*	0'	2	0	+	1	0*	0'	2	0	+				
0*	1	1	1	1		0*	1	1	1	1					
1	3	0	0	0		1	3	0'	0	0	+				
3	1	1	0	1		3	1	1	0	1					
	<i>з)</i>				<i>д)</i>										

+	+	+	+												
1	0	0'	0*	1	+	1	0	0*	0	1					
1	0*	0	2	0'	+	1	0*	0	2	0					
0*	1	1	1	1		0*	1	1	1	1					
1	3	0*	0	0'	+	1	3	0	0	0*					
3	1	1	0'	1	+	3	1	1	0*	1					
	<i>е)</i>				<i>ж)</i>										

Рис. 7.4. Ход решения задачи о назначении венгерским методом

Этап 1

Шаги 4–6–7–8. Сканируем матрицу по строкам, находим первый невыделенный нуль в позиции (23). Штрихуем его, выделяем + строку 2. В строке 2 есть 0^* в 1-м столбце, далее на шаг 8. Снимаем выделение с 1-го столбца. На шаг 4.

Шаги 4–5. Сканируем матрицу по строкам, невыделенных нулей нет. Переход на этап 3 (шаг 12).

Этап 3

Шаги 12–13. Определяется минимальный невыделенный элемент $h = 1$. От невыделенных строк $h = 1$ вычитается, к выделенным столбцам прибавляется. Получается матрица с дополнительными нулями (рис. 7.4, з). На этап 1 (шаг 4).

Этап 1

Шаги 4–6–7–8. Сканируем матрицу по строкам, находим первый невыделенный нуль в позиции (12). Штрихуем его, выделяем строку 1. В строке 1 есть 0^* в 4-м столбце. Снимаем выделение с 4-го столбца. На шаг 4.

Шаги 4–6–7. Сканируем матрицу по строкам, находим первый невыделенный нуль в позиции (43). Штрихуем его, выделяем строку 4. В строке 4 нет 0^* (см. рис. 7.4, д), переход на этап 2 (шаг 9).

Этап 2

Шаги 9–10–11. Возникшая в данный момент ситуация интересна тем, что чередующаяся цепь, начавшись на элементе (43), сразу же и кончается. То есть этот $0'$ просто превращается в 0^* . Стираем все значки выделения и переходим на начало итерации (шаг 3).

И т е р а ц и я 3

Шаг 3. Выделяем столбцы 1, 2, 3, 4. Число выделенных столбцов меньше 5.

Этап 1

Шаги 4–6–7–8. Сканируем матрицу, находим первый невыде-

ленный нуль в позиции (45). Штрихуем его, выделяем строку 4. В этой строке есть 0^* в 3-м столбце. Снимаем выделение со 3-го столбца. На шаг 4.

Шаги 4–6–7–8. Сканируем матрицу, находим ставший невыделенным нуль в позиции (13). Штрихуем его, выделяем строку 1. В этой строке есть 0^* в 4-м столбце. Снимаем выделение с 4-го столбца. На шаг 4.

Шаги 4–6–7–8. Сканируем матрицу, находим первый невыделенный нуль в позиции (25). Штрихуем его, выделяем строку 2. В этой строке есть 0^* во 2-м столбце. Снимаем выделение со 2-го столбца. На шаг 4.

Шаги 4–6–7. Сканируем матрицу, находим первый невыделенный нуль в позиции (54). Штрихуем его, выделяем строку 5. В строке 3 нет 0^* , переход на этап 2 (шаг 9) (см. рис. 7.4, е).

Этап 2

Шаги 9–10–11. От только что заштрихованного $0'$ в позиции (54) движемся по столбцу к 0^* в позиции (14), затем по строке к $0'$ в позиции (13), затем по столбцу к 0^* в позиции (43), затем по строке к $0'$ в позиции (45)

Инvertируем чередующуюся цепь. Итерация закончена, возвращаемся на шаг 3.

И т е р а ц и я 4

Шаг 3. Так как число 0^* равно 5, то задача решена. Назначение претендентов на должности определяется расположением 0^* (см. рис. 7.4, ж).

Достигнутое максимальное значение целевой функции:

$$L = \begin{array}{c|ccccc} & a & б & в & г & д \\ \hline A & 3 & 4 & \boxed{2} & 2 & 1 \\ B & 4 & \boxed{5} & 3 & 1 & 3 \\ B & \boxed{4} & 3 & 1 & 1 & 1 \\ Г & 3 & 1 & 2 & 2 & \boxed{2} \\ Д & 1 & 3 & 1 & \boxed{2} & 1 \end{array} = 15.$$

7.5. Исторические замечания

В цитировавшемся нами историческом эссе (с. 107) Данциг приводит такой образный пример трудоемкости решения задачи о назначении с 70 должностями: «70! — это огромное число, больше чем 10^{100} . Предположим, что мы имели бы ИВМ 370-168¹ во время Большого взрыва 15 миллиардов лет тому назад. Способен ли этот компьютер просмотреть все 70! комбинаций к 1981 г.? Нет! Предположим вместо этого, что он способен проверять 1 миллиард назначений в секунду. Ответ все такой же — "нет". Даже если бы Земля была заполнена такими компьютерами, работающими параллельно, ответ все равно будет "нет". Если, однако, было бы 1050 планет, подобных Земле, заполненных компьютерами с наносекундной длительностью такта и работающими параллельно с момента Большого взрыва до выгорания Солнца, тогда, возможно, ответ будет "да"».

Маленькая Венгрия дала миру многих великих математиков, включая Джона (настоящее имя Янош) фон Неймана. Еще в 1916 г. Денеш Кёниг (König, Dénes; 1884–1944) изучал двудольные графы и сформулировал задачу о наибольших паросочетаниях. Он же в 1936 г. опубликовал первый в мировой литературе учебник по теории графов.

Метод чередующихся цепей был опубликован в 1931 г. на венгерском языке Евгением (Жено) Эгервари (Egerváry, Jenő; 1891–1951) (рис. 7.5), однако из-за экзотического языка публикации не стал достоянием научной общест-венности. Только в 1955 г. американский математик Гарольд Кун (ученик Таккера в Принстонском университете, см. комментарий на с. 128) перевел статью Эгервари на английский язык, доработал метод и, отдавая честь первооткрывателю, назвал его венгерским.



Рис. 7.5.
Евгений Эгервари

¹Самый мощный компьютер из класса мэйнфреймов в то время.

Глава 8

Дискретное линейное программирование

8.1. Классификация задач и методов дискретного программирования

В ряде случаев переменные задачи линейного программирования подвергаются дополнительным ограничениям дискретности:

$$\begin{aligned} L(\vec{X}) &= \sum_{j=1}^n c_j x_j \rightarrow \min, \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m, \\ x_j &\geq 0, \\ x_j &\in R_j \text{ — конечное или счетное множество.} \end{aligned}$$

В зависимости от вида множеств R_j различают следующие задачи:

- целочисленное линейное программирование: $R_j = Z$ — множество целых чисел;

- комбинаторное программирование: $R_j = \{r_1, \dots, r_k\}$ — конечное множество;
- булево программирование: $R_j = \{0, 1\}$.

Замечание 1. Строго говоря, дополнительные ограничения превращают линейную задачу в нелинейную. Например, условие $x_j \in \{0, 1\}$ эквивалентно квадратичному условию $x_j(1 - x_j) = 0$. Однако, учитывая, что остальные условия линейные, принято считать такие задачи задачами линейного дискретного программирования.

Замечание 2. Если число вариантов (планов) конечно, то задачу можно решить методом полного перебора, но при большом числе переменных этот путь явно неэффективен.

Замечание 3. Некоторые задачи линейного программирования обладают *естественной целочисленностью*, гарантирующей целочисленность оптимального плана при целочисленности исходных данных. К ним относится, как мы убедились, транспортная задача и ее разновидность — задача о назначениях. К сожалению, класс таких задач весьма узок, помимо классической транспортной задачи он охватывает лишь некоторые ее обобщения, называемые *задачами транспортного типа*.

Существует множество методов дискретного линейного программирования — от более-менее универсальных до узкоспецифических. Наиболее популярными универсальными методами являются:

- *методы отсечения*, пригодные для решения задач целочисленного линейного программирования;
- *методы ветвей и границы*, широко применяемые для решения комбинаторных задач.

8.2. Методы отсечения

Предположим, что из содержания задачи следует, что переменные должны быть целочисленными.

Первое, что приходит в голову, — решить задачу линейного программирования без ограничения на целочисленность, а потом округлить результат. Однако этот путь порочен, простое округление оптимального нецелочисленного решения может привести к грубым ошибкам. В этом можно убедиться на следующем примере.

Пример.

$$\begin{array}{l}
 L(x_1, x_2) = 2x_1 + x_2 \rightarrow \max, \\
 4x_1 + 5x_2 \leq 20, \\
 3x_1 - 2x_2 \leq 6, \\
 x_1, x_2 \geq 0, \\
 x_i \text{ — целые.}
 \end{array}
 \left. \vphantom{\begin{array}{l} \\ \\ \\ \\ \\ \end{array}} \right\} T_0 \left. \vphantom{\begin{array}{l} \\ \\ \\ \\ \\ \end{array}} \right\} T_{\text{ц}}$$

Здесь мы выделяем две задачи:

- обычную нецелочисленную задачу T_0 ;
- целочисленную задачу $T_{\text{ц}}$.

Решим обе задачи графическим методом. На рис. 8.1 построена область планов D_0 , вырезаемая из плоскости линейными неравенствами, и целочисленная решетка. Если ограничения на целочисленность не учитывать, то оптимальное решение соответствующей задачи T_0 будет $\vec{X}_0^* = (3\frac{1}{23}, 1\frac{13}{23})^T$. Целочисленное же оптимальное решение есть $\vec{X}_{\text{ц}}^* = (2, 2)^T$. Видно, что округление нецелочисленного плана в любую сторону не только не дает правильного решения, но вообще выходит за границы области планов.

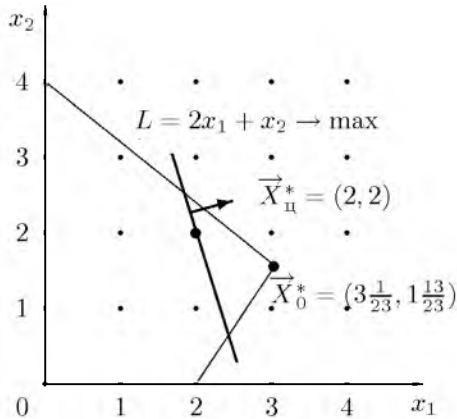


Рис. 8.1. Иллюстрация к примеру

Общая идея

Пусть задана задача T_{Π} целочисленного линейного программирования в канонической форме, в которой выделена подзадача T_0 :

$$\left. \begin{aligned}
 L(\vec{X}) = \sum_{j=1}^n c_j x_j \rightarrow \min, \\
 \sum_{j=1}^n a_{ij} x_j = b_i; \quad i = 1, \dots, m, \\
 x_j \geq 0; \quad j = 1, \dots, n, \\
 x_i \text{ — целые.}
 \end{aligned} \right\} T_0 \left. \vphantom{\sum_{j=1}^n} \right\} T_{\Pi}$$

Общая схема методов отсечения следующая:

- Решается задача T_0 на множестве планов D_0 . Находится оптимальный план \vec{X}_0^* .
- Если план \vec{X}_0^* оказался целочисленным, то задача T_{Π} решена.
- В противном случае строится дополнительное ограничение, которое имеет следующие свойства:

- все целочисленные планы задачи T_{\square} ему удовлетворяют, т. е. оно не затрагивает целочисленных планов задачи T_{\square} (свойство прозрачности);
 - оно отсекает план \vec{X}_0^* , т. е. этот план ему не удовлетворяет (свойство отсечения).
- Дополнительное ограничение (оно называется *правильным отсечением*) добавляется к задаче T_0 , после чего получается задача T_1 на множестве планов D_1 .
 - Решается задача T_1 , находится оптимальный план \vec{X}_1^* , он проверяется на целочисленность и т. д.

В результате получается сходящаяся последовательность вложенных множеств планов

$$D_0 \supset D_1 \supset D_2 \supset \dots \supset D_{\square}$$

и последовательность нецелочисленных решений, сходящаяся к целочисленному решению

$$\vec{X}_0, \vec{X}_1, \vec{X}_2, \dots, \vec{X}_k \rightarrow \vec{X}_{\square}^*.$$

Метод Гомори

Идея методов отсечения довольно проста, но для ее реализации необходимо научиться строить правильные отсечения, а также доказать сходимость метода за конечное число шагов.

Ральф Гомори (Gomory, Ralph E.; p. 1938) предложил для построения правильного отсечения использовать некоторую строчку заключительной симплексной таблицы, с помощью которой было найдено оптимальное нецелочисленное решение. Эта строчка называется *производящей*.

Пусть заключительная симплексная таблица имеет структуру, представленную в табл. 8.1 (для наглядности предполагаем, что заключительный базис состоит из векторов $\vec{P}_1, \dots, \vec{P}_m$).

Таблица 8.1. Структура заключительной симплексной таблицы для задачи T_0

		\overrightarrow{P}_1	\cdots	\overrightarrow{P}_i	\cdots	\overrightarrow{P}_m	\overrightarrow{P}_{m+1}	\cdots	\overrightarrow{P}_n
\overrightarrow{P}_1	x_1^*	1	\cdots	0	\cdots	0	$x_{1,m+1}$	\cdots	x_{1n}
\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots
\overrightarrow{P}_i	x_i^*	0	\cdots	1	\cdots	0	$x_{i,m+1}$	\cdots	x_{in}
\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots
\overrightarrow{P}_m	x_m^*	0	\cdots	0	\cdots	1	$x_{m,m+1}$	\cdots	x_{mn}
	L^*	0	\cdots	0	\cdots	0	$z_{m+1} - c_{m+1}$	\cdots	$z_n - c_n$

Возьмем i -ю строчку этой таблицы с нецелым значением x_i^* (в таблице она выделена прямоугольником). Этой строчке соответствует уравнение

$$0 \cdot x_1 + \cdots + 1 \cdot x_i + \cdots + 0 \cdot x_m + \sum_{j \in N} x_{ij} x_j = x_i^*, \quad (8.1)$$

где x_i — базисная переменная; x_{ij} — коэффициенты разложения небазисных векторов P_j , стоящие в i -й строке (числа); x_j — небазисные переменные; N — множество небазисных переменных, x_i^* — компонента нецелочисленного оптимального плана (число).

Данное уравнение получилось в результате равносильного жорданова преобразования из исходного уравнения, поэтому все планы исходной задачи ему удовлетворяют, и оно не является отсечением.

Попытаемся сделать из него правильное отсечение. Воспользуемся обычными обозначениями для целой и дробной части числа: $z = [z] + \{z\}$. Например, $[-1, 6] = -2$, $\{-1, 6\} = 0, 4$.

Поскольку $[x_{ij}] \leq x_{ij}$ и все $x_j \geq 0$, то из (8.1) следует, что

$$x_i + \sum_{j \in N} [x_{ij}] x_j \leq x_i^*. \quad (8.2)$$

Если все x_i, x_j целочисленны, то левая часть неравенства (8.2) также будет целым числом и

$$x_i + \sum_{j \in N} [x_{ij}] x_j \leq [x_i^*]. \quad (8.3)$$

Вычтем из неравенства (8.3) равенство (8.1):

$$-\sum_{j \in N} \{x_{ij}\} x_j \leq -\{x_i^*\},$$

или, меняя знак неравенства на обратный,

$$\sum_{j \in N} \{x_{ij}\} x_j \geq \{x_i^*\}. \quad (8.4)$$

Так как (8.3) было справедливо для целочисленных планов, то следующее из него неравенство (8.4) также прозрачно для целочисленных планов. Однако легко убедиться, что оно не выполняется для оптимального нецелочисленного плана.

Действительно, в оптимальном плане все небазисные переменные x_j равны 0. Это приводит (8.4) к выражению

$$0 \geq \{x_i^*\},$$

что невозможно по определению дробной части для нецелых чисел. Таким образом, ограничение (8.4) является правильным отсечением.

Заслуга Гомори не только в том, что он предложил путь построения отсечения, но и в том, что он доказал сходимость метода за *конечное* число шагов (мы это доказательство опускаем).

В реализации метода Гомори осталось определить, какую из строчек симплексной таблицы взять для построения правильного отсечения, если несколько компонент оптимального плана нецелочисленны.

Возможны несколько правил:

- брать первую по порядку строку;

- брать строку с наибольшим значением дробной части.

Эксперименты показали, что оба варианта сравнимы по числу итераций, приводящих к целочисленному решению.

Пример. Решим наш пример целочисленного линейного программирования методом Гомори. Запишем ограничения задачи T_0 в форме равенств:

$$\begin{aligned} L &= 2x_1 + x_2 && \rightarrow \max, \\ 4x_1 + 5x_2 + x_3 &= 20, \\ 3x_1 - 2x_2 + x_4 &= 6, \\ x_1, \dots, x_4 &\geq 0. \end{aligned}$$

Решаем ее симплексным методом. Матрица условий имеет естественный единичный базис, поэтому метод искусственного базиса не понадобится.

Составляем исходную симплексную таблицу с учетом того, что при переходе к канонической форме нужно сменить знаки стоимостных коэффициентов. Определяем ведущий элемент для первой итерации (он выделен прямоугольником):

$$\begin{array}{c|ccccc} & \vec{X} & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 & \vec{P}_4 \\ \hline \vec{P}_3 & 20 & 4 & 5 & 1 & 0 \\ \vec{P}_4 & 6 & \boxed{3} & -2 & 0 & 1 \\ \hline & 0 & 2 & 1 & 0 & 0 \end{array}$$

Вторая итерация:

$$\begin{array}{c|ccccc} & \vec{X} & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 & \vec{P}_4 \\ \hline \vec{P}_3 & 12 & 0 & \boxed{\frac{23}{3}} & 1 & -\frac{4}{3} \\ \vec{P}_1 & 2 & 1 & -\frac{2}{3} & 0 & \frac{1}{3} \\ \hline & -4 & 0 & \frac{5}{3} & 0 & -\frac{2}{3} \end{array}$$

Третья итерация:

$$\begin{array}{c|cccc}
 & \vec{X} & \vec{P}_1 & \vec{P}_2 & \vec{P}_3 & \vec{P}_4 \\
 \hline
 \vec{P}_2 & 1\frac{13}{23} & 0 & 1 & \frac{3}{23} & -\frac{4}{23} \\
 \vec{P}_1 & 3\frac{1}{23} & 1 & 0 & \frac{2}{23} & \frac{5}{23} \\
 \hline
 & -7\frac{15}{23} & 0 & 0 & -\frac{7}{23} & -\frac{6}{23}
 \end{array}$$

Получили оптимальное решение задачи T_0 :

$$\vec{X}_0^* = \left(3\frac{1}{23}, 1\frac{13}{23}, 0, 0 \right)^T.$$

Оно получилось нецелочисленным и, естественно, равным тому, которое получилось графическим методом. Следовательно, необходимо построить дополнительное правильное отсечение.

Будем следовать тому правилу, который предписывает выбрать в качестве производящей строку, соответствующую компоненте оптимального плана с наибольшим значением дробной части. Это будет первая строка в заключительной симплексной таблице, она соответствует базисному вектору \vec{P}_2 , поэтому мы полагаем ее индекс i равным двум:

$$i = 2, \quad x_i^* = 1\frac{13}{23}, \quad \{x_i^*\} = \frac{13}{23}.$$

Множество индексов небазисных переменных $N = \{3, 4\}$, коэффициенты разложения небазисных векторов \vec{P}_3, \vec{P}_4 , стоящие в производящей строке, и их дробные части:

$$\begin{aligned}
 x_{23} &= \frac{3}{23}, & \{x_{23}\} &= \frac{3}{23}; \\
 x_{24} &= -\frac{4}{23}, & \{x_{24}\} &= \frac{19}{23}.
 \end{aligned}$$

Строим правильное отсечение:

$$i = 2, \quad \sum_{j=3,4} \{x_{ij}\}x_j \geq \{x_i^*\} : \quad \frac{3}{23}x_3 + \frac{19}{23}x_4 \geq \frac{13}{23}.$$

Добавляем его к исходным ограничениям, получаем задачу T_1 :

$$\begin{aligned} 2x_1 + x_2 &\rightarrow \max, \\ 4x_1 + 5x_2 + x_3 &= 20, \\ 3x_1 - 2x_2 + x_4 &= 6, \\ 3x_3 + 19x_4 &\geq 13, \\ x_1, \dots, x_4 &\geq 0. \end{aligned}$$

Попробуем визуализировать задачу T_1 . Для этого сначала избавляемся от неравенства, вводя дополнительную неотрицательную переменную x_5 :

$$\begin{aligned} 2x_1 + x_2 &\rightarrow \max, \\ 4x_1 + 5x_2 + x_3 &= 20, \\ 3x_1 - 2x_2 + x_4 &= 6, \\ 3x_3 + 19x_4 - x_5 &= 13, \\ x_1, \dots, x_5 &\geq 0. \end{aligned}$$

Затем обработаем матрицу ограничений калькулятором Жордана с целью получения единичной подматрицы (см. пример на с. 69):

$$\begin{aligned} 2x_1 + x_2 &\rightarrow \max, \\ 4x_1 + 5x_2 + x_3 &= 20, \\ 3x_1 - 2x_2 + x_4 &= 6, \\ 3x_1 - x_2 + x_5 &= 7, \\ x_1, \dots, x_5 &\geq 0. \end{aligned}$$

Отбрасывая последние три переменные с нулевыми стоимостными коэффициентами, получаем задачу с двумя переменными и

ограничениями-неравенствами:

$$\begin{aligned} 2x_1 + x_2 &\rightarrow \max, \\ 4x_1 + 5x_2 &\leq 20, \\ 3x_1 - 2x_2 &\leq 6, \\ 3x_1 - x_2 &\leq 7, \\ x_1, x_2 &\geq 0. \end{aligned}$$

На рис. 8.2 хорошо видно, как дополнительное третье ограничение $3x_1 - x_2 \leq 7$ отсекает \vec{X}_0^* , не затрагивая узлов целочисленной решетки, попавших в область планов D_0 .

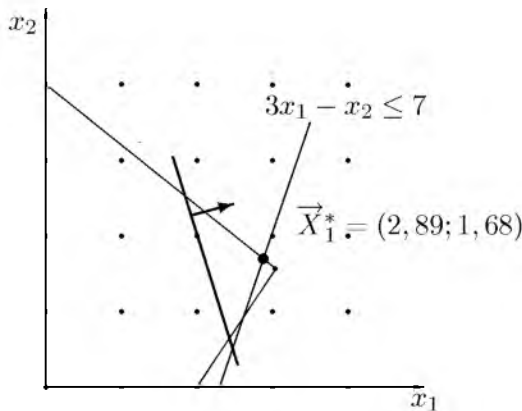


Рис. 8.2. Правильное отсечение и решение задачи T_1

Решая задачу T_1 , получаем $\vec{X}_1^* = (2, 894; 1, 684, 0, 0)^T$ (первые две координаты вектора \vec{X}_1^* представлены на рис. 8.2).

За недостатком места мы не можем столь же подробно описать следующие несколько итераций, в результате которых в конце концов найдется $\vec{X}_n^* = (2, 2)$.

Как видим, метод Гомори хоть и точный, но достаточно трудоемкий. По этой причине в эпоху слабых компьютеров он не был широко востребован. В настоящее время, когда мощность обычного ноутбука превышает вычислительные возможности суперкомпьютеров прошлых десятилетий, отношение к методу меняется.

Замечание 1. Приведенный алгоритм (он называется *Первым алгоритмом Гомори*) предназначен для нахождения **п о л н о с т ь ю** **ц е л о ч и с л е н н ы х** решений, охватывающих и те дополнительные переменные, которые вводились для приведения задачи к канонической форме. В связи с этим нужно проявлять определенную осторожность. Например, если бы первое неравенство в нашем примере имело вид $4x_1 + 5x_2 \leq 20\frac{1}{2}$, то, приводя его к равенству путем введения переменной x_3 , мы получили бы уравнение $4x_1 + 5x_2 + x_3 = 20\frac{1}{2}$, которое не имеет решения ни при каких целых значениях переменных. Поэтому, прежде чем вводить дополнительную переменную, следует избавиться от дробей, умножив неравенство на соответствующую константу: $8x_1 + 10x_2 \leq 41$. Теперь неравенство приводится к виду $8x_1 + 10x_2 + x_3 = 41$, и целочисленность x_1, x_2 автоматически дает целочисленность дополнительной переменной x_3 .

Замечание 2. Для решения частично целочисленных задач существует несколько более сложный *Второй алгоритм Гомори*, с которым можно познакомиться в [12].

8.3. Метод ветвей и границ

Метод ветвей и границ (Branch & Bound) — обобщенное название специального подхода к решению комбинаторных задач оптимизации с конечным числом вариантов.

Метод ветвей и границ является переборным, как правило, он позволяет сократить число анализируемых вариантов по сравнению с полным перебором, хотя и не гарантирует этого.

Общая схема В основе метода лежат две процедуры: 1) процедура ветвления; 2) процедура оценивания.

Процедура ветвления состоит в рекурсивном разбиении мно-

жества вариантов (планов) на подмножества меньшей мощности. Полученные подмножества образуют *дерево поиска*, корнем которого является множество всех планов, а конечными вершинами — конкретные планы задачи.

Алгоритм состоит из последовательности *прямых и обратных ходов* по дереву поиска.

Во время прямого хода происходит движение вниз по уровням дерева поиска, при этом каждый раз выбирается наиболее перспективная ветвь. Перспективность ветви определяется с помощью *процедуры оценивания*, которая должна давать оценку сверху (при поиске максимума) или снизу (при поиске минимума) целевой функции на данном множестве планов. Прямой ход заканчивается на конечной вершине — конкретном плане задачи.

Во время обратного хода алгоритма происходит движение вверх по дереву поиска, при этом для каждой вершины дерева — подмножества планов — определяется *рекорд*, т. е. фактически достигнутое значение целевой функции на данном подмножестве.

Сокращение поиска по сравнению с полным перебором основано на очевидной идее: если перспективная оценка на некоторой исследуемой ветви хуже уже достигнутого рекорда на параллельной ветви, то исследуемая ветвь заведомо является бесперспективной, она исключается из рассмотрения.

Процедуры ветвления и оценивания изобретаются для каждой отдельной задачи, они и определяют конкретную реализацию метода.

Булево линейное программирование

принимают только два значения: 0 и 1.

Идея метода ветвей и границ очень ярко проявляется в *булевом линейном программировании*, когда переменные задачи при-

Формальная постановка задачи:

$$L(\vec{X}) = \sum_{j=1}^n c_j x_j \rightarrow \min (\max),$$

$$\sum_{j=1}^n a_{ij} x_j \begin{cases} = \\ \leq \\ \geq \end{cases} b_i, \quad i = 1, \dots, m,$$

$$x_j \in \{0, 1\}.$$

В этом случае дерево поиска строится очевидным образом. Кроме начального нулевого, в нем имеется столько уровней, сколько переменных в задаче, то есть n , при этом на i -м уровне происходит ветвление по двум направлениям: $x_i = 0$ и $x_i = 1$ (рис. 8.3).

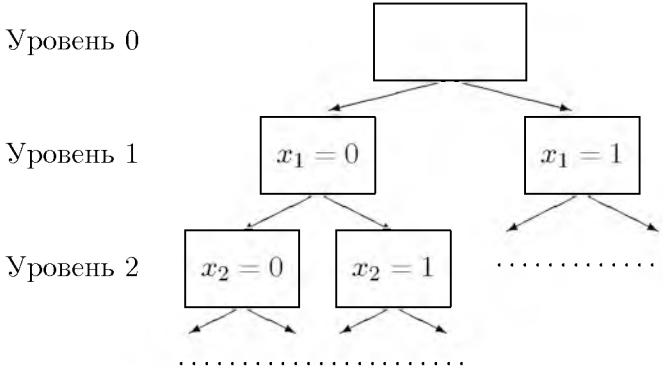


Рис. 8.3. Ветвление в булевом программировании

Процедура оценивания также напрашивается сама собой: в качестве оценки принять оптимальное значение целевой функции *без ограничения на дискретность* переменных, то есть заведомо лучшее значение.

**Задача
о ранце**

Рассмотрим метод ветвей и границ в булевом программировании на примере классической задачи, известной в литературе как *задача о ранце* (*Knapsack Problem*). Ее постановка в контексте известной сказки из «Тысячи и одной ночи» выглядит следующим образом. Али-баба попал в пещеру к разбойникам, где перед ним лежат n драгоценных предметов A_1, \dots, A_n . Каждый i -й предмет характеризуется стоимостью c_i , а также объемом v_i и весом p_i . Али-баба был бы рад забрать их все, но объем имеющегося рюкзака (ранца) ограничен величиной V , а максимальный вес, который можно унести, равен P . Возникает задача отбора предметов, имеющих наибольшую суммарную стоимость при ограничениях на объем и вес:

$$\begin{aligned} L(\vec{X}) &= \sum_{j=1}^n c_j x_j \rightarrow \max, \\ \sum_{j=1}^n v_j x_j &\leq V, \\ \sum_{j=1}^n p_j x_j &\leq P, \\ x_j &\in \{0, 1\}. \end{aligned}$$

Количество ограничений можно увеличить, введя дополнительные характеристики предметов, например радиоактивность и т. п., суть задачи от этого не меняется.

Пример. Пусть имеется всего три предмета, характеристики которых приведены в табл. 8.2.

Формальная постановка задачи:

$$\begin{aligned} L &= 3x_1 + 4x_2 + 5x_3 \rightarrow \max, \\ 7x_1 + 3x_2 + 5x_3 &\leq 10, \\ 2x_1 + 6x_2 + 5x_3 &\leq 11, \\ x_j &\in \{0, 1\}. \end{aligned}$$

Таблица 8.2. Исходные данные для задачи о ранце

	A_1	A_2	A_3	Ограничение
Стоимость	3	4	5	max
Объем	7	3	5	10
Вес	2	6	5	11

Этот пример предельно простой, имеется всего $2^3 = 8$ вариантов, поэтому мы можем проследить работу метода ветвей и границ с начала до конца.

Каждая вершина дерева поиска, с которым мы будем работать, изображена прямоугольником. В нижней части его записана подзадача линейного программирования, которая получается, когда часть переменных приняла фиксированные значения (ограничения на неотрицательность переменных мы для экономии места опускаем). Поэтому на нулевом уровне мы имеем исходную задачу, а на третьем — терминальные планы задачи при всех известных значениях переменных. Для каждой вершины, кроме нулевой и терминальных, в верхней части прямоугольника указываются значения оценки \bar{L} и рекорда L^* на множестве планов, соответствующем данной ветви.

Работа алгоритма начинается с прямого хода с нулевого по терминальный уровень (см. рис. 8.4). Находясь в вершине уровня i , мы формулируем две задачи, соответствующие выбору $x_i = 0$ и $x_i = 1$ без ограничений на целочисленность. Так как число переменных не более двух, эти задачи можно решить простейшим графическим методом. Полученные оптимальные значения дадут оценки для левой и правой исходящих ветвей.

Например, для вычисления оценки по левой ветви на первом уровне имеем следующую задачу линейного программирования (она записана в соответствующем прямоугольнике):

$$\begin{aligned}
 L &= 4x_2 + 5x_3 \rightarrow \max, \\
 3x_2 + 5x_3 &\leq 10, \\
 6x_2 + 5x_3 &\leq 11, \\
 x_2, x_3 &\geq 0.
 \end{aligned}$$

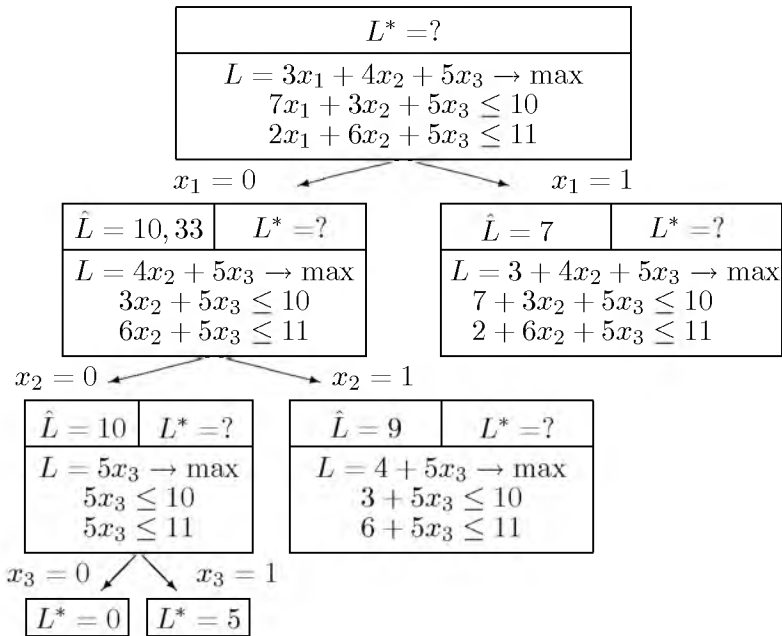


Рис. 8.4. Прямой ход с нулевого на третий уровень

Решая ее, получаем $x_2^* = 0,33$; $x_3^* = 1,8$; $\hat{L} = 10,33$.

В процессе прямого хода всегда спускаемся по той ветви, у которой оценка больше (задача на максимум). Таким образом, идя на первом уровне налево, потом снова налево и, наконец, направо, мы приходим к двум терминальным планам: $\vec{X}_{000} = (0,0,0)^T$, для которого $L^* = L(\vec{X}_{000}) = 0$, и $\vec{X}_{001} = (0,0,1)^T$, для которого $L^* = L(\vec{X}_{001}) = 5$.

Далее делаем шаг обратного хода с третьего уровня на второй (см. рис. 8.5). Достигнутое наибольшее терминальное значение целевой функции становится рекордом на левой ветви второго уровня. Теперь мы можем сравнить две ветви на втором уровне. Слева имеем достигнутый рекорд $L^* = 5$ (хотя оценка была 10), а

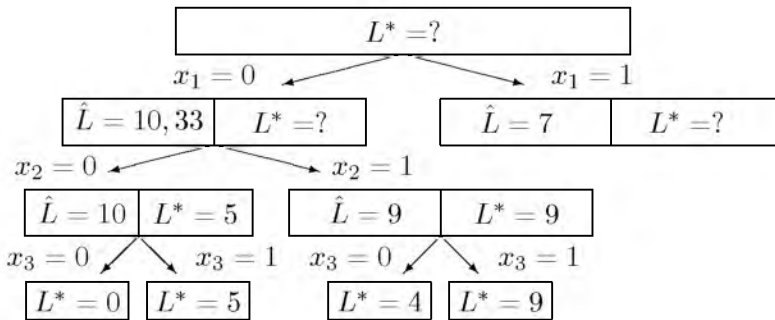


Рис. 8.5. Обратный ход с 3-го на 2-й уровень

справа — оценку сверху $\hat{L} = 9$. То есть перспектива на правой ветви имеется. Это не значит, конечно, что оценка будет достигнута, но исследовать правую ветвь придется.

Делаем шаг прямого хода со второго на третий уровень. Вычисляем оба терминальных значения целевой функции.

Делаем шаг обратного хода с третьего уровня на второй (рис. 8.5). В качестве рекорда выбираем максимальное значение по обоим нисходящим ветвям, то есть $L^* = \max(4, 9) = 9$. Как видим, значение оценки здесь достигнуто.

Продолжая исследование дерева поиска, делаем шаг обратного хода со второго уровня на первый (см. рис. 8.6). В качестве рекорда левой ветви устанавливаем $L^* = \max(5, 9) = 9$. Теперь можно сравнить две ветви на первом уровне. Слева имеем достигнутый рекорд $L^* = 9$, а справа — оценку сверху $\hat{L} = 7$. Таким образом, правая ветвь явно бесперспективна, и мы делаем шаг обратного хода на нулевой уровень. Поднимая рекорд на самый верх, имеем окончательно для всей задачи $L^* = 9$.

Для того чтобы определить оптимальный план, нужно спуститься по дереву сверху вниз по вершинам, для которых сохраняется рекорд $L^* = 9$, запоминая ветви, по которым происходит спуск. У нас $x_1^* = 0$, $x_2^* = 1$, $x_3^* = 0$.

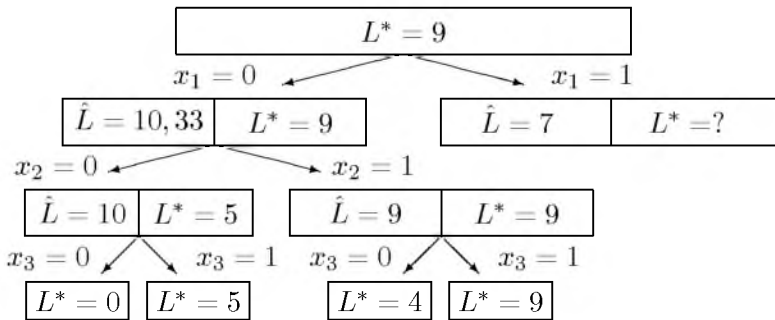


Рис. 8.6. Завершение обратного хода

В процессе решения задачи нам пришлось просмотреть четыре терминальных варианта, что составляет 50% от их общего числа. В целом же эффективность метода (степень сокращения перебора) целиком зависит от качества процедуры оценивания. Если оценка тривиальна, то метод сведется к полному перебору вариантов.

8.4. Исторические замечания

Автор метода отсеечения Ральф Гомори (Gomory, Ralph E.; р. 1929) известен не только как математик, но и как организатор науки высокого уровня (рис. 8.7). Получив ученую степень в Принстонском университете, он с 1954 по 1957 г. служил в Военно-морском флоте, где занимался исследованием операций. С 1959 г. в течение 30 лет работал в исследовательском отделении компании ИВМ, прошел путь до старшего вице-президента по исследованиям. С его помощью компания превратилась в одно из ведущих иссле-



Рис. 8.7. Р. Гомори

довательских учреждений в мире. С 1989 г. возглавляет Alfred P. Sloan Foundation. Член президентского Совета по науке и технологиям.

Метод ветвей и границ предложен в 1960 г. американскими математиками Лэнд (Ailsa H. Land) и Дойг (Alison G. Doig) для решения частной задачи дискретного линейного программирования¹, но впоследствии был распространен на другие комбинаторные задачи.

¹Land A. H., Doig A. G. An automatic method of solving discrete programming problem // *Econometrica*. — 1960. — V. 28, No 3. — P. 497–520.

Литература

1. *Перегудов Ф. И., Тарасенко Ф. П.* Основы системного анализа. — 2-е изд. — Томск: Изд-во НТЛ, 1997. — 396 с.
2. *Вагнер Г.* Основы исследования операций: Пер. с англ. — М.: Мир, 1972. — Т. 1. — 335 с.; 1973. — Т. 2. — 488 с.; 1973. — Т. 3. — 501 с.
3. *Таха Х. А.* Введение в исследование операций. — 7-е изд.: Пер. с англ. — М.: Изд. дом «Вильямс», 2005. — 912 с.
4. *Кельтон В., Лоу А.* Имитационное моделирование. Классика CS. — 3-е изд.: Пер. с англ. — СПб.: Питер; Киев: Изд. группа BHV, 2004. — 847 с.
5. *Гермейер Ю. Б.* Введение в теорию исследования операций. — М.: Наука, 1971. — 383 с.
6. *Пфанцагль И.* Теория измерений. — М.: Мир, 1976.
7. *Гасс С.* Линейное программирование (методы и приложения): Пер. с англ. — М.: ГИФМЛ, 1964.
8. *Черчмен У., Акоф Р., Арноф Л.* Введение в исследование операций: Пер. с англ. — М.: Наука, 1967. — 488 с.
9. *Канторович Л. В.* Математические методы организации и планирования производства. — Л.: Изд-во ЛГУ, 1939. — 67 с.

10. *Данциг Дж.* Линейное программирование, его применения и обобщения: Пер. с англ. — М.: Прогресс, 1966.
11. *Диксон П.* Фабрики мысли: Пер. с англ. — М.: Прогресс, 1976. — 451 с.
12. *Корбут А. А., Финкельштейн Ю. Ю.* Дискретное программирование. — М.: Наука, 1969.
13. *Юдин Д. Б., Гольштейн Е. Г.* Задачи и методы линейного программирования. — М.: Сов. радио, 1961. — 736 с.
14. *Гольштейн Е. Г., Юдин Д. Б.* Новые направления в линейном программировании. — М.: Сов. радио, 1966.
15. *Форд Л., Фалкерсон Д.* Потоки в сетях: Пер. с англ. — М.: Мир, 1966. — М.: Сов. радио, 1966.

Учебное пособие

Гладких Борис Афанасьевич

**МЕТОДЫ ОПТИМИЗАЦИИ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
ДЛЯ БАКАЛАВРОВ ИНФОРМАТИКИ**

**Часть I. Введение в исследование операций.
Линейное программирование**

Редактор *Н. И. Шидловская*
Вёрстка *Б. А. Гладких*
Дизайн *Д. В. Фортеса*

Изд. лиц. ИД № 04000 от 12.02.2001. Подписано к печати 05.05.09.
Формат 60 × 84¹/₁₆. Бумага офсетная. Печать офсетная.
Гарнитура «Computer Modern Super». Усл.-печ. л. 11,63. Уч.-изд. л. 13,1.
Тираж 200 экз.

ООО «Издательство научно-технической литературы»
634050, г. Томск, пл. Ново-Соборная, 1, тел. (3822) 533-335

Отпечатано в типографии ЗАО «М-Принт», г. Томск, ул. Пролетарская, 38/1