ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

Д.А. ЖОЛОБОВ

ВВЕДЕНИЕ
В МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Рекомендовано УМО «Ядерные физика и технологии» в качестве учебного пособия для студентов высших учебных заведений

Москва 2008

УДК 519.85(075)
ББК 22.18я7
Ж79

Жолобов Д.А. Введение в математическое программирование: Учебное пособие.-М.:МИФИ, 2008.-376 с.

Рассмотрены методы решения задач линейного и дискретного программирования. Теория линейного программирования содержит обоснование симплекс-метода, теорию двойственности. Рассмотрены методы увеличения вычислительной эффективности решения задач большой размерности. Раздел, посвященный дискретному программированию, включает в себя транспортную задачу, методы решения задач целочисленного и булевого программирования. Представлен подход для решения прикладных задач, основанный на идеях динамического программирования.

Данное учебное пособие предназначено студентам, обучающимся по специальности «Прикладная математика», и может быть использовано при изучении дисциплины «Методы оптимизации». Пособие подготовлено в рамках Инновационной образовательной программы.

Рецензент д-р техн. наук, проф. Ю.П. Кулябичев

ISBN 978-5-7262-1082-7

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В широком круге научных, экономических и управленческих проблем в процессе анализа возникают разнородные задачи поиска оптимального значения той или иной величины при соблюдении некоторого набора требований к самой величине, либо другим факторам, находящимся в предметной области исследуемой проблемы. Большое распространение такие задачи получили, в частности, в области ядерной физики и энергетики, как в научных [1,4,6], так и в экономических [5,9] аспектах.

В самом общем виде подобные задачи аналитически могут быть представлены в виде следующей системы:

$$\varphi_0(x_1,x_2,\ldots,x_n)\rightarrow extr \tag{1}$$

$$\varphi_i(x_1,x_2,\ldots,x_n)=0;\ \ i=1,2,\ldots,m \tag{2}$$

$$(x_1,x_2,\ldots,x_n)\in G. \tag{3}$$

Здесь $(x_1,x_2,\ldots,x_n)$ – вектор переменных – параметров управления; G – множество, элементы которого обладают определенными свойствами; $\varphi_0(x_1,x_2,\ldots,x_n)$ – количественный показатель качества решения – <u>*целевая функция*</u>.

Область допустимых решений задается системой ограничений (2) и множеством G (3).

В задаче требуется из всех допустимых решений, т.е. решений, удовлетворяющих ограничениям (2) и (3), найти такое, которое обеспечивает экстремальное значение целевой функции – максимальное или минимальное, в зависимости от содержательной постановки задачи

Задача, сформулированная в виде (1)-(3), является задачей <u>*математического программирования*</u> – раздела прикладной математики, в котором разрабатываются методы поиска условного экстремума функции многих переменных.

Важнейшей особенностью задачи, сформулированной в общем виде (1)-(3), является отсутствие общего, универсального метода ее решения, за исключением, разве что, полного перебора всех допустимых решений.

Следует отметить, что для задач даже небольшой размерности полный перебор является нереализуемой процедурой для современных супер-ЭВМ.

Поэтому, в зависимости от конкретного вида функций $\varphi_i(x_1,x_2,...,x_n)=0$ ($i=0,1,2,...,m$) и от вида множества G, экстремальная задача (1)-(3) относится к тому или иному классу задач математического программирования. Для каждого такого класса в рамках соответствующего раздела математического программирования разрабатываются свои методы решения.

Так, если φ_0 и φ_i ($i=1,2,...,m$) – линейные функции параметров управления, т.е.

$$\varphi_0=\sum_{j=1}^{n} c_j x_j \ , \ \varphi_i=\sum_{j=1}^{n} a_{ij} x_j - a_i \ ,$$

а G - положительный ортант ($x_j \geq 0, j=1,2,...,n$) или гиперпараллелепипед $(a_j^- \le x_j \le a_j^+; j=\overline{1,n})$ евклидова n-мерного пространства, задача (1)-(3) является задачей *линейного программирования*.

Если G – некоторое подмножество точек евклидова n-мерного пространства, все или отдельные координаты которых принимают целочисленное значение, задача (1)-(3) является задачей *целочисленного программирования*.

Важнейшим подклассом задач целочисленного программирования являются задачи *булевого программирования*, в которых переменные могут принимать только одно из двух значений – 0 или 1.

Если G – выпуклая область, а φ_0 и φ_i ($i=1,2,...,m$) – выпуклые функции параметров управления, задача (1)-(3) является задачей *выпуклого программирования*.

Выпуклое программирование – основное направление *нелинейного программирования*. Особенность выпуклых задач – это совпадение глобального оптимума с локальным.

Невыпуклые (многоэкстремальные) задачи очень сложны и трудоемки. Конструктивных методов решения этих задач практически нет.

Если параметры задачи – недетерминированные величины, а именно это часто имеет место на практике, речь идет о задаче *стохастического программирования*.

Наконец, если искомые параметры оптимального управления представляют собой функции (например, от времени или от номера этапа принятия решения), соответствующий класс задач называется задачами *динамического программирования*.

Приведенная классификация задач математического программирования является очень общей. В рамках перечисленных направлений существуют и более узкие разделы. Более того, на практике встречаются задачи, обладающие свойствами разных классов (например, нелинейные задачи, отдельные переменные которых – булевы).

1. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

1.1. Постановка прикладных задач в терминах линейного программирования

Термин *линейное программирование* впервые возник в 1951 г. в работах американских математиков Дж. Данцига и Т. Купманса [10].

Первые же исследования линейных моделей относятся к концу 30-х годов, когда ленинградский математик Л.В. Канторович сформулировал ряд линейных условно-экстремальных задач и разработал способ их решения, довольно близкий к симплекс-методу.

В 1975 г. Канторовичу и Купмансу была присуждена Нобелевская премия. Уже одно это – признание роли оптимизационных моделей в науке и практике.

В настоящее время линейные модели широко используются во всем мире в практике планирования и управления объектами самой различной природы.

Популярность линейного программирования связана не только с развитостью соответствующего математического аппарата, но и с тем обстоятельством, что широкий класс реальных объектов управления достаточно хорошо и, что самое главное, эти объекты достаточно ясно описываются именно линейными моделями.

Для линейных моделей характерно свойство аддитивности и пропорциональности.

Пример 1.1

> *Аддитивность*: продали 4 шила по 2 рубля и 5 кусков мыла по 4 рубля. Имеем доход 28=2*4+4*5.
>
> *Пропорциональность*: на изготовление одного утюга идет 2 кг чугуна. Значит, на изготовление 4-х утюгов пойдет 8 кг чугуна.

Построение математических моделей на основании словесного описания задачи является сложноформализуемым процессом. Однако можно выделить следующие общие этапы:

1. Идентификация переменных;
2. Формализация цели – построение целевой функции в терминах переменных;
3. Формализация ограничений – построение соответствующих математических выражений в терминах переменных;
4. Спецификация переменных (определение множества G).

Пример 1.2

Атомная электростанция состоит из двух энергоблоков: Э1 и Э2. В силу конструктивных различий и разного уровня оснащения прибыль (упрощенно будем считать ─ разницу между рыночной ценой и издержками на выработку), которую приносит продажа одного мегаватт-часа электричества, отличается в зависимости от энергоблока. Для производства электроэнергии используются три вида сырья – S1, S2 и S3. Количественные характеристики :

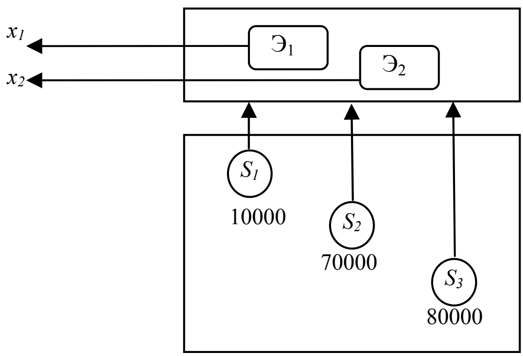
Запас сырья вида S1 ─ 100000 ед.

Запас сырья вида S2 ─ 70000 ед.

Запас сырья вида S3 ─ 80000 ед.

Количество единиц сырья вида S_i (i=1.2.3), необходимое для производства одного мегаватт-часа на энергоблоке $Э_j$ (j=1,2) , а также прибыль от его реализации представлены в таблице:

	Э1	Э2
Сырье S1	2	6
Сырье S2	1	1
Сырье S3	2	2
Прибыль	700 руб.	1400 руб.



Необходимо определить план выработки электроэнергии каждым энергоблоком на один месяц, чтобы обеспечить максимальную прибыль АЭС.

Математическая модель задачи:

$$\begin{cases} 700x_1 + 1400x_2 \to \max \\ 2x_1 + 6x_2 \le 100000 \\ x_1 + x_2 \le 70000 \\ 2x_1 + 2x_2 \le 80000 \\ x_1, x_2 \ge 0. \end{cases}$$

Решение:

Суммарная прибыль: 31.5 млн руб.

Выработка электроэнергии

Э1: 35000 МВтч

Э2: 5000 МВтч

Затраты сырья

S1 : 100000 ед.

S2 : 40000 ед.

S3 : 80000 ед.

Пример 1.3

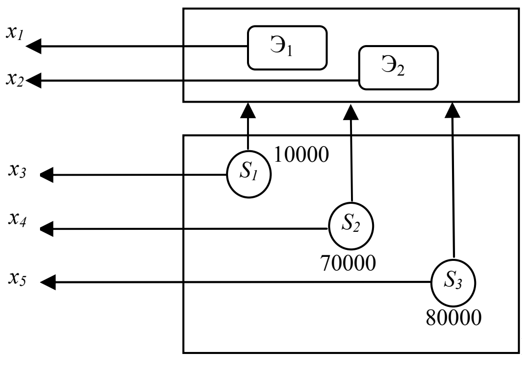
Условия примера 1.2 сохраняются, но предполагаем, что сырье можно продавать другим АЭС по ценам:

30 руб. – за единицу сырья вида S1;

40 руб. – за единицу сырья вида S2;

100 руб. – за единицу сырья вида S3.

Естественно, продажа сырья увеличивает прибыль электростанции.



Математическая модель задачи:

$$\begin{cases} 700x_1 + 1400x_2 + 30x_3 + 40x_4 + 100x_5 \to \max \\ 2x_1 + 6x_2 \quad + x_3 \le 100000 \\ x_1 \;\; + x_2 \quad\; + x_4 \le 70000 \\ 2x_1 + 2x_2 + x_5 \le 80000 \\ x_j \ge 0, (j = \overline{1,5}). \end{cases}$$

Решение

Суммарная прибыль: 32.7 млн руб.

Выработка электроэнергии

Э1: 35000 МВтч

Э2: 5000 МВтч

Продажа сырья

S1 : 0 ед.

S2 : 30000 ед.

S3 : 0 ед.

Заметим, что если продать все сырье, не вырабатывая электроэнергии, то будет получена прибыль:

100000×30+70000×40+80000×100=13.8 млн руб.

Пример 1.4

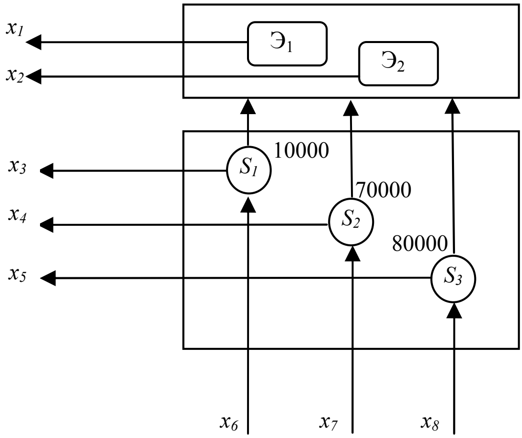
Условия примера 1.3 сохраняются, но предполагаем, что сырье можно докупать со стороны по ценам:

30 руб. – за единицу сырья вида S1;

50 руб. – за единицу сырья вида S2;

20 руб. – за единицу сырья вида S3.

При этом затраты на дополнительную покупку сырья не должны превышать 10 млн руб. Естественно, что если продажа сырья увеличивает доход АЭС, то покупка – уменьшает доход.



Математическая модель задачи:

$$700x_1 + 1400x_2 + 30x_3 + 40x_4 + 100x_5 - 30x_6 - 50x_7 - 20x_8 \to \max$$

$$2x_1 + 6x_2 \quad + x_3 - x_6 \le 100000$$

$$x_1 \;\; + x_2 \quad + x_4 - x_7 \le 70000$$

$$2x_1 + 2x_2 + x_5 - x_8 \le 80000$$

$$30x_6 \;\; + 50x_7 \quad + 20x_8 \le 10000000$$

$$x_j \ge 0, (j = 1,..,5).$$

Решение
Суммарный доход: 83.33 млн руб.
Выработка электроэнергии
П1: 6666 МВтч
П2: 63333 МВтч
Продажа сырья
S1 : 0 ед.
S2 : 0 ед.
S3 : 2 ед.
Покупка сырья
S1 : 293333 ед.
S2 : 0 ед.
S3 : 60000 ед.

Этот пример показателен с экономической точки зрения. Так, если ничего не производить, а взять, например, сырье S3 (покупка – по 20 руб. за ед.; продажа – по 100 руб.), то:

продажа всего запаса принесет доход 13.8 млн руб;

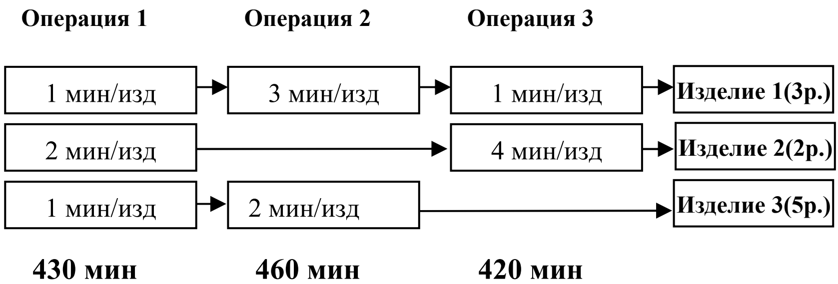
покупка на 10 млн руб. сырья S3 увеличит его запас на 500000 ед,

> а последующая продажа по цене 100 руб. за ед. принесет доход 50 млн руб. Таким образом, доход от перепродажи составит 40 млн руб. Результирующий итог такой операции равен 13.8+40=53.8 млн руб.
>
> Следовательно, в результате реализации простой спекуляционной сделки будет иметь место потеря в 83.3-53.8=29.53 млн руб.

Контрольные вопросы и задачи к разделу 1.1

1. Фирма выпускает три вида продукции (изделий). В процессе производства используются три технологические операции.

На рисунке показана технологическая схема производства.



В прямоугольниках указана длительность технологических операций при изготовлении одного изделия каждого вида.

Фонд рабочего времени, в течение которого операции могут быть применены для производства рассматриваемых изделий, ограничен:

для первой операции – 430 мин,
для второй операции – 460 мин,
для третьей операции – 420 мин.

Изучение рынка сбыта показало, что ожидаемая прибыль от продажи одного изделия видов 1, 2 и 3 составляет 3, 2 и 5 тыс. руб. соответственно.

Каков наиболее выгодный суточный объем производства каждого вида продукции?

2. Компания, специализирующаяся на разработке инновационных средств радиационной защиты, получила заказ от государства на выпуск 20000 индивидуальных защитных костюмов. На производство одного костюма расходуется не менее одного килограмма материалов. К костюмам предъявляются требования по задержке внешнего радиоактивного излучения снаружи, сохранения температурного режима внутри костюма и проницаемости костюмов для кислорода. Для производства костюмов как правило используется широкий перечень материалов. Мы же для простоты ограничимся только тремя: хлопком, инновационным наноматериалом, хорошо задерживающим радиацию и синтетической тканью. Также предположим пропорциональность характеристик материала его доле, содержащейся в костюме, и их аддитивность в случае комбинации материалов в костюме.

Данные по характеристикам каждого из трех материалов и их удельной стоимости сведем в таблицу 1.1.

Таблица 1.1

Материал	Характеристики			Стоимость руб./кг
	Задержка радиации	Проницаемость для кислорода	Сохранение тепла	
Наноматериал	100%	10%	70%	2000
Синтетика	80%	10%	85%	800
Хлопок	70%	50%	30%	400

К костюму предъявляются следующие требования:

1) сохранение не менее 40% и не более 90% тепла организма;

2) пропускание не менее 20% кислорода;

3) задержка не менее 90% радиоактивного излучения.

Необходимо определить количество (в килограммах) каждого из трех материалов, которое необходимо приобрести для выполнения госзаказа на костюмы, обладающее минимальной стоимостью при условии, чтобы характеристики готовых изделий находились в требуемых диапазонах.

3. Промышленная фирма производит изделие, представляющее собой сборку из трех узлов. Эти узлы выпускаются на двух заводах.

Из-за различия в составе технологического оборудования производительность заводов по выпуску изделий каждого вида неодинакова.

В табл.1.2 содержатся исходные данные, характеризующие как производительность заводов по выпуску каждого из узлов, так и максимальный суммарный ресурс времени, которым в течение недели располагает каждый из заводов для производства этих узлов.

Таблица 1.2

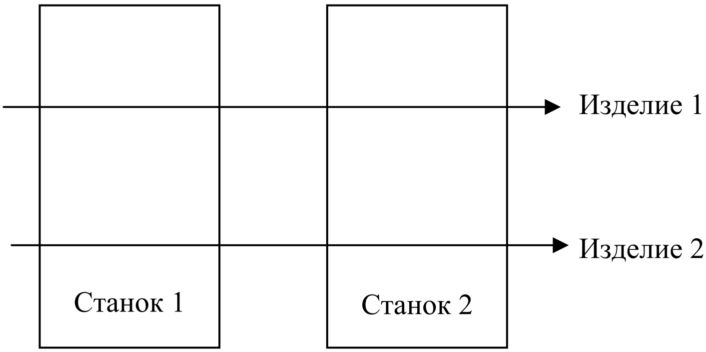
Завод	Максимальный недельный фонд времени, ч	Производительность узел/ч		
		Узел 1	Узел 2	Узел 3
1	100	8	5	10
2	80	6	12	4

Идеальной является ситуация, когда производственные мощности обоих заводов используются таким образом, что в итоге обеспечивается выпуск одинакового количества каждого из видов узлов. Однако этого трудно добиться из-за различий в производительности заводов.

Более реальная цель состоит в максимизации выпуска готовых изделий – это, по существу, эквивалентно минимизации дисбаланса, возникающего вследствие некомплектности поставки по одному или двум видам узлов.

Требуется определить еженедельные затраты времени (в часах) на производство каждого из трех узлов на каждом заводе (переменные), не превышающие в сумме временные ресурсы каждого завода (ограничения) и обеспечивающие максимальный выпуск изделий (целевая функция).

4. При изготовлении изделий двух видов осуществляется последовательная обработка соответствующих заготовок на двух различных станках.



Каждый станок может использоваться для производства изделий по 8 часов в сутки. Однако этот фонд может быть увеличен еще на 4 часа. Каждый час сверхурочных работ требует дополнительных расходов в размере 500 руб.

Производительность станков и прибыль в расчете на одно изделие приведены в таблице 1.3.

Таблица 1.3

Станок	Производительность изд./час	
	Изделие 1	Изделие 2
1	5	6
2	4	8
Прибыль (руб./ед.)	600	400

Требуется определить количество изделий каждого вида (переменные), которые нужно производить, чтобы максимизировать чистую прибыль (целевая функция), при условии, что время использования станков можно увеличить только за счет сверхурочных работ (ограничения).

1.2. Симплекс-метод

1.2.1. Виды задач линейного программирования

В линейном программировании различают три основные формы задач.

Задача линейного программирования в *<u>общей форме</u>* имеет вид:

$$\begin{gathered}\sum_{j=1}^{n} c_j x_j \to \max \\ \sum_{j=1}^{n} a_{ij} x_j \le a_i \quad (i=\overline{1,k}) \\ \sum_{j=1}^{n} a_{ij} x_j = a_i \quad (i=\overline{k+1,m}) \\ x_j \ge 0, \quad (j=\overline{1,n_1}, \quad n_1 \le n).\end{gathered} \tag{1.1}$$

Задача в *<u>стандартной (естественной, симметричной) форме</u>*:

$$\begin{gathered}\sum_{j=1}^{n} c_j x_j \to \max \\ \sum_{j=1}^{n} a_{ij} x_j \le a_i \quad (i=\overline{1,m}) \\ x_j \ge 0, \quad (j=\overline{1,n}).\end{gathered} \tag{1.2}$$

или

$$\sum_{j=1}^{n} c_j x_j \to \min$$

$$\sum_{j=1}^{n} a_{ij} x_j \ge a_i \quad (i = \overline{1,m}) \tag{1.3}$$

$$x_j \ge 0, \ (j = \overline{1,n}).$$

Задача в <u>*канонической форме*</u>:

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

$$\sum_{j=1}^{n} a_{ij} x_j = a_i \quad (i = \overline{1,m}) \tag{1.4}$$

$$x_j \ge 0, (\, j = \overline{1,n}).$$

Любой задаче линейного программирования можно придать любую из перечисленных форм, используя простейшие преобразования.

<u>1.Изменение направления оптимизации</u>

Исходная задача: Эквивалентная задача:

$$\sum_{j=1}^{n} c_j x_j \to \min \qquad\qquad -\sum_{j=1}^{n} c_j x_j \to \max$$

Оптимальные решения исходной задачи и задачи, полученной в результате преобразования 1, будут совпадать, однако оптимальные значения целевых функций будут иметь противоположные знаки.

<u>2.Придание ограничениям – нестрогим неравенствам противоположного направления</u>

Исходная задача: Эквивалентная задача:

$$\sum_{j=1}^{n} a_{ij} x_j \ge a_i \qquad\qquad -\sum_{j=1}^{n} a_{ij} x_j \le -a_j$$

<u>3.Наложение на переменные требования неотрицательности</u>

Исходная задача: Эквивалентная задача:

Замена переменной:

x_j не ограничена в знаке . $x_j = x_j^1 - x_j^2$; $x_j^1, x_j^2 \ge 0.$

4.Замена уравнений неравенствами

Исходная задача: Эквивалентная задача:

$$\sum_{i=1}^{n} a_{ij} x_j = a_i$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \le a_i \\ -\sum_{j=1}^{n} a_{ij} x_j \le -a_i \end{cases}$$

5.Замена нестрогих неравенств уравнениями

В этом преобразовании на каждое нестрогое неравенство вводится "своя" переменная, которая называется дополнительной переменной. При этом возможны два случая.

Случай 1

Исходная задача: Эквивалентная задача:

$$\sum_{j=1}^{n} a_{ij} x_j \le a_i$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j + x_{доп}^{i} = a_i \\ x_{доп}^{i} \ge 0 \end{cases}$$

Случай 2

Исходная задача: Эквивалентная задача:

$$\sum_{j=1}^{n} a_{ij} x_j \ge a_i$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j - x_{доп}^{i} = a_i \\ x_{доп}^{i} \ge 0 \end{cases}$$

Преобразование 2, по существу, является результатом трех преобразований:

Шаг 1. Изменение направления неравенства

Исходная задача: Эквивалентная задача:

$$\sum_{j=1}^{n} a_{i\,j} x_j \ge a_i \qquad -\sum_{j=1}^{n} a_{i\,j} x_j \le -a_i$$

Шаг 2. Замена неравенства уравнением

Исходная задача:

$$-\sum_{j=1}^{n} a_{ij} x_j \le -a_i$$

Эквивалентная задача:

$$\begin{cases} -\sum_{j=1}^{n} a_{ij} x_j + x_{доп}^{i} = -a_i \\ x_{доп}^{i} \ge 0 \end{cases}$$

Шаг 3. Умножение на "-1" левой и правой части уравнения.

Исходная задача:

$$\begin{cases} -\sum_{j=1}^{n} a_{ij} x_j + x_{доп}^{i} = -a_i \\ x_{доп}^{i} \ge 0 \end{cases}$$

Эквивалентная задача:

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j - x_{доп}^{i} = a_i \\ x_{доп}^{i} \ge 0 \end{cases}$$

Пример 1.5

Придать следующей задаче форму канонической:

$$\begin{array}{rcrcrcrcrcl}
z^* = & 2x_1 & + & 14x_2 & + & 18x_3 & - & 7x_4 & \to & \min \\
 & 4x_1 & + & 20x_2 & - & 4x_3 & + & 8x_4 & \le & 200 \\
 & x_1 & - & 7x_2 & + & 2x_3 & - & x_4 & \ge & 130 \\
 & 4x_1 & - & 2x_2 & + & 4x_3 & + & 5x_4 & = & 90
\end{array}$$

$x_j \ge 0,\ (j=1,2,3),\ x_4 -$ не ограничена в знаке.

Преобразование производится по следующей схеме:

1. Наложение на переменную x_4 требования неотрицательности путем замены этой переменной разностью двух неотрицательных переменных: $x_4=x_5-x_6$ (преобразование 3);
2. Изменение направления оптимизации (преобразование 1);
3. Замена нестрогих неравенств строгими равенствами (преобразование 5): в первое ограничение вводится дополнительная переменная x_7, а во второе – переменная x_8.

В результате этих преобразований задача приобретает вид:

$$\begin{array}{rcrcrcrcrcrcl}
-2x_1 & - & 14x_2 & - & 18x_3 & + & 7x_5 & - & 7x_6 & & \to & \mathbf{max} \\
4x_1 & + & 20x_2 & - & 4x_3 & + & 8x_5 & - & 8x_6 & +x_7 & = 200 \\
x_1 & - & 7x_2 & + & 2x_3 & - & x_5 & + & x_6 & -x_8 & = 130 \\
4x_1 & - & 2x_2 & + & 4x_3 & + & 5x_5 & - & 5x_6 & & = 90
\end{array}$$

$$x_j \ge 0 \qquad j=1 \div 8.$$

Оптимальное решение этой задачи:

$$x_1 = 82.2, x_2 = 0, x_3 = 0, x_5 = 0, x_6 = 47.8, x_7 = 253.3, x_8 = 0;$$

$$Z^*_{опт} = -499.$$

Оптимальное решение исходной задачи формируется следующим образом:

1. Изменяется знак оптимального значения целевой функции;
2. Отбрасываются дополнительные переменные x_7 и x_8 ;
3. Восстанавливается переменная $x_4 = x_5 - x_6$.

В итоге имеем:

$$X_{опт} = \left(82.2,\ \ 0,\ \ 0, -47.8\right); \qquad Z_{опт} = 499.$$

Задачи в форме канонической (1.4) и симметричной (1.2), (1.3) принято называть задачами с однотипными ограничениями. Для их исследования удобно использовать компактные формы записи: *матрично-векторную и векторную.*

Эти формы рассмотрим на примере канонической задачи, которую приведем в развернутом виде:

$$c_1x_1 + c_2x_2 + \ldots + c_jx_j + \ldots + c_nx_n \to \max$$

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n = a_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n = a_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n = a_i$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mj}x_j + \ldots + a_{mn}x_n = a_m$$

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

Примем следующие обозначения:

$c = (c_1, c_2, ..., c_n)$ – вектор коэффициентов целевой функции;

$x = (x_1, x_2, ..., x_n)$ – вектор переменных задачи;

$A = \left\{a_{ij}\right\} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$ – матрица, составленная из коэффициентов левой части системы ограничений;

$A_0 = (a_1, a_2, ..., a_m)^T$ – матрица-столбец, составленная из свободных членов системы ограничений;

$A_j = (a_{1j}, a_{2j}, ..., a_{mj})^T$ – матрица-столбец из коэффициентов при переменной x_j;

$X = (x_1, x_2, ..., x_n)^T$ – матрица-столбец из переменных задачи.

Тогда в матрично-векторной форме задача будет иметь вид:

$$\begin{cases} <c, x> \to \max \\ AX = A_0 \\ x \geq 0. \end{cases}$$

Здесь "$< \ >$"– скалярное произведение векторов; $x \geq 0$ означает, что все $x_j \geq 0 (j = 1, 2, ..., n)$.

В векторной форме эта задача будет записана следующим образом:

$$\begin{cases} <c, x> \to \max \\ A_1 x_1 + A_2 x_2 + ... + A_n x_n = A_0 \\ x \geq 0. \end{cases}$$

1.2.2. Свойства задач линейного программирования

Вектор $x = (x_1, x_2, ..., x_n)$, удовлетворяющий всем ограничениям задачи, называется *допустимым решением или планом задачи*.

Допустимое решение, на котором целевая функция достигает своего оптимального значения, называется *оптимальным решением или оптимальным планом задачи*.

Множество всех допустимых решений задачи линейного программирования называется *допустимым множеством*.

Рассмотрим структуру допустимого множества задачи линейного программирования на примере общей задачи:

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

$$\sum_{j=1}^{n} a_{ij} x_j \le a_i \ \ (i = \overline{1,k})$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_i \ \ (i = \overline{k+1,m})$$

$$x_j \ge 0, \ (j = \overline{1,n_1}, \ \ n_1 \le n)$$

Множество называется <u>*выпуклым*</u>, если оно вместе с любыми своими двумя точками x^1 и x^2 содержит все точки отрезка, соединяющего эти точки (рис.1.1).

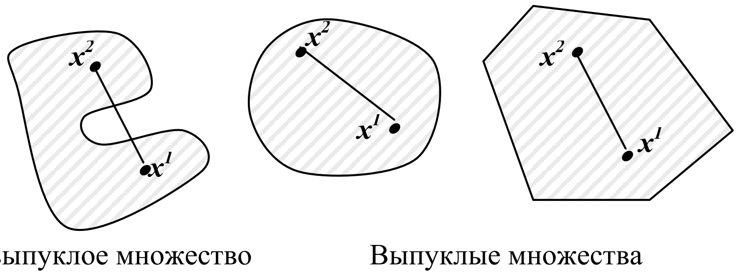


Рис.1.1. Выпуклые и невыпуклые множества

Известно, что множество точек, удовлетворяющих ограничению – нестрогому неравенству, представляет собой полупространство n-мерного евклидова пространства. Множество же точек, удовлетворяющих ограничению-уравнению – это гиперплоскость n-мерного евклидова пространства.

Следовательно, допустимое множество произвольной задачи линейного программирования – это пересечение полупространств и гиперплоскостей.

Ввиду того, что как гиперплоскости, так и полупространства являются выпуклыми множествами, их пересечение, если оно не пусто, также является выпуклым множеством. Отсюда вытекает следующее свойство задачи линейного программирования.

Допустимое множество произвольной задачи линейного программирования – это выпуклое множество. В дальнейшем будем обозначать это множество символом D.

В задачах линейного программирования градиент целевой функции не зависит от того, в какой точке он вычисляется:

$$grad(c_1x_1 + c_2x_2 + ...c_nx_n) = (c_1, c_2, ..., c_n) .$$

Следовательно, локальные экстремумы отсутствуют, и, если коэффициенты целевой функции одновременно не равняются нулю, целевая функция может достичь своего экстремального значения только на границе допустимого множества *D*. Отсюда следует второе свойство задачи ЛП.

Задача линейного программирования на максимум(минимум) целевой функции имеет решение только в том случае, если ее целевая функция на множестве *D* ограничена сверху (снизу).

Границей множества *D* называется подмножество точек, каждая из которых в любой своей ε-окрестности содержит как точки этого множества, так и точки, не принадлежащие ему.

В задаче линейного программирования граница в явном виде не задана, поэтому основной проблемой является определение этой границы – ведь согласно свойству 2 экстремум целевой функции, если он существует, достигается на границе допустимого множества *D*.

Особая роль в методах решения задач линейного программирования отводится вершинам множества *D*, которые называются также угловыми точками.

Точка $x \in D$ называется *вершиной множества* *D*, если она не может быть представлена в виде выпуклой линейной комбинации с отличными от нуля коэффициентами никаких двух других точек этого множества (рис.1.2).

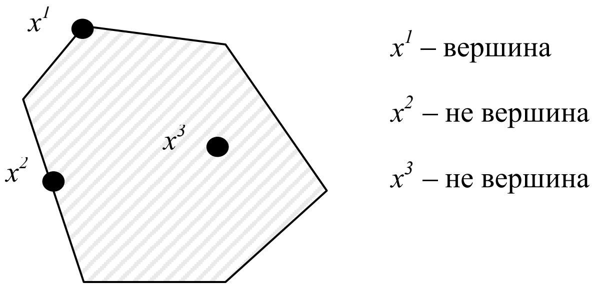


Рис.1.2. Вершины множества

Роль вершин раскрывается в следующем свойстве задачи линейного программирования.

Если задача имеет решение и ее допустимое множество D имеет хотя бы одну вершину, то и оптимального значения целевая функция достигает в вершине множества D.

1.2.3. Геометрическая интерпретация задач линейного программирования

Рассмотрим задачу линейного программирования в стандартной форме:

$$c_1x_1 + c_2x_2 \to \max,$$
$$a_{i1}x_1 + a_{i2}x_2 \le a_i,\ i = 1, 2, ..., m,$$
$$x_1, x_2 \ge 0.$$

Что представляет собой ограничение $a_{i1}x_1 + a_{i2}x_2 \le a_i$? Это ограничение определяет на плоскости одну из полуплоскостей, на которые прямая $a_{i1}x_1 + a_{i2}x_2 = a_i$ разбивает плоскость (рис.1.3).

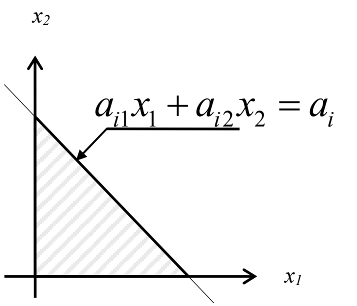


Рис.1.3. Допустимое множество

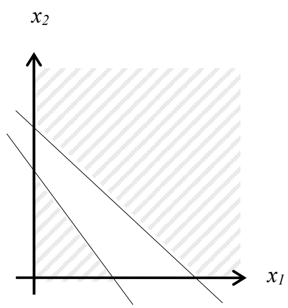


Рис.1.4. Пустое допустимое множество

Очевидно, что допустимое множество задачи – это пересечение m полуплоскостей положительного квадранта, каждая из которых является множеством точек, удовлетворяющих соответствующему ограничению – нестрогому неравенству.

Ясно, что такое пересечение может соответствовать "пустому" множеству (рис.1.4), ограниченному множеству (рис.1.5) и, наконец, неограниченному множеству (рис.1.6).

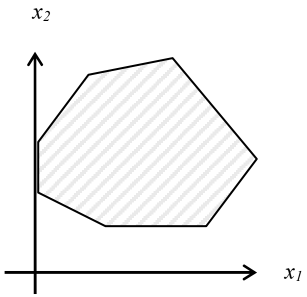


Рис.1.5. Ограниченное допустимое множество

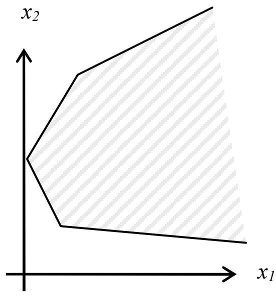


Рис.1.6. Неограниченное допустимое множество

Предположим, что множество D – ограниченное множество и представляет собой выпуклый многогранник с вершинами S_1, $S_2,...,S_7$ (рис.1.7).

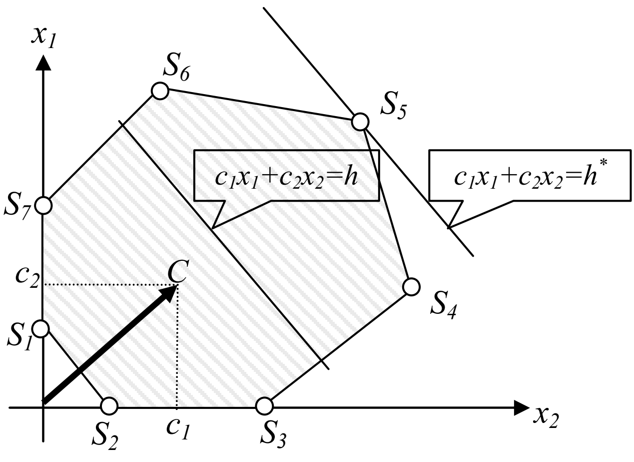


Рис.1.7. Интерпретация задачи линейного программирования

Целевая функция $c_1x_1+c_2x_2$ принимает одно и то же значение h на всех точках прямой $c_1x_1+c_2x_2 = h$, которая называется <u>*линией уровня.*</u>

Приняв h в качестве параметра, начнем увеличивать этот параметр. Очевидно, что увеличение h приведет к перемещению линии уровня в направлении нормали – вектора $c=(c_1,c_2)$**.**

При некотором значении $h = h^*$ линия уровня станет опорной к многоугольнику D. Очевидно, что h^* – максимальное значение целевой функции, которому соответствует оптимальное решение задачи линейного программирования (x_1^*, x_2^*) – вершина S_5.

В данном случае имеет место единственное решение.

Из геометрических соображений ясно, что задача линейного программирования может иметь бесконечное множество оптимальных решений (рис.1.8). Кроме того, задача может иметь оптимальное решение и при неограниченном допустимом множестве (рис.1.9).

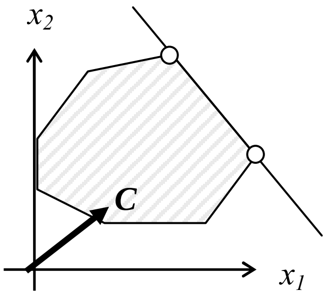


Рис.1.8. Пример задачи с бесконечным числом оптимальных решений

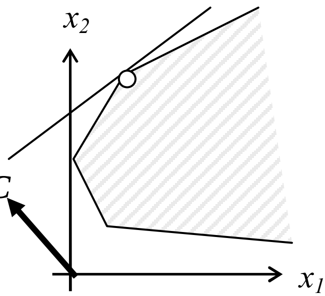


Рис.1.9. Пример оптимального решения задачи на бесконечном множестве

Во всех же случаях, когда задача линейного программирования имеет решение, по крайней мере одно из оптимальных решений соответствует вершине ее допустимого множества.

Это обстоятельство позволяет не рассматривать все точки границы допустимого множества: достаточно знать лишь координаты его вершин. Для каждой вершины можно вычислить соответствующее значение целевой функции, после чего выбрать одну вершину, координаты которой доставляют целевой функции максимальное значение.

Однако даже для задач небольшой размерности такой путь не является приемлемым, так как количество вершин может оказаться весьма большим.

Допустим, известны координаты одной какой-либо вершины. В этом случае, очевидно, не имеет смысла рассматривать вершины, в которых целевая функция принимает худшее значение. То есть, необходимо найти путь перехода к вершине с лучшим значением целевой функции. Такой переход можно осуществить по ребру допустимого множества – к смежной вершине.

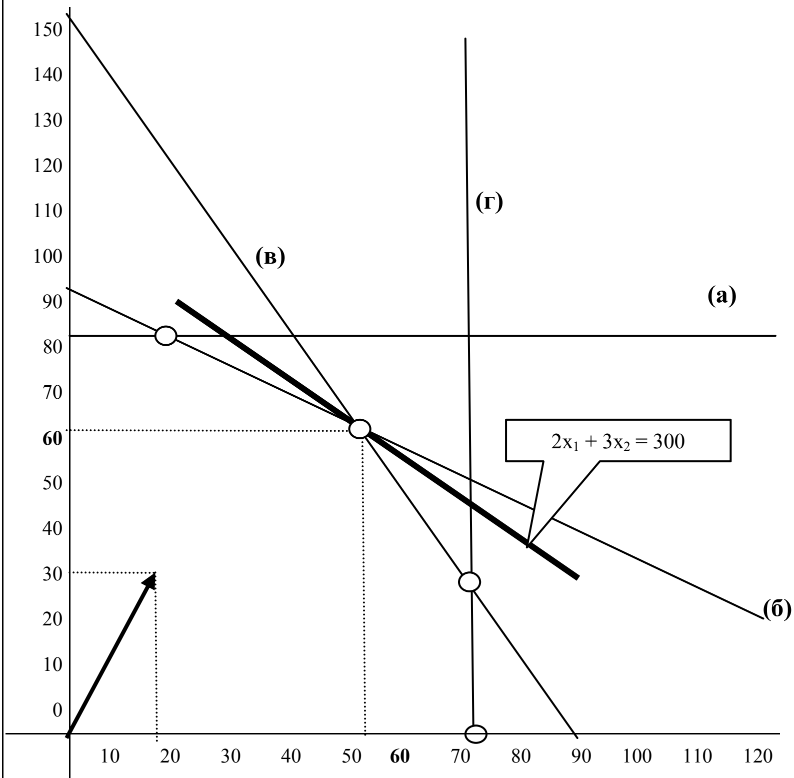
Естественно, такой подход существенно сокращает количество просматриваемых вершин допустимого множества. Однако, его следует дополнить: при переходе к очередной вершине необходимо уметь устанавливать признак того, что лучших вершин нет (текущая вершина соответствует максимальному значению целевой функции), а также признак неограниченности сверху целевой функции (признак того, что ребро, по которому следует переходить в сторону увеличения целевой функции, является неограниченным ребром).

В этом, собственно, и заключается суть симплекс-метода, самого распространенного в настоящее время метода решения задач линейного программирования.

Пример 1.6

Решить графически задачу линейного программирования:

$$
\begin{array}{lr}
2x_1 + 3x_2 \to \max & \\
x_2 \le 80 & (а) \\
x_1 + 2x_2 \le 180 & (б) \\
3x_1 + 2x_2 \le 300 & (в) \\
x_1 \qquad \le 80 & (г) \\
x_1,\ x_2 \ge 0 &
\end{array}
$$



$x_{opt}=(60,\ 60),\ z_{opt}=2*60+3*60=300$

1.2.4. Основные понятия симплекс-метода

Пусть дана каноническая задача линейного программирования в векторной форме:

$$\langle c,x\rangle \to \max \tag{1.5}$$

$$A_1x_1 + A_2x_2 + \ldots + A_nx_n = A_0 \tag{1.6}$$

$$x \geq 0 \tag{1.7}$$

Пусть ранг матрицы $A=(A_1,A_2,\ldots,A_n)$ равен m – числу ограничений-уравнений. Будем считать, что $m < n$. Заметим, что в случае $m = n$ система уравнений (1.6), если она совместна, является крамеровской и имеет единственное решение.

Задачу (1.5-1.7) будем трактовать следующим образом. Из всех возможных представлений вектора A_0 в виде линейной комбинации векторов $A_1, A_2, \ldots, A_n$ с неотрицательными коэффициентами выбрать такое, коэффициенты которого доставляют целевой функции (1.5) максимальное значение.

Учитывая роль вершин допустимого множества D, которое определяется ограничениями (1.6), (1.7), необходимо уметь определять, в каком случае допустимому решению $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ соответствует вершина допустимого множества D.

Известно, что векторы $A_1, A_2, \ldots, A_n$ составляют линейно-независимую систему, если не существует такого набора чисел $\alpha_1, \alpha_2, \ldots, \alpha_n$ одновременно не равных нулю, что:

$$\alpha_1 A_1 + \alpha_2 A_2 + \ldots + \alpha_n A_n = 0.$$

Допустимое решение $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ называется *опорным решением* (или опорным планом), если векторы A_j $(j=1,2,\ldots n)$, соответствующие положительным координатам вектора $\overline{x}$ в разложении вектора A_0 (2), составляют линейно-независимую систему. Если нулевой вектор – допустимое решение, то его также будем считать опорным решением.

Пример 1.7

Допустим, система ограничений задачи линейного программирования имеет вид:

$$\begin{aligned} 2x_1 + x_2 + 3x_3 + \quad\;\; 4x_5 &= 2, \\ x_1 \quad\;\; - 2x_3 + x_4 + 2x_5 &= 1 \\ x_j \geq 0, \ (j=1,2,\ldots,5). \end{aligned}$$

Здесь ранг системы равен двум – количеству ограничений-уравнений.

Рассмотрим следующие допустимые решения:

$\overline{x}^{(1)}$ = (*0, 2, 0, 1, 0*) – это решение опорное, так как векторы A_2 и A_4 составляют линейно-независимую систему;

$\overline{x}^{(2)}$ = (*1, 0, 0, 0, 0*) опорное;

$\overline{x}^{(3)}$ = (*1/2, 0, 0, 0, 1/4*) не опорное, так как векторы A_1 и A_5 - линейно-зависимые векторы;

$\overline{x}^{(4)}$ = (*1/2, 1, 0, 1/2, 0*) – это решение не опорное, так как имеет три положительные координаты, а ранг равен двум (векторы A_1 , A_2 , A_4 линейно-зависимы).

Опорное решение называется *__невырожденным__*, если оно содержит точно *m* положительных координат. Если количество положительных координат меньше, чем *m,* то решение называется вырожденным.

В примере 1.7 решение $\overline{x}^{(1)}$ – невырожденное опорное решение; $\overline{x}^{(2)}$ – вырожденное.

__Базисом__ опорного решения называется такой упорядоченный набор *m* линейно-независимых векторов: $A_{i_1}, A_{i_2}, ..., A_{i_m}$; $(i_k \in \{1, 2, ..., n\},\ k = \overline{1, m})$, который включает в себя все векторы, соответствующие положительным координатам этого опорного решения.

Очевидно, что базис однозначно определяется по невырожденному опорному решению. Если же решение является вырожденным, то возникает неопределенность. Так, для решения X^2 = (*1,0,0,0,0*) не ясно: в совокупности с каким вектором вектор A_1 составляет базис этого решения.

Если некоторая подсистема *m* векторов является базисом опорного решения, переменные, соответствующие векторам этой подсистемы в разложении вектора A_0 (2), называются *__базисными__*; остальные переменные – *__свободными__*.

Связь вершин допустимого множества *D* задачи линейного программирования с опорными решениями устанавливается следующей теоремой.

ТЕОРЕМА 1.1 (об опорном решении задачи линейного программирования).

Вектор $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ *тогда и только тогда является опорным решением, когда* $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ *– вершина допустимого множества D.*

Доказательство. Пусть $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ – вершина множества *D*. Предположим, что $\bar{x}$ не является опорным решением, т.е. подсистема векторов $A_{i_1}, A_{i_2}, ..., A_{i_m}$,соответствующая положительным координатам $\bar{x}$ линейно зависима. Это означает существование такого набора чисел $\alpha_1, ..., \alpha_m$, не равных нулю одновременно, что

$$\alpha_1 A_{i_1} + \alpha_2 A_{i_2} + ... + \alpha_m A_{i_m} = 0 \qquad (1.8)$$

С другой стороны, для $\bar{x}$, как для любого допустимого решения задачи, справедливо

$$\bar{x}_{i_1} A_{i_1} + \bar{x}_{i_2} A_{i_2} + ... + \bar{x}_{i_m} A_{i_m} = A_0 \qquad (1.9)$$

Сложим и вычтем (1.8) и (1.9):

$$(x_{i_1} + \alpha_1) A_{i_1} + (x_{i_2} + \alpha_2) A_{i_2} + ... + (x_{i_m} + \alpha_m) A_{i_m} = A_0$$

$$(x_{i_1} - \alpha_1) A_{i_1} + (x_{i_2} - \alpha_2) A_{i_2} + ... + (x_{i_m} - \alpha_m) A_{i_m} = A_0$$

Из этого следует, что точки $x^{(1)} = (\bar{x}_{i_1} + \alpha_1, \bar{x}_{i_2} + \alpha_2, ..., \bar{x}_{i_m} + \alpha_m)$ и $x^{(2)} = (\bar{x}_{i_1} - \alpha_1, \bar{x}_{i_2} - \alpha_2, ..., \bar{x}_{i_m} - \alpha_m)$ попадают в допустимое множество задачи (1.5)-(1.7)[1]. Но $\bar{x} = \frac{1}{2} x^{(1)} + \frac{1}{2} x^{(2)}$, т.е. не является вершиной. Получено противоречие, из которого следует неверность изначального предположения о линейной зависимости подсистемы $A_{i_1}, A_{i_2}, ..., A_{i_m}$.

---

[1] Если обнаружится, что какой-либо $x^*_{i_k} - \alpha_k < 0$, набор чисел $\alpha_1, ..., \alpha_m$ всегда можно разделить на такое положительное число *M*, что $x^*_{i_k} - \alpha_k \geq 0$, поскольку все $x^*_{i_k} > 0$. При этом условие (1.8) будет выполняться и изначальное предположение не будет нарушено.

Пусть теперь $\bar{x}$ – опорное решение задачи. Предположим от противного, что точка $\bar{x}$ не является вершиной. Пусть она представима в виде комбинации точек $x^{(1)}$ *и* $x^{(2)}$ допустимого множества:

$\bar{x} = \alpha x^{(1)} + (1-\alpha)x^{(2)}$, причем $0 < \alpha < 1$.

Координаты точек $x^{(1)}$ и $x^{(2)}$, соответствующие нулевым координатам точки $\bar{x}$, также равны нулю, поскольку уравнение

$$\alpha x_k^{(1)} + (1-\alpha)x_k^{(2)} = 0$$

при строго положительном α и неотрицательных $x_k^{(1)}$ и $x_k^{(2)}$ (они не могут быть отрицательными, в противном случае нарушалось бы требование (1.7) задачи и эти точки не попадали бы в допустимое множество) имеет решение только в случае $x_k^{(1)} = x_k^{(2)} = 0$. Рассмотрим другую комбинацию этих точек: $\bar{x} = \beta x^{(1)} + (1-\beta)x^{(2)}$, $0 < \beta < 1; \beta \neq \alpha$.

Ненулевым координатам точки $\bar{x}$ соответствуют векторы из подсистемы $A_{i_1}, A_{i_2}, ..., A_{i_m}$ (остальные векторы соответствуют только нулевым координатам, поскольку только что показано, что $x_k^{(1)} = x_k^{(2)} = 0, k = 1..n, k \notin \left\{i_1, .., i_m\right\}$). В силу того, что допустимое множество выпукло, точка $\bar{x}$ лежит в этом множестве. В результате мы получили другое разложение вектора A_0 по базису $A_{i_1}, A_{i_2}, ..., A_{i_m}$, что невозможно. Это означает неверность предположения, что точка $\bar{x}$ не является вершиной. Теорема доказана.

В соответствии с утверждением, сформулированным в этой теореме, переход от одной вершины к другой – это переход от одного опорного решения задачи линейного программирования к другому.

Известно, что *базисы опорных решений, соответствующие смежным вершинам, отличаются только одним вектором.*

В этой связи необходимо, прежде всего, ответить на вопрос: в каком случае замена одного базисного вектора другим, небазисным, приведет к новому базису?

Ответ на этот вопрос дает следующая лемма, имеющая принципиальное значение в симплекс-методе.

ЛЕММА 1.1 (переходе к новому базису).

Пусть дана система m-мерных векторов, имеющая ранг m ($m<n$):

$$A_1, A_2, \dots, A_n \tag{1.10}$$

Пусть, далее, подсистема:

$$A_1, A_2, \dots, A_{r-1}, A_r, A_{r+1}, \dots, A_m \tag{1.11}$$

составляет базис системы (1.10).

Заменим в системе (1.11) *базисный вектор* A_r *некоторым небазисным вектором* A_S.

Тогда полученная система векторов

$$A_1, A_2, \dots, A_{r-1}, A_S, A_{r+1}, \dots, A_m \tag{1.12}$$

будет являться базисом системы (1.10) *в том и только том случае, если в разложении вектора* A_S *по базису* (1.11)

$$A_S = x_{1S}A_1 + x_{2S}A_2 + \dots + x_{rS}A_r + \dots + x_{mS}A_m \tag{1.13}$$

будет иметь место:

$$x_{rS} \neq 0. \tag{1.14}$$

Доказательство. Пусть $x_{rS} = 0$. Рассмотрим линейную комбинацию системы векторов (1.12), получаемую из (1.13) переносом вектора A_S в правую часть и подстановки $x_{rS} = 0$:

$$x_{1S}A_1 + x_{2S}A_2 + \dots + 0A_r + \dots + x_{mS}A_m - A_S = 0$$

Поскольку в правой части стоит 0, система векторов (1.12) является линейно зависимой. Покажем теперь, что при $\forall x_{rs} \neq 0$ система (1.12) будет составлять базис, т.е. являться линейно независимой (поскольку она содержит m векторов). Предположим от противного, что система векторов (1.12) является линейно зависимой, т.е.

существует такой набор коэффициентов $\alpha_1, \alpha_2, ..., \alpha_m$, не равных нулю одновременно ($\sum_{i=1}^{m} \alpha_i^2 \neq 0$), что $\alpha_1 A_1 + \alpha_2 A_2 + ... + \alpha_{r-1} A_{r-1} + \alpha_r A_s + \alpha_{r+1} A_{r+1} + ... + \alpha_m A_m = 0$. Очевидно, коэффициент $\alpha_r \neq 0$, поскольку подсистема векторов $A_1, A_2, ... ,A_{r-1}, A_{r+1}, ..., A_m$, являясь частью базиса (1.11), линейно независима . Подставим в этом выражении вместо вектора A_S правую часть выражения (1.13) и сгруппируем коффициенты для кадого вектора:

$$\sum_{\substack{i=1 \\ i \neq r}}^{m} \left(\alpha_i + \alpha_r x_{is} \right) A_i + \alpha_r x_{rs} A_r = 0$$

Из полученного выражения, учитывая $\alpha_r x_{rs} \neq 0$, следует линейная зависимость системы векторов (1.11), что противоречит условию леммы (поскольку система (1.11) является базисом). Лемма доказана.

Следует заметить, что любой m-мерный вектор m-мерного евклидова пространства A_j может быть представлен в виде линейной комбинации базисных векторов:

$$A_j = x_{1j}A_1 + x_{2j}A_2 + ... + x_{rj}A_r + ... + x_{mj}A_m \qquad (1.15)$$

Причем это представление единственно.

1.2.5.Связь координат векторов в последовательных базисах

Базисы системы векторов $A_1, A_2, ... ,A_n$, отличающиеся друг от друга единственным вектором, будем называть *последовательными базисами*.

Пусть дана задача линейного программирования (1.5)-(1.7) и известно ее опорное решение

$$\overline{x} = (\overline{x_1}, \overline{x_2}, ..., \overline{x_n}). \qquad (1.16)$$

Пусть далее базисом этого опорного решения являются первые m векторов:

$$B = \{A_1, A_2, ..., A_{r-1}, A_r, A_{r+1}, ..., A_m\}. \qquad (1.17)$$

Очевидно, что значения базисных переменных решения (1.16) определяются через коэффициенты разложения вектора A_0 по базису (1.17):

$$A_0 = x_{10}A_1 + x_{20}A_2 + \ldots + x_{r0}A_r + \ldots + x_{m0}A_m. \qquad (1.18)$$

Свободные же переменные этого решения имеют нулевое значение. То есть

$$\bar{x}_k = x_{k0}\ (k=1,2,\ldots,m);\quad \bar{x}_k = 0\ (k=m+1,m+2,\ldots,n).$$

Рассмотрим разложение любого вектора A_j ($j=0,1,2,\ldots,n$) по базису (1.17):

$$A_j = x_{1j}A_1 + x_{2j}A_2 + \ldots + x_{rj}A_r + \ldots + x_{mj}A_m. \qquad (1.19)$$

Заменим в системе (1.17) базисный вектор A_r некоторым небазисным вектором A_S ($r \in \{1,2,\ldots,m\}$; $s \in \{m+1,m+2,\ldots,n\}$) , причем будем считать, что условие леммы 1.1 выполняются, т.е., $x_{rS} \neq 0$. Следовательно, новая система векторов будет соответствовать новому базису:

$$B_{нов} = \{A_1, A_2, \ldots, A_{r-1}, A_s, A_{r+1}, \ldots, A_m\}. \qquad (1.20)$$

Разложение вектора A_j в новом базисе будет выглядеть следующим образом:

$$A_j = x'_{1j}A_1 + x'_{2j}A_2 + \ldots + x'_{rj}A_s + \ldots + x'_{mj}A_m. \qquad (1.21)$$

Получим аналитическое выражение, связывающее координаты разложения любого вектора A_j в новом базисе (1.21) с координатами разложения векторов в старом базисе (1.17). Из выражения (1.18), учитывая, что $x_{rS} \neq 0$, имеем:

$$A_r = \frac{1}{x_{rs}}A_s - \frac{x_{1s}}{x_{rs}}A_1 - \frac{x_{2s}}{x_{rs}}A_2 - \ldots - \frac{x_{ms}}{x_{rs}}A_m. \qquad (1.22)$$

Подставим в (1.19) вместо вектора A_r полученное выражение. После перегруппировки членов имеем разложение вектора A_j по новому базису:

$$A_j = \left(x_{1j} - \frac{x_{rj}}{x_{rs}}x_{1s}\right)A_1 + \left(x_{2j} - \frac{x_{rj}}{x_{rs}}x_{2s}\right)A_2 + \ldots + \left(x_{r-1j} - \frac{x_{rj}}{x_{rs}}x_{r-1s}\right)A_{r-1} +$$

$$+\frac{x_{rj}}{x_{rs}}A_s + \left(x_{r+1j} - \frac{x_{rj}}{x_{rs}}x_{r+1,s}\right)A_{r+1} + \ldots + \left(x_{mj} - \frac{x_{rj}}{x_{rs}}x_{ms}\right)A_m.$$

Отсюда, учитывая (1.21), получаем основные формулы, связывающие координаты любого вектора в новом и старом базисах:

$$x'_{kj} = x_{kj} - \frac{x_{rj}}{x_{rs}} x_{ks}, (k \neq r, k = \overline{1,m}); \; x'_{rj} = \frac{x_{rj}}{x_{rs}}. \qquad (1.23)$$

Используя основные формулы, можно вычислить координаты разложения вектора A_0 в новом базисе:

$$x'_{k0} = x_{k0} - \frac{x_{r0}}{x_{rs}} x_{ks}, (k \neq r, k = \overline{1,m}); \; x'_{r0} = \frac{x_{r0}}{x_{rs}}. \qquad (1.24)$$

Если вычисленные координаты окажутся неотрицательными, будет получено новое опорное решение.

Для того чтобы можно было воспользоваться формулами (1.23), необходимо знать коэффициенты разложения не только вектора A_0, но и вектора A_S в старом базисе. Но поскольку конкретный вектор, который будет введен в базис, неизвестен, в общем случае, необходимо хранить разложения всех векторов по текущему базису. При ручном счете, а также в ряде пакетов прикладных программ для хранения соответствующей информации используются специальные таблицы – так называемые симплекс-таблицы. Фрагмент симплекс-таблицы представлен на рис.1.10.

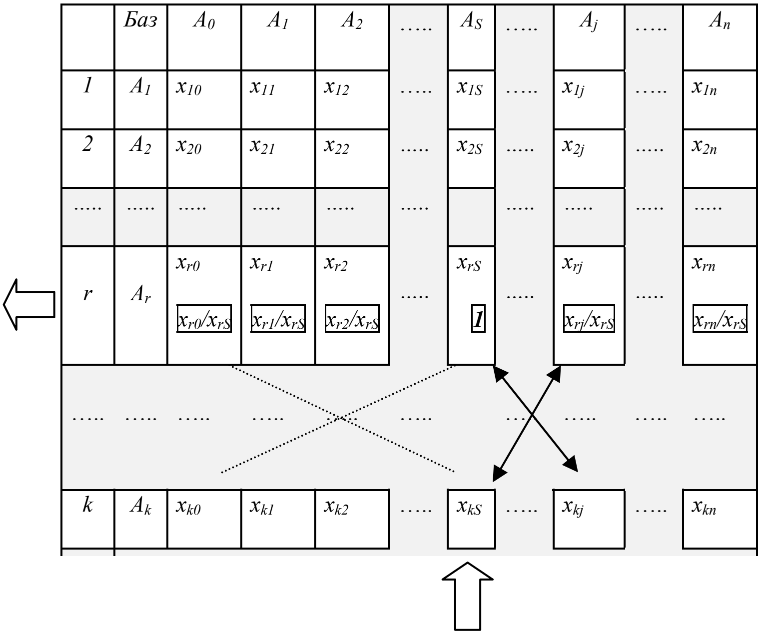


Рис.1.10 Иллюстрация пересчета симплекс-таблицы по правилу «крест-накрест»

На этом рисунке проиллюстрируем технологию пересчета координат разложения векторов в последовательных базисах. Пусть в базис вводится вектор A_S вместо базисного вектора A_r . Пересчет выполняется в следующем порядке:

1. Вычисляется r-я координата всех векторов таблицы. Для этого все элементы r-й строки делятся на x_{rS} – так называемый, <u>*ведущий элемент*</u>, причем это деление выполняется непосредственно в таблице.

2. Для вычисления всех остальных координат используется правило «крест-накрест», которое заключается в следующем.

Пусть пересчитывается координата x_{kj} **.** Мысленно строится прямоугольник, одна диагональ которого (главная диагональ) соединяет элемент x_{kj} с элементом, находящимся на пересечении r-й строки и s-го столбца (его значение – единица – это новое значение координаты x'_{rS}). Далее, из произведения элементов, соединенных главной диагональю, вычитается произведение элементов, соединенных второй диагональю.

Полученная величина $x_{kj} - \frac{x_{rj}}{x_{rs}} x_{ks}$ – это значение координаты x'_{kj} (k-я координата в разложении вектора A_j по новому базису). Результаты вычислений записываются в соответствующие клетки новой симплекс-таблицы.

Использование правила "крест-накрест" позволяет при ручном счете не запоминать основные формулы: они реализуются автоматически.

1.2.6. Переход к новому опорному решению

В основу симплекс-метода положен последовательный переход от одного опорного решения к другому, или, что эквивалентно, переход от одной вершины допустимого множества к другой – смежной вершине, в которой целевая функция имеет большее значение (точнее, не меньшее значение).

В лемме 1.1 указано условие, при выполнении которого замена в базисе одного вектора на другой оставляет систему базисных векторов линейно независимой (т.е. сохраняется базис). Для формулировки алгоритма симлекс-метода, необходимо ответить на следующие вопросы:

- При каких условиях новому базису будет соответствовать новое опорное решение?
- При каких условиях новое опорное решение будет лучшим предыдущего?
- Каков критерий оптимальности текущего опорного решения?
- Каков признак неограниченности целевой функции сверху на допустимом множестве?

Пусть дана задача линейного программирования (1.5)-(1.7):

$$<c,x> \rightarrow \max$$
$$A_1x_1 + A_2x_2 + \ldots + A_nx_n = A_0$$
$$x \geq 0,$$

причем, эта задача является <u>*невырожденной*</u>.

Задача линейного программирования называется <u>*невырожденной*</u>, если она не имеет вырожденных опорных решений.

Пусть известно опорное решение этой задачи

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n), \quad (1.25)$$

и базисом этого опорного решения являются первые m векторов[2]:

$$B = \{A_1, \ \dots \ , A_{r-1}, A_r, A_{r+1}, \ \dots, A_m\}. \qquad (1.26)$$

Значения базисных переменных решения (1.25) определяются через коэффициенты разложения вектора A_0 по базису (1.26):

$$A_0 = x_{10} A_1 + \dots + x_{r-1,0} A_{r-1} + x_{r0} A_r + x_{r+1,0} A_{r+1} + \dots + x_{m,0} A_{m}. \qquad (1.27)$$

Свободные же переменные этого решения имеют нулевое значение. То есть:

$$\overline{x_k} = x_{k0} \ (k=1,2,\dots,m); \ \overline{x_k} = 0 \ (k=m+1,m+2,\dots,n). \qquad (1.28)$$

Заменим базисный вектор A_r некоторым небазисным вектором A_S, $r \in \{1, 2, \dots, m\}$; $s \in \{m+1, m+2, \dots, n\}$, причем будем считать, что условия леммы 1.1 выполняются ($x_{rs} \neq 0$).

Новому базису

$$B_{нов} = \{A_1, \ \dots \ , A_{r-1}, A_S, A_{r+1}, \ \dots, A_m\} \qquad (1.29)$$

будет соответствовать некоторое новое решение системы уравнений (1.6)

$$x' = (x'_1, x'_2, \ \dots, x'_n), \qquad (1.30)$$

компоненты которого однозначно определяются коэффициентами разложения вектора A_0 по новому базису (1.29):

$$A_0 = x'_{10} A_1 + \dots + x'_{r-1,0} A_{r-1} + x'_{r0} A_S + x'_{r+1,0} A_{r+1} + \dots + x'_{m,0} A_m \qquad (1.31)$$

а именно:

$$\begin{cases} x'_k = x'_{k0}, \ (k = \overline{1,m}; \ k \neq r); \quad x'_r = 0; \ \ x'_s = x'_{r0}; \\ \qquad x'_l = 0, \quad (l = \overline{m+1,n}; \ \ l \neq s). \end{cases} \qquad (1.32)$$

<u>**ТЕОРЕМА 1.2**</u> (о переходе к новому опорному решению задачи линейного программирования).

Если номер вводимого в базис вектора A_S и выводимого из базиса вектора A_r таковы, что $x_{rS} > 0$ и для всех k ($k=1,2,\dots,m$; $k \neq r$), для которых $x_{kS} > 0$, имеет место

$$\frac{x_{r0}}{x_{rs}} \le \frac{x_{k0}}{x_{ks}}, \qquad (1.33)$$

то новому базису (1.29) будет соответствовать опорное решение задачи линейного программирования (1.5)-(1.7).

---

[2] Это предположение здесь и в дальнейшем делается для упрощения изложения материала. Очевидно, что все положения действительны и для другого состава базиса (в конце концов, это вопрос нумерации векторов).

Доказательство. Для вычисления коэффициентов разложения вектора A_0 по новому базису воспользуемся основными формулами.

Новое решение должно удовлетворять условиям (1.7) – только тогда оно будет опорным.

Рассмотрим выражение для вычисления r-той координаты вектора A_0. Ввиду того, что $x_{r0} > 0$ (старое решение – опорное), неравенство $x_{r0} = \dfrac{x_{r0}}{x_{rs}} > 0$ будет иметь место только в том случае, если $x_{rS} > 0$.

Рассмотрим остальные неравенства:

$$x'_{k0} = x_{k0} - \frac{x_{r0}}{x_{rs}} x_{ks} > 0, \ (k = 1, 2, ..., m; \quad k \neq r). \qquad (1.34)$$

Учитывая тот факт, что $x_{r0} > 0$, $x_{k0} > 0$ и, как только что показано, должно иметь место $x_{rS} > 0$, неравенства (1.34) будут выполняться автоматически для всех k ($k=1,2,...,m;\ k\neq r$), для которых $x_{kS} < 0$.

Для всех же $x_{kS} > 0$ ($k=1,2,...,m;\ k\neq r$) неравенства (1.34) будут выполняться только в том случае, если имеет место (1.33). Теорема доказана.

1.2.7. Переход к лучшему опорному решению и критерий оптимальности

Пусть дана задача линейного программирования (1.5)-(1.7); известно ее опорное решение (1.25) и базис этого решения (1.26).

Пусть далее базисный вектор A_r заменяется свободным вектором A_S и при этом выполняются условия теоремы 1.2 о переходе к новому опорному решению.

Величина $\Delta_S = (c_1x_{1S}+c_2x_{2S}+...+c_mx_{mS} - c_S)$ называется *оценкой вектора A_S.*

ТЕОРЕМА 1.3 (о возможности улучшения опорного решения задачи линейного программирования).

Если для данного опорного решения существует такая отрицательная оценка ($\Delta_S<0$), что среди координат разложения вектора A_S по данному базису (т.е. среди чисел x_{kS}, $k=1,2,...,m$) есть положительные, то базис, которому будет соответствовать лучшее опорное решение, будет получен, если вектором A_S заменить тот вектор A_r, для которого

$$x_{rS}>0 \quad и \quad \frac{x_{r0}}{x_{rs}} \le \frac{x_{k0}}{x_{ks}} \quad для\ всех\ x_{kS}>0,\quad k=1,2,...,m.$$

Доказательство. Новому опорному решению (1.30), учитывая (1.31) и (1.32), будет соответствовать новое значение целевой функции:

$$Z_{нов}=c_1x'_{10}+...+c_{r1}x'_{r-1,0}+c_Sx'_{r0}+c_{r+1}x'_{r+1,0}+...+c_mx'_{m0}. \quad (1.35)$$

Используя основные формулы, выразим новые координаты разложения вектора A_0 через координаты разложения векторов по старому базису.

После простых преобразований (добавления нулевой суммы c_rx_{r0} - c_rx_{r0} и перегруппировки членов) будет получено следующее выражение:

$$Z_{нов}=c_1x_{10}+c_2x_{20}+...+c_mx_{m0} - \frac{x_{r0}}{x_{rs}}(c_1x_{1S}+c_2x_{2S}+...+c_mx_{mS} - c_S). \quad (1.36)$$

Очевидно, $Z_{ст}=c_1x_{10}+c_2x_{20}+...+c_mx_{m0}$ – это значение, которое имеет целевая функция на старом опорном решении. В скобках же написана в точности величина оценки Δ_S вектора A_S. Совершив изложенные подстановки, получим новую запись выражения (1.36):

$$Z_{нов} = Z_{ст} - \frac{x_{r0}}{x_{rs}}\Delta_S. \quad (1.38)$$

Рассмотрим выражение (1.38). Ввиду того, что $x_{rS} > 0$ и $x_{r0} > 0$, новое значение целевой функции будет больше старого значения в том и только том случае, если вектор A_S, вводимый в базис, будет иметь отрицательную оценку ($\Delta_S<0$). Теорема доказана.

Следствие. *Если для некоторого опорного решения оценки всех свободных векторов неотрицательны, данное опорное решение оптимальным.*

Следует отметить, что оценки базисных векторов всегда нулевые. Действительно, если вектор A_j – базисный, то его разложение по базису тривиально:

$$A_j = x_{1j}A_1 + x_{2j}A_2 +...+ x_{jj}A_j+...+ x_{mj}A_m.$$

Здесь все коэффициенты, кроме x_{jj}, имеют нулевое значение, а $x_{jj}=1$. Следовательно, в соответствии с (1.39):

$$\Delta_j = c_j - c_j = 0.$$

Подводя итог, можно заметить, что если для вводимого в базис вектора $\Delta_S < 0$, то очередному решению будет соответствовать большее значение целевой функции; если $\Delta_S > 0$ – меньшее; если же $\Delta_S = 0$, значение целевой функции не изменится.

В старом базисе оценка вектора A_j ($j=1,2,...,n$) определяется выражением (1.39):

$$\Delta_j = c_1 x_{1j} + c_2 x_{2j} + ... + c_m x_{mj} - c_j.$$

В новом базисе этот же вектор будет иметь оценку Δ'_j :

$$\Delta'_j = c_1 x'_{1j} + ... + c_{r-1} x'_{r-1,j} + c_S x'_{rj} + c_{r+1,j} x'_{r+1,j} + ... + c_m x'_{mj} - c_j.$$

Используя основные формулы, выразим новые координаты через старые. Путем простых преобразований будет получено следующее выражение:

$$\Delta_j' = \Delta_j - \frac{x_{rj}}{x_{rs}} \Delta_s. \tag{1.40}$$

1.2.8. Признак неограниченности сверху целевой функции

Рассмотрим последний случай, возможный при решении задачи линейного программирования симплекс-методом. Этот случай оформим в виде следующей теоремы.

ТЕОРЕМА 1.4 (о неограниченности сверху целевой функции задачи линейного программирования).

Пусть для некоторого опорного решения задачи

$$\langle c,x\rangle \rightarrow \max,$$
$$A_1x_1 + A_2x_2 + ... + A_nx_n = A_0,$$
$$x \geq 0,$$

существует свободный вектор A_s, имеющий отрицательную оценку $\Delta_s < 0$, для которого выполняется

$$x_{ks} \leq 0, \ (k = 1,2,...,m). \tag{1.41}$$

Тогда целевая функция задачи не ограничена сверху на допустимом множестве.

Доказательство. Пусть

$$\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_m, 0, 0, ..., 0) \tag{1.42}$$

опорное решение, для которого справедливы условия теоремы (1.41).

Рассмотрим вектор $x^* = (x_1^*, x_2^*, ..., x_n^*)$, координаты которого определим следующим образом:

$$x_k^* = \overline{x}_k - \xi\ x_{ks}, (k=1,2,...,m);$$
$$x_s^* = \xi \tag{1.43}$$
$$x_l^* = 0,\ (l = m+1, m+2, ..., n\ ,\ l \neq s).$$

Здесь ξ – некоторое положительное число.

Относительно вектора x^* можно утверждать следующее:

- все координаты этого вектора неотрицательные, так как по условию теоремы $x_{ks} \leq 0,$ $(k = 1,2,...,m)$ и решение (1.42) – опорное, не имеющее отрицательных координат;
- вектор $x^* = (x_1^*, x_2^*, ..., x_n^*)$ является допустимым решением задачи.

Действительно, подставим его в систему ограничений задачи:

$$\left(\overline{x}_1 - \xi\ x_{1s}\right)A_1 + ... + \left(\overline{x}_m - \xi\ x_{ms}\right)A_m + \xi\ A_s =$$
$$= \overline{x}_1 A_1 + \overline{x}_2 A_2 + ... + \overline{x}_m A_m - \xi\ \left(A_1 x_{1s} + A_2 x_{2s} + ... + A_m x_{ms}\right) +$$
$$+ \xi\ A_s = A_0 - \xi\ A_s + \xi\ A_s = A_0.$$

Вектору x^* соответствует следующее значение целевой функции:

$$Z^* = c_1\left(\overline{x}_1 - \xi\ x_{1s}\right) + ... + c_m\left(\overline{x}_m - \xi\ x_{ms}\right) + c_s \xi =$$
$$= c_1 \overline{x}_1 + c_2 \overline{x}_2 + ... + c_m \overline{x}_m - \xi\ \left(c_1 x_{1s} + c_2 x_{2s} + ... + c_m x_{ms} - c_s\right) =$$
$$= Z - \xi\ \Delta_s,$$

где Z – значение целевой функции на решении (1.43).

По условию теоремы $\Delta_s < 0$. Следовательно, взяв достаточно большое ξ, можно сделать Z^* сколь угодно большим. Таким образом, доказано, что целевая функция задачи линейного программирования не ограничена сверху.

1.2.9. Определение коэффициентов разложения векторов по базису

Пусть дана задача линейного программирования в канонической форме:

$$<c,x> \to \max,$$
$$A_1x_1 + A_2x_2 + \ldots + A_nx_n = A_0\,,$$
$$x \geq 0.$$

Предположим, что ранг матрицы $A=(A_1,A_2,...,A_n)$ равен m – количеству ограничений-уравнений.

Пусть известно некоторое опорное решение $x^* = (x_1^*, x_2^*, ..., x_n^*)$ и базис этого решения:

$$B = (A_{i_1}, A_{i_2}, ..., A_{i_m}). \tag{1.44}$$

Для того чтобы на практике использовать результаты, сформулированные в теоремах 1.2, 1.3 и 1.4, необходимо знать коэффициенты разложения векторов A_0, A_1, …, A_n по базису (1.44).

Разложение любого базисного вектора тривиально. Так, если вектор A_{i_k} – базисный, его разложение по базису будет иметь вид:

$$A_{i_k} = 0A_{i_1} + 0A_{i_2} + ... + 1A_{i_k} + ... + 0A_{i_m}\,.$$

В этом выражении все коэффициенты, кроме x_{k,i_k}, имеют нулевое значение, а $x_{k,i_k} = 1$.

Рассмотрим разложение свободного вектора A_j $(j=0,1,2,...,n)$ по базису:

$$A_j = x_{1j}A_{i_1} + x_{2j}A_{i_2} + ... + x_{mj}A_{i_m}\,. \tag{1.45}$$

Перепишем это выражение в виде матричного уравнения:

$$A_j = BX_j\,, \tag{1.46}$$

где

$$X_j = (x_{1j}, x_{2j}, ..., x_{mj})^T - \tag{1.47}$$

матрица-столбец, составленная из коэффициентов разложения вектора A_j по базису (1.44).

Решение матричного уравнения (1.46) находится путем умножения слева обеих частей равенства на матрицу, обратную к матрице B,

$$B^{-1}A_j = B^{-1}BX_j = EX_j = X_j, \tag{1.48}$$

где B^{-1} – обратная матрица, а E – единичная матрица:

$$E = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 1 \end{pmatrix}.$$

Таким образом, для нахождения коэффициентов разложения любого вектора A_j $(j=0,1,2,...,n)$ по базису (1.44) необходимо знать обратную матрицу B^{-1}.

Вычисление B^{-1} – весьма трудоемкий процесс. Вместе с тем, известно, что если B является единичной матрицей, то и B^{-1} – единичная матрица, т.е.

$$B = B^{-1} = E.$$

Следовательно, учитывая (1.48):

$$X_j = B^{-1}A_j = EA_j = A_j.$$

Таким образом, *вектор коэффициентов разложения любого вектора* A_j $(j=0,1,2,...n)$ *по единичному базису совпадает с самим этим вектором.* В симплекс-методе это обстоятельство используется следующим образом.

Допустим, в исходной системе ограничений задачи линейного программирования имеется полный набор единичных векторов: $A_{i_1}, A_{i_2}, ..., A_{i_m}$; $i_k \in \{1, 2, ..., n\}$, $k = 1, 2, ..., m$,

т.е., система ограничений имеет вид

$$\begin{cases} a_{11}x_1 + ... + 1x_{i_1} + ... + 0x_{i_2} + ... + 0x_{i_k} + ... 0x_{i_m} + ... + a_{1n}x_n = a_1 \\ a_{21}x_1 + ... + 0x_{i_1} + ... + 1x_{i_2} + ... + 0x_{i_k} + ... 0x_{i_m} + ... + a_{2n}x_n = a_2 \\ .... \\ a_{k1}x_1 + ... + 0x_{i_1} + ... + 0x_{i_2} + ... + 1x_{i_k} + ... 0x_{i_m} + ... + a_{kn}x_n = a_k \\ .... \\ a_{m1}x_1 + ... + 0x_{i_1} + ... + 0x_{i_2} + ... + 0x_{i_k} + ... 1x_{i_m} + ... + a_{mn}x_n = a_m \\ x_j \geq 0, j = 1, 2, ..., n. \end{cases}$$

Эта форма задачи линейного программирования называется <u>*приведенной*</u>.

В случае, если все $a_i \geq 0$ $\left(i = 1,2,...,m\right)$, векторы $(A_{i_1}, A_{i_2}, ..., A_{i_m})$ можно принять в качестве базиса опорного решения $\overline{x} = (\overline{x_1}, \overline{x_2}, ..., \overline{x_n})$, значения координат которого определяются следующим образом:

$$\overline{x}_{i_k} = x_{k0}, k \in \{1,2,...,m\}; \overline{x}_j = 0, j \notin \{i_1, i_2, ..., i_m\}.$$

При этом известны коэффициенты разложения всех векторов по базису: $x_{ij} = a_{ij} \left(i = 1,2,...,m; j = 1,2,...,n\right)$. Следовательно, можно сразу заполнить соответствующий фрагмент симплекс-таблицы.

Пример 1.8

Дана задача линейного программирования:

$$2x_1 - 8x_2 - 2x_3 + 6x_4 + 3x_5 \to \max,$$

$$\frac{1}{4}x_1 + x_2 - \frac{1}{4}x_3 + x_4 = \frac{1}{2}$$

$$-\frac{1}{4}x_1 - 4x_2 + \frac{3}{4}x_3 + x_5 = 1$$

$$x_j \geq 0, \left(j = 1,2,...,5\right).$$

Здесь векторы A_4 и A_5 составляют единичный базис, которому соответствует опорное решение. $X^* = (0,0,0,1/2,1)$.

Ниже приводится фрагмент симплекс-таблицы для этой задачи.

Баз	A_0	A_1	A_2	A_3	A_4	A_5
A_4	*1/2*	*1/4*	*1*	*-1/4*	*1*	*0*
A_5	*1*	*-1/4*	*-4*	*3/4*	*0*	*1*

Если в системе ограничений задачи отсутствует полный набор единичных векторов, вопрос нахождения исходного опорного решения и коэффициентов разложения векторов по базису этого решения несколько осложняется.

В частном случае, когда задача имеет симметричную форму и неотрицательные свободные члены, поступают следующим образом.

Пусть задача имеет вид

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n \to \max$$

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le a_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le a_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le a_m$$

$$x_j \ge 0, \left(j = 1,2,\ldots,n\right);\; a_i \ge 0,\; \left(i = 1,2,\ldots,m\right).$$

Путем введения дополнительных переменных: $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ – по одной на каждое ограничение, задаче придается каноническая форма:

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n \to \max$$

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} = a_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + x_{n+2} = a_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+m} = a_m$$

$$x_j \ge 0,\; (j = 1,2,\ldots,n+m).$$

Теперь векторы $A_{n+1}, A_{n+2}, \ldots, A_{n+m}$ составляют полный единичный базис, которому соответствует опорное решение $\bar{x}=$ $=(0, 0, \ldots, 0, a_1, a_2, \ldots, a_m)$. В этом решении

$$\bar{x}_j = 0 \;\left(j = 1,2,\ldots,n\right) \quad \text{и} \quad \bar{x}_{n+k} = a_k \;\left(k = 1,2,\ldots,m\right).$$

Известны также коэффициенты разложения всех векторов по базису:

$$x_{kj}=a_{kj},\; x_{k0}=a_k,\; (k=1,2,\ldots,m;\; j=1,2,\ldots,n).$$

В общем же случае, когда задача имеет каноническую форму, и в системе ограничений отсутствует полный набор единичных векторов, для нахождения исходного решения, которому соответствует единичный базис, используется специальный метод – *метод искусственного базиса*, основанный на использовании вычислительной процедуры самого же симплекс-метода. Две разновидности этого метода рассмотрены в разделах 1.2.13 и 1.2.14.

1.2.10. Алгоритм симплекс-метода для невырожденной задачи

Пусть дана задача ЛП в канонической форме:

$$<c,x> \to max,$$
$$A_1+A_2+\ldots+A_n=A_0,$$
$$x \geq 0.$$

Пусть известно некоторое опорное решение

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$$

и базис этого решения:

$$B=(A_{i_1}, A_{i_2}, \ldots, A_{i_m}).$$

Будем считать, что базисная матрица является единичной матрицей ($B=E$), т.е. система ограничений задачи имеет форму приведенной системы, а вектор A_0 не имеет отрицательных координат: $a_i \geq 0$, ($i=1,2,\ldots,m$).

Очевидно, что ранг базисной матрицы B равен m – количеству ограничений-уравнений (m единичных векторов $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ составляют линейно независимую систему).

Рассмотрим основные шаги алгоритма.

<u>Шаг 1</u>. Разложить векторы A_0, A_1, A_2, ... A_n по базису, т.е. найти все числа x_{kj} ($j=0,1,2,\ldots,n;\ k=1,2,\ldots,m$) такие, что:

$$A_j=x_{1j} A_{i_1} + x_{2j} A_{i_2} +\ldots+ x_{mj} A_{i_m}.$$

Учитывая тот факт, что $B=(A_{i_1}, A_{i_2}, \ldots, A_{i_m})=E=B^{-1}$, коэффициенты разложения определяются непосредственно параметрами задачи:

$$x_{kj}=a_{kj},\quad x_{k0}=a_k\ (j=1,2,\ldots,n;\ k=1,2,\ldots,m).$$

При этом, координаты опорного решения $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ определяются следующим образом:

$$\bar{x}_{i_k} = a_k = x_{k0}, (k=1,2,\ldots,m);\ \bar{x}_j = 0, (j=1,2,\ldots,n;\ j \notin \{i_1, i_2, \ldots, i_m\}).$$

<u>Шаг 2</u>. Найти оценки всех свободных векторов A_j, $j \notin \{i_1, i_2, \ldots, i_m\}$:

$$\Delta_j = c_{i_1} x_{1j} + c_{i_2} x_{2j} + \ldots + c_{i_m} x_{mj} - c_j.$$

$$Z_0 = c_{i_1} x_{10} + c_{i_2} x_{20} + \ldots + c_{i_m} x_{m0}.$$

Результаты вычислений занести в симплекс-таблицу.

Ниже представлена полная симплекс-таблица.

Баз	$C_{баз}$	A_0	c_1 / A_1	c_2 / A_2		c_s / A_s		c_j / A_j		c_n / A_n
A_{i_1}	c_{i_1}	x_{10}	x_{11}	x_{12}		x_{1s}		x_{1j}		x_{1n}
A_{i_2}	c_{i_2}	x_{20}	x_{21}	x_{22}		x_{2s}		x_{2j}		x_{2n}
A_{i_r}	c_{i_r}	x_{r0}	x_{r1}	x_{r2}		x_{rs}		x_{rj}		x_{rn}
A_{i_k}	c_{i_k}	x_{k0}	x_{k1}	x_{k2}		x_{ks}		x_{kj}		x_{kn}
A_{i_m}	c_{i_m}	x_{m0}	x_{m1}	x_{m2}		x_{ms}		x_{mj}		x_{mn}
		Z_0	Δ_1	Δ_2		Δ_s		Δ_j		Δ_n

Шаг 3. Если все оценки $\Delta_j \geq 0$ $(j=1,2,...,n)$, процесс закончен. Очередное опорное решение – оптимальное решение задачи. Координаты этого решения $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ однозначно определяются коэффициентами разложения вектора A_0 :

$$\overline{x}_{i_k} = a_k = x_{k0}, (k=1,2,...,m); \ \overline{x}_j = 0, (j=1,2,...,n; j \notin \{i_1, i_2, ..., i_m\}).$$

Шаг 4. Последовательно просматриваются все свободные векторы, имеющие отрицательные оценки. Если обнаруживается такой вектор A_j ($\Delta_j < 0$), для которого все $x_{kj} \leq 0$, $(j=1,2,...,n)$, вычисления прекращаются: задача не имеет решения, так как ее целевая функция не ограничена сверху на допустимом множестве. В противном случае выполняется следующий шаг.

Шаг 5. Выбирается любой вектор, имеющий отрицательную оценку (обычно предпочтение отдается вектору с максимальной по абсолютной величине отрицательной оценкой – это, как правило, сокращает общее количество вычислительных операций). Пусть, для определенности, выбран вектор A_s $(\Delta_s < 0)$. Этот вектор будет вводиться в базис.

Шаг 6. Определяется вектор, который будет выводиться из базиса. Для этого просматриваются все элементы s-го столбца симплекс-таблицы и для всех k $(k=1,2,...,m)$, для которых имеет место $x_{ks}>0$, вычисляется отношение x_{k0}/x_{ks}.

Из полученных отношений выбирается минимальное. Пусть

$$\frac{x_{r0}}{x_{rs}} = \min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\}.$$

Тогда из базиса будет выводиться вектор A_{i_r}. Выполняется следующий шаг.

Шаг 7. Для нового базиса

$$B_{нов}=(A_{i_1}, A_{i_2}, ..., A_{i_{r-1}}, A_s, A_{i_{r-1}}, ..., A_{i_m}),$$

с использованием основных формул пересчитываются координаты разложения всех векторов A_j ($j=0,1,2,...,n$):

$$x'_{kj} = x_{kj} - \frac{x_{rj}}{x_{rs}} x_{ks}\ ,\ (k \neq r, k=\overline{1,m});\ x'_{rj} = \frac{x_{rj}}{x_{rs}}.$$

Далее, вычисляется значение целевой функции на новом опорном решении:

$$Z_{нов} = Z_{ст} - \frac{x_{r0}}{x_{rs}} \Delta_s.$$

В этом выражении $Z_{нов}$ и $Z_{ст}$ – соответственно, новое и старое значения целевой функции. Наконец, вычисляются оценки свободных векторов в новом базисе. Здесь используется выражение:

$$\Delta_j' = \Delta_j - \frac{x_{rj}}{x_{rs}} \Delta_s .$$

При ручном счете для вычислений по приведенным выше формулам используется правило “крест-накрест”. Результаты вычислений заносятся в новую симплекс-таблицу и выполняется шаг 3.

Пример 1.9

Решить симплекс-методом следующую задачу линейного программирования.

$$5x_1 + 4x_2 - 4x_3 - 9x_4 \rightarrow max$$

$$\frac{2}{3}x_1 + \frac{1}{3}x_2 + x_3 = \frac{8}{3}$$

$$\frac{1}{3}x_1 + \frac{2}{3}x_2 + x_4 = \frac{10}{3}$$

$$x_j \geq 0,\ j=1 \div 4$$

Система ограничений этой задачи имеет форму приведенной системы: векторы A_3 и A_4 оставляют полный единичный базис, которому соответствует опорное решение $x^*=(0,0,8/3,10/3)$.

Заполним первую симплекс-таблицу коэффициентами разложения всех векторов по базису $B=(A_3 , A_4)=E$. Эти коэффициенты запишем в левые верхние углы соответствующих клеток таблицы.

			5	4	-4	-9
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A 3	*-4*	*8/3*	*2/3*	*1/3*	*1*	*0*
A4	*-9*	*10/3* $\boxed{5}$	*1/3* $\boxed{1/2}$	*2/3* $\boxed{1}$	*0* $\boxed{0}$	*1* $\boxed{3/2}$
Табл.1		*-122/ 3*	*-32/ 3*	*-34/ 3*	*0*	*0*

Вычислим значение целевой функции. Для этого попарно перемножим элементы столбцов $C_{баз}$ и A_0 и сложим полученные произведения:

$$Z_0= -4\times(8/3) -9\times(10/3)=-122/3.$$

Фактически, получено скалярное произведение двух векторов:

$$C_{баз}=(c_3,c_4)=(-4,-9) \text{ и } X_0=(x_{10},x_{20})=(8/3,10/3).$$

Для вычисления оценок Δ_j ($j=1,2,...,n$) используется тот же прием: вычисляется скалярное произведение векторов $C_{баз}=(c_3,c_4)$ и $X_j=(x_{1j}, x_{2j})$, а затем из полученного произведения вычитается c_j .

Так, например, оценки векторов A_1 и A_2 подсчитываются следующим образом:

$$\Delta_1 =-4\times(2/3) - 9\times(1/3) - 5 =-32/3$$

$$\Delta_2 =-4\times(1/3) - 9\times(2/3) - 4 =-34/3.$$

Наличие отрицательных оценок свидетельствует о том, что рассматриваемое опорное решение не является оптимальным.

В каждом столбце с отрицательной оценкой имеются положительные элементы, поэтому нет основания для принятия решения о неограниченности целевой функции сверху: решение можно улучшить путем введения в базис вектора с отрицательной оценкой.

Из двух таких векторов A_1 и A_2 выбираем вектор A_2, обладающий наибольшей по абсолютной величине отрицательной оценкой.

Следует отметить, что выбор вектора A_1 не приведет к ошибке: и в том, и в другом случае будет получено лучшее опорное решение.

В столбце A_2 есть два положительных элемента: x_{12} и x_{22}. Следовательно, для того чтобы принять решение, какой вектор выводить из базиса (A_3 или A_4), необходимо вычислить отношения:

$$\frac{x_{10}}{x_{12}} = \frac{8/3}{1/3} = 8\,;\ \frac{x_{20}}{x_{22}} = \frac{10/3}{2/3} = 5.$$

Минимальное отношение соответствует вектору A_4 — его нужно вывести из базиса.

Начинаем пересчет координат разложения векторов по новому базису, а также нового значения целевой функции и оценок. Пересчет проводится в следующем порядке:

- Вычисляются вторые координаты всех векторов. Для этого вторая строка таблицы делится на ведущий элемент (в нашем случае это элемент $x_{22} = 2/3$).
- Результат деления записывается непосредственно в эту же строку (в правый нижний угол соответствующих клеток таблицы).
- Эти же данные записываются во вторую строку новой симплекс-таблицы. Для вычисления всех остальных координат, нового значения целевой функции и оценок используется правило "крест-накрест". Результаты записываются в новую таблицу.

			5	4	-4	-9
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_3	-4	1 $\boxed{2}$	1/2 $\boxed{1}$	0 $\boxed{0}$	1 $\boxed{2}$	-1/2 $\boxed{-1}$
A_2	4	5	1/2	1	0	3/2
Табл.2		16	-5	0	0	17

Начинаем анализ нового опорного решения $x^*=(0,5,1,0)$.

В базис вводится вектор A_1, имеющий отрицательную оценку, а выводится вектор A_3, так как этому вектору соответствует минимальное отношение:

$$\frac{x_{10}}{x_{11}} = \frac{1}{1/2} = 2 < \frac{x_{20}}{x_{21}} = \frac{5}{1/2} = 10.$$

Третья симплекс-таблица.

			5	4	-4	-9
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_1	5	2	1	0	2	-1
A_2	4	4	0	1	-1	2
Табл.3		26	0	0	10	12

Все оценки неотрицательны, следовательно, получено оптимальное решение:

$x^*=(2,4,0,0)$; $Z_{опт.}= 26$.

Разберем еще один пример решения задачи симплекс-методом.

<u>Пример 1.10</u>

Решить симплекс-методом следующую задачу линейного программирования:

$$
\begin{array}{rrrrrcl}
2x_1 & +x_2 & -5x_3 & -x_4 & -x_5 & \to & max \\
x_1 & -x_2 & +x_3 & +x_4 & & = & 5 \\
-x_1 & -x_2 & +x_3 & & +x_5 & = & 1
\end{array}
$$

$x_j \geq 0, j=1 \div 5.$

Решение этой задачи представлено ниже двумя симплекс-таблицами. На второй итерации срабатывает признак неограниченности целевой функции: вектор A_2 имеет отрицательную оценку, а среди координат разложения этого вектора по базису (A_1, A_5) нет ни одной положительной координаты.

Целевая функция задачи не ограничена сверху на допустимом множестве.

			2	1	-5	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5
A_4	-1	5	1	-1	1	1	0
A_5	-1	1	-1	-1	1	0	1
Табл.1		-6	-2	1	3	0	0

			2	1	-5	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5
A_1	2	5	1	**-1**	1	1	0
A_5	-1	6	0	**-2**	2	1	1
Табл.2		4	0	**-1**	5	2	0

1.2.11. Симплекс-метод в общем случае

До сих пор рассматривался случай невырожденной задачи линейного программирования. Рассмотрим общий случай.

Пусть дана задача, в которой возможны вырожденные опорные решения:

$$< c, x > \to \max,$$

$$A_1x_1 + A_2x_2 + \ldots + A_nx_n = A_0,$$

$$x \geq 0.$$

причем, ранг матрицы $A = (A_1, A_2, \ldots, A_n)$ равен, по-прежнему, m – количеству ограничений-уравнений.

Предположим сначала, что имеется некоторое опорное решение $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ и базис этого решения:

$$B = (A_{i_1}, A_{i_2}, ..., A_{i_m}),\ i_k \in \{1,2,...,n\}.$$

Пусть далее не выполняется ни условие оптимальности этого решения, ни условие неограниченности целевой функции сверху на допустимом множестве.

Возьмем любой вектор A_S, имеющий отрицательную оценку:

$$\Delta_s < 0,\ (s \in \{1,2,...,n\};\ s \notin \{i_1, i_2, ..., i_m\}).$$

Предположим, что $\theta_0 = \min\limits_{x_{ks}>0} \left\{ \dfrac{x_{k0}}{x_{ks}} \right\}$.

Для невырожденной задачи линейного программирования существует единственный базисный вектор, которому соответствует θ_0.

Замена в базисе именно этого вектора небазисным вектором A_S гарантирует, что новому базису будет соответствовать опорное решение, и оно будет невырожденным. Если же таких векторов несколько, новое опорное решение будет вырожденным.

Действительно, пусть

$$\theta_0 = \min_{x_{ks}>0} \left\{ \frac{x_{k0}}{x_{ks}} \right\} = \frac{x_{r0}}{x_{rs}} = \frac{x_{l0}}{x_{ls}},\quad (r,l \in \{1,2,...,m\}, r \neq l).$$

Заменим базисный вектор A_{i_r} вектором A_S ($\Delta_s < 0$).

Используя основные формулы, вычислим значение l-й координаты в разложении вектора A_0 по новому базису:

$$x'_{l0} = x_{l0} - \frac{x_{r0}}{x_{rs}} x_{ls} = x_{l0} - \frac{x_{l0}}{x_{ls}} x_{ls} = 0\ .$$

Таким образом, в новом опорном решении появилась базисная переменная, имеющая нулевое значение.

Пусть теперь имеется некоторое вырожденное опорное решение $\overline{x} = (\overline{x_1}, \overline{x_2}, ..., \overline{x_n})$, в котором, для определенности, отличными от нуля переменными являются первые l переменные ($l < m$), а остальные имеют нулевое значение.

Пусть базис этого решения составляют векторы $A_1, A_2, ..., A_m$.

Пусть далее не выполняется ни условие оптимальности этого решения, ни условие неограниченности целевой функции сверху на допустимом множестве.

На рис.1.11 представлен фрагмент соответствующей симплекс-таблицы.

Баз	$C_{баз}$	A_0		c_s		c_n
				A_s		A_n
A_1	c_1	x_{10}		x_{s1}		x_{1n}
A_2	c_2	x_{20}		x_{s2}		x_{2n}
A_l	c_l	x_{l0}		x_{s3}		x_{ln}
A_{l+1}	c_{l+1}	0		$x_{l+1,s}$		$x_{l+1,n}$
A_m	c_m	0		x_{ms}		x_{mn}
		Z_0		Δ_s		Δ_n

Рис.1.11. Симплекс-таблица вырожденной задачи

Возьмем любой вектор A_S , имеющий отрицательную оценку: $\Delta_s<0,\ s\in\{1,2,...,n\}$.

Предположим, что среди элементов $x_{l+1,s},\ x_{l+2,s},\ ...,\ x_{ms}$ есть положительный элемент $x_{rs} > 0$**,** где $r\in\{l+1,l+2,...,m\}$. Тогда, учитывая, что $x_{l+1,0}= x_{l+2,0}= ... = x_{m0}=0$:

$$\theta_0 = \min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\} = \frac{x_{r0}}{x_{rs}} = 0.$$

Найдем базисные координаты опорного решения, которое будет соответствовать новому базису (этот базис образуется заменой базисного вектора A_r свободным вектором A_s):

$$x'_{ko} = x_{k0} - \frac{x_{r0}}{x_{rs}} x_{ks} = x_{k0} - \frac{0}{x_{rs}} x_{ks} = x_{k0},\quad k = 1,2,...,m;\ k \neq r;$$

$$x'_{ro} = \frac{x_{r0}}{x_{rs}} = \frac{0}{x_{rs}} = 0.$$

Таким образом, в новой симплекс-таблице столбец A_0 не изменился – *новое опорное решение совпадает со старым*. Не изменилось и значение целевой функции:

$$z' = z - \frac{x_{r0}}{x_{rs}}\Delta_s = z - \frac{0}{x_{rs}}\Delta_s = z\ ,$$

т.е. *лучшее значение целевой функции не получено*. Однако новому базису, в общем случае, соответствует *новая* симплекс-таблица: в ней записаны новые коэффициенты разложения векторов $A_1, A_2, \ldots, A_n$.

Изменились и оценки свободных векторов. Теперь может возникнуть одна из трех ситуаций:

1. Получено оптимальное решение (нет отрицательных оценок);
2. Целевая функция не ограничена сверху на допустимом множестве;
3. Решение следует продолжить.

В третьем случае, возможно, снова не удастся улучшить значение целевой функции.

Ввиду того, что количество базисов, соответствующих опорным решениям любой задачи линейного программирования, конечно, после некоторого числа переходов от одного опорного решения к другому, в принципе, возможно возникновение четвертой ситуации, а именно, *очередное опорное решение будет иметь базис, который фигурировал ранее*.

Такая ситуация называется <u>*зацикливанием*</u>. В настоящее время разработан целый ряд способов защиты от зацикливания. Рассмотрим наиболее известный из них.

1.2.12. Лексикографический симплекс-метод

Рассмотрим подробнее эффект зацикливания на примере. Решим следующую задачу линейного программирования.

<u>**Пример 1.11**</u>

$$300x_5 + 80x_6 - 1219x_7 - x_8 \to \max$$

$$x_2 - 8x_5 - 2x_6 + 30x_7 + \tfrac{1}{2}x_8 = 0$$

$$x_1 + \tfrac{19}{2}x_5 + \tfrac{5}{2}x_6 - 38x_7 - \tfrac{2}{3}x_8 = 0$$

$$x_3 + 40x_5 - 3x_6 + 90x_7 + x_8 = 1$$

$$x_4 + x_8 = 1.$$

Условимся всегда вводить в базис первый в таблице вектор с отрицательной оценкой, считая от начала, при наличии альтернатив всегда выводить из базиса первый вектор, считая сверху таблицы. В этом случае процедура решения задачи выглядит следующим образом.

Баз	$C_{баз}$	A_0	0	0	0	0	300	80	-1219	-1
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_2	0	0	0	1	0	0	-8	-2	30	1/2
A_1	0	0	1	0	0	0	**19/2**	5/2	-38	-2/3
A_3	0	1	0	0	1	0	40	-3	90	1
A_4	0	1	0	0	0	1	0	0	0	1
Табл.1		0	0	0	0	0	-300	-80	1219	1
A_2	0	0	16/19	1	0	0	0	**2/19**	-2	-7/114
A_5	300	0	2/19	0	0	0	1	5/19	-4	-4/57
A_3	0	1	-80/19	0	1	0	0	-257/19	250	217/57
A_4	0	1	0	0	0	1	0	0	0	1
Табл.2		0	600/19	0	0	0	0	-20/19	19	-1143/57
A_6	80	0	8	19/2	0	0	0	1	-19	-7/12
A_5	300	0	-2	-5/2	0	0	1	0	**1**	1/12
A_3	0	1	104	257/2	1	0	0	0	-7	-49/12
A_4	0	1	0	0	0	1	0	0	0	1
Табл.3		0	40	10	0	0	0	0	-1	-62/3
A_6	80	0	-30	-38	0	0	19	1	0	**1**
A_7	-1219	0	-2	-5/2	0	0	1	0	1	1/12
A_3	0	1	90	111	1	0	7	0	0	-7/2
A_4	0	1	0	0	0	1	0	0	0	1
Табл.4		0	38	15/2	0	0	1	0	0	-247/12
A_8	-1	0	-30	-38	0	0	19	1	0	1
A_7	-1219	0	**1/2**	2/3	0	0	-7/12	-1/12	1	0
A_3	0	1	-15	-22	1	0	147/2	7/2	0	0
A_4	0	1	30	38	0	1	-19	-1	0	0
Табл.5		0	-1159/2		0	0	4705/12	247/12	0	0
A_8	-1	0	0	**2**	0	0	-16	-4	60	1
A_1	0	1	1	4/3	0	0	-7/6	-1/6	2	0
A_3	0	0	0	-2	1	0	-56	1	30	0
A_4	0	0	0	-2	0	1	16	4	-60	0
Табл.6		0	0	-2	0	0	-284	-76	1159	0

A₂	*0*	*0*	*0*	*1*	*0*	*0*	*-8*	*-2*	*30*	*1/2*
A₁	*0*	*0*	*1*	*0*	*0*	*0*	*19/2*	*5/2*	*-38*	*-2/3*
A₃	*0*	*1*	*0*	*0*	*1*	*0*	*40*	*-3*	*90*	*1*
A₄	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*
Табл.7	*0*	*0*	*0*	*0*	*0*	*-300*	*-80*	*1219*	*1*	

Табл. 7 полностью эквивалентна табл. 1 – произошло зацикливание.

Одним из наиболее простых и эффективных способов защиты от зацикливания является лексикографический метод. Для его обоснования введем ряд определений.

Вектор q – лексикографически положителен ($q \succ 0$), если его первый отличный от нуля элемент положителен

Вектор *q лексикографически больше вектора p* ($q \succ p$), если $q - p \succ 0$. Из этого определения автоматически следует, что если p – некоторый вектор, а q – лексикографически положительный вектор, тогда $p + q \succ p$.

Вектор q_j *лексикографически минимальный* среди множества векторов $\{q_1,...,q_N\}$, если q_j все векторы этого множества за исключением самого q_j лексикографически больше вектора q_j. Это условие записывается как $q_j = lex\min\{q_1,...,q_N\}$.

Рассмотрим векторы, получаемые из строк симплекс-таблицы: $S_k = (x_{k0}, x_{k1},...x_{kn})$.

Симплекс-таблицу, все строки которой лексикографически положительны ($\forall S_k \succ 0$), будем называть *лексикографически допустимой.*

Лексикографически допустимая симплекс-таблица получается из произвольной путем перенумерации столбцов – например, если перенумеровать базисные векторы так, чтобы им соответствовали первые m столбцов симплекс-таблицы, то новая симплекс-таблица будет лексикографически допустимой.

Пусть имеется лексикографически допустимая симплекс таблица, в которой не срабатывает ни признак оптимальности, ни признак неограниченности сверху целевой функции, т.е. имеется вектор A_s с отрицательной оценкой $\Delta_s < 0$. Рассмотрим следующие векторы, получающиеся из строк этой симплекс таблицы: $\tilde{S}_k = (\frac{x_{k0}}{x_{ks}}, \frac{x_{k1}}{x_{ks}}, ..., \frac{x_{kn}}{x_{ks}})$ (для всех $x_{ks} > 0$).

<u>ЛЕММА 1.2.</u> *Если симплекс-таблица лексикографически допустима, а номера вводимого и выводимого из базиса векторов таковы, что выполняется условие*

$$\tilde{S}_r = (\frac{x_{r0}}{x_{rs}}, \frac{x_{r1}}{x_{rs}}, ..., \frac{x_{rn}}{x_{rs}}) = lex \min_{x_{ks}>0} \{\tilde{S}_k\}, \tag{1.49}$$

то новая симплекс-таблица будет также лексикографически допустимой.

<u>Доказательство</u>. Из того факта, что $x_{rs} > 0$, следует лексикографическая положительность вектора $\tilde{S}_r$. Заменим в базисе вектор A_r соответствующий этому вектору $\tilde{S}_r$ на вектор A_s. Запишем основные формулы пересчета коэффициентов в векторном виде:

$$S'_r = \tilde{S}_r \tag{1.50}$$

$$S'_k = S_k - \tilde{S}_r x_{ks} \tag{1.51}$$

Из (1.50) следует, что $S' \succ 0$, т.е. строка *r* новой симплекс-таблицы лексикографически положительна. Рассмотрим (1.51). Если $x_{ks} < 0$, то автоматически $S'_k \succ 0$, поскольку она получается из суммы двух лексикографически положительных строк.

Если $x_{ks} > 0$, то перепишем (1.51) в следующем виде: $S'_k = x_{ks}\left(\frac{S_k}{x_{ks}} - \tilde{S}_r\right) = x_{ks}\left(\tilde{S}_k - \tilde{S}_r\right)$. Из (1.49) следует, что вектор в скобках лексикографически положителен, а значит строка *k* новой симплекс-таблицы лексикографически положительна при любых x_{ks}, что и требовалось доказать.

<u>ТЕОРЕМА 1.5.</u> *Если на каждом шаге симплекс-метода при выборе вводимого и выводимого из базиса векторов выполняется условие (1.49), то количество шагов, которое необходимо осуществить до остановки по признаку оптимальности или неограниченности целевой функции сверху, конечно.*

<u>Доказательство.</u> Рассмотрим вектор, получаемый из строки с оценками $D=(\Delta_1,...,\Delta_n)$. При переходе к новому опорному решению этот вектор пересчитывается по следующей формуле: $D'=D-\tilde{S}_r\Delta_s$. Поскольку $\Delta_s<0$, а $\tilde{S}_r$ –всегда лексикографически положителен, то для каждого шага симплекс-метода $D'\succ D$. Однако неравенство типа «лексикографическое больше» является строгим, а поэтому при выполнении условия (1.49) невозможен возврат к предыдущему базису (это означало бы, что на каком-то из шагов выполняется условие $D\succ D'$). Теорема доказана.

Сформулируем алгоритм лексикографического симплекс-метода по шагам.

<u>Шаг 0.</u> Привести симплекс-таблицу к лексикографически допустимому виду.

<u>Шаг 1.</u> Пусть

$$\theta_0=\min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\}=\frac{x_{q_1 0}}{x_{q_1 s}}=\frac{x_{q_2 0}}{x_{q_2 s}}=...=\frac{x_{q_t 0}}{x_{q_t s}},$$

т.е. имеется t базисных векторов, каждый из которых может быть заменен небазисным вектором A_s. Если $t=1$, из базиса выводится вектор A_{q_1}. В противном случае производится вычисление:

<u>Шаг 2.</u>

$$\theta_1=\min_{x_{ks}>0}\left\{\frac{x_{k1}}{x_{ks}}\right\}=\frac{x_{q_1 1}}{x_{q_1 s}}=\frac{x_{q_2 1}}{x_{q_2 s}}=...=\frac{x_{q_p 1}}{x_{q_p s}},$$

т.е. вместо вектора A_0 используется вектор A_1.

Теперь имеется p базисных векторов, каждый из которых может быть заменен небазисным вектором A_s. Если $q=1$, из базиса выводится вектор A_{q_1}. В противном случае производится аналогичные вычисление, но уже с использованием вектора A_2 (**<u>шаг 3</u>**), и так далее. В итоге, учитывая то обстоятельство, что в симплекс-таблице каждая строка отличается от любой другой, по крайней мере, одним коэффициентом, будет найден единственный вектор, который нужно вывести из базиса.

Рассмотрим работу этого правила на примере 1.11.

Пример 1.11(продолжение)

Исходная табл. 1 уже является лексикографически допустимой, поэтому остается лишь следовать алгоритму лексикографического симплекс-метода при определении выводимого из базиса вектора, чтобы избежать зацикливания. В табл. 1 альтернатива отсутствует – единственный вектор, который может быть выведен из базиса – это вектор A_1:

Баз	$C_{баз}$	A_0	*0*	*0*	*0*	*0*	*300*	*80*	*-1219*	*-1*
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_2	*0*	*0*	*0*	*1*	*0*	*0*	*-8*	*-2*	*30*	*1/2*
A_1	*0*	*0*	*1*	*0*	*0*	*0*	***19/2***	*5/2*	*-38*	*-2/3*
A_3	*0*	*1*	*0*	*0*	*1*	*0*	*40*	*-3*	*90*	*1*
A_4	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*
Табл. 1		*0*	*0*	*0*	*0*	*0*	*-300*	*-80*	*1219*	*1*
A_2	*0*	*0*	*16/19*	*1*	*0*	*0*	*0*	*2/19*	*-2*	*-7/114*
A_5	*300*	*0*	*2/19*	*0*	*0*	*0*	*1*	***5/19***	*-4*	*-4/57*
A_3	*0*	*1*	*-80/19*	*0*	*1*	*0*	*0*	*-257/19*	*250*	*217/57*
A_4	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*
Табл.2		0	600/19	0	0	0	0	-20/19	19	-1143/57

Однако, в табл. 2 появляются две альтернативы:

$$\theta_0 = \min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\} = \frac{x_{10}}{x_{1s}} = \frac{x_{20}}{x_{2s}} = 0.$$

Рассмотрим столбец A_1 вместо столбца A_0:

$$\theta_1 = \min_{x_{ks}>0}\left\{\frac{x_{k1}}{x_{ks}}\right\} = \min\left\{\frac{16/19}{2/19};\frac{2/19}{5/19}\right\} = \frac{2}{5} = \frac{x_{21}}{x_{2s}}.$$

Выведем из базиса вектор A_5. В этой таблице проиллюстрировано решение, принимаемое с помощью лексикографического правила, отличающееся от решения, принимаемого с использованием обычного симплекс-метода. Продолжим решение задачи.

Баз	$C_{баз}$	A_0	*0*	*0*	*0*	*0*	*300*	*80*	*-1219*	*-1*
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_2	*0*	*0*	*4/5*	*1*	*0*	*0*	*-2/5*	*0*	*-2/5*	*-1/30*
A_6	*80*	*0*	*2/5*	*0*	*0*	*0*	*19/5*	*1*	*-76/5*	*-4/15*
A_3	*0*	*1*	*6/5*	*0*	*1*	*0*	*257/5*	*0*	*222/5*	*1/5*
A_4	*0*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*
Табл.3		0	32	0	0	0	4	0	3	-61/3
A_2	*0*	*1/30*	*4/5*	*1*	*0*	*1/30*	*-2/5*	*0*	*-2/5*	*-7/114*
A_6	*80*	*4/15*	*2/5*	*0*	*0*	*4/15*	*19/5*	*1*	*-76/5*	*-4/57*
A_3	*0*	*4/5*	*6/5*	*0*	*1*	*-1/5*	*257/5*	*0*	*222/5*	*217/57*
A_8	*-1*	*1*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*1*
Табл.4		61/3	32	0	0	61/3	4	0	3	0

В табл. 4 получено оптимальное решение.

1.2.13. Метод вспомогательной задачи

В том случае, когда система ограничений задачи не содержит полного набора единичных векторов, который можно было бы принять в качестве базиса исходного опорного решения, прибегают к методу *искусственного базиса*.

Основная идея этого метода –использование симплекс-метода для отыскания опорного решения и вычисления коэффициентов разложения всех векторов задачи по базису найденного решения.

Для этого в задачу вводятся *искусственные переменные*, векторы коэффициентов при которых называются *искусственными векторами*.

Целевая функция строится таким образом, чтобы в процессе решения задачи симплекс-методом при переходе от одного опорного решения к другому искусственные векторы, входящие в базис, заменялись векторами основной задачи – *основными векторами*.

В зависимости от конкретного способа построения ЦФ, в результате решения новой задачи формируется допустимое или оптимальное решение исходной задачи или устанавливается факт неразрешимости последней.

Наиболее популярными способами реализации метода искусственного базиса являются два метода: *метод вспомогательной задачи* ("двухэтапный метод") и *метод М-задачи* ("метод больших штрафов"). Рассмотрим сначала первый метод.

Пусть дана задача ЛП:

$$
\begin{array}{l}
c_1x_1 + c_2x_2 + \ldots + c_nx_n \rightarrow max \\
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = a_m \\
x_j \geq 0,\ j=1,2,\ldots,n,
\end{array}
\tag{1.52}
$$

в которой все свободные члены неотрицательны ($a_i \geq 0$, $i=1,2,\ldots,m$).

Вспомогательная задача имеет вид:

$$
\begin{array}{l}
-x_{n+1} - x_{n+2} - \ldots - x_{n+m} \rightarrow max, \\
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} \qquad = a \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \quad + x_{n+2} \quad = a_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \qquad + x_{n+m} = a_m \\
x_j \geq 0,\ j=1,2,\ldots,n+m.
\end{array}
\tag{1.53}
$$

В этой задаче переменные $x_{n+1}, x_{n+2},\ldots,x_{n+m}$ – искусственные. Им соответствует полный набор единичных (искусственных) векторов.

Задача (1.53) имеет очевидное опорное решение:

$$x' = (0,0,\ldots,0,a_1,a_2,\ldots,a_m),$$

в котором первые n нулей – значения основных переменных $x_1,x_2,\ldots,x_n$, а базисные переменные $x_{n+1}, x_{n+2},\ldots,x_{n+m}$ (искусственные) имеют значения $a_1, a_2, \ldots, a_m$ соответственно.

Это решение имеет единичный базис:

$$B=(A_{n+1},A_{n+2},\ldots,A_{n+m})=E,$$

следовательно, можно заполнить первую симплекс-таблицу и решить задачу симплекс-методом.

Задача (1.53) всегда имеет оптимальное решение, так как ее ЦФ ограничена сверху на допустимом множестве - не может принять значение большее нуля.

Пусть получено оптимальное решение вспомогательной задачи:

$$x^* = (x_1^*, x_2^*, \ldots, x_n^*, x_{n+1}^*, x_{n+2}^*, \ldots, x_{n+m}^*) \tag{1.54}$$

В зависимости от того, входят ли искусственные векторы в базис оптимального решения (1.54) и какие при этом значения имеют искусственные переменные, относительно исходной задачи (1.52) можно сделать различные выводы.

Случай 1. Среди чисел $x^*_{n+1}, x^*_{n+2}, ..., x^*_{n+m}$ есть отличные от нуля. В этом случае исходная задача не имеет допустимых решений.

Действительно, предположим от противного, что исходная задача имеет допустимое решение

$$x'=(x'_1, x'_2, ..., x'_n).$$

Тогда очевидно, что вектор

$$x''=(x'_1, x'_2, ..., x'_n, 0, 0, ..., 0),$$

в котором все искусственные переменные имеют нулевое значение, является допустимым решением вспомогательной задачи (1.53). На этом решении ее ЦФ принимает нулевое значение - большее, чем на оптимальном решении (1.54), так как среди чисел $x^*_{n+1}, x^*_{n+2}, ..., x^*_{n+m}$ есть строго положительные.

Полученное противоречие доказывает неправомерность предположения о допустимости исходной задачи.

Пример 1.12. Дана задача линейного программирования:

$$2x_1 - 15x_2 \rightarrow \max$$
$$x_1 + 4x_2 = 2$$
$$3x_1 - 8x_2 = 7$$
$$x_1, x_2 \geq 0.$$

Вспомогательная задача имеет вид:

$$-x_3 - x_4 \rightarrow \max$$
$$x_1 + 4x_2 + x_3 = 2$$
$$3x_1 - 8x_2 + x_4 = 7$$
$$x_1, x_2 \geq 0.$$

Ниже представлено решение вспомогательной задачи:

			0	0	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_3	-1	2	**1**	4	1	0
A_4	-1	7	3	-8	0	1
Табл.1		-9	-4	4	0	0
			0	0	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_1	0	2	1	4	1	0
A_4	-1	1	0	-20	-3	1
Табл.2		-1	0	20	4	0

Ввиду того, что искусственная переменная x_4 в оптимальном решении вспомогательной задачи отлична от нуля, исходная задача не имеет допустимых решений.

Случай 2. В оптимальном решении вспомогательной задачи все искусственные переменные имеют нулевое значение. В этом случае существенным фактором является вхождение искусственных переменных в базис оптимального решения вспомогательной задачи.

Случай 2а. В оптимальном базисе нет искусственных векторов. В этом случае получено опорное решение исходной задачи и известно разложение всех векторов задачи по базису опорного решения.

Для решения исходной задачи достаточно:

- в полученной симплекс-таблице отбросить столбцы, связанные с искусственными переменными;
- восстановить коэффициенты целевой функции исходной задачи;
- пересчитать значение ЦФ и оценки всех свободных векторов.

Пример 1.13

Дана задача линейного программирования:

$$x_1 + 4x_2 + x_3 \to max$$
$$x_1 - x_2 + x_3 = 3$$
$$2x_1 - 5x_2 - x_3 = 0$$
$$x_1, x_2, x_3 \geq 0.$$

Вспомогательная задача имеет вид:

$$-x_4 - x_5 \to max$$
$$x_1 - x_2 + x_3 + x_4 = 3$$
$$2x_1 - 5x_2 - x_3 + x_5 = 0$$
$$x_j \geq 0, (j=1\div 5).$$

Ниже представлено решение вспомогательной задачи:

			0	0	0	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5
A_4	-1	3	1	-1	1	1	0
A_5	-1	0	**2**	-5	-1	0	1
Табл. 1		-3	-3	6	0	0	0
A_4	-1	3	0	**3/2**	3/2	1	-1/2
A_1	0	0	1	-5/2	-1/2	0	1/2
Табл.2		-3	0	-3/2	-3/2	0	3/2
A_2	0	2	0	1	1	2/3	-1/3
A_1	0	5	1	0	2	5/3	-1/3
Табл.3		0	0	0	0	1	1

Следует отметить, что во второй симплекс-таблице столбец A_0 не изменился – новое опорное решение совпадает со старым. Не изменилось и значение целевой функции. Это связано с тем, что исходное опорное решение вспомогательной задачи – вырожденное.

Оптимальное решение вспомогательной задачи:

$$x_{всп.}=(5,2,0,0,0).$$

В базисе нет искусственных векторов, следовательно, получено опорное решение исходной задачи $x_{.}=(5,2,0)$, базис которого составляют векторы A_2, A_1.

Теперь, отбросив столбцы, связанные с искусственными векторами, восстановив коэффициенты целевой функции, пересчитав значение целевой функции и оценки свободных векторов, сформируем симплекс-таблицу для решения исходной задачи симплекс-методом.

Баз	$C_{баз}$	A_0	1	4	1
			A_1	A_2	A_3
A_2	4	2	0	1	1
A_1	1	5	1	0	2
Табл.3		13	0	0	5

Оценки всех векторов не отрицательные. Следовательно, в этой таблице записано оптимальное решение исходной задачи:

$$x_{опт}=(5,2,0);\ Z_{опт}=13.$$

Случай 2b. В оптимальный базис вспомогательной задачи входят искусственные векторы.

Рассмотрим симплекс-таблицу, соответствующую оптимальному решению вспомогательной задачи в случае 2b.

Баз	$C_{баз}$	A_0	0	0	…	0	-1	…	-1
			A_1	A_2		A_n	A_{n+1}		A_{n+m}
A_{i_1}	0	x_{10}	x_{11}	x_{12}		x_{1n}	$x_{1,n+1}$		$x_{1,n+m}$
…	…	…	…	…	…	…	…	…	…
A_{i_l}	0	x_{l0}	x_{ll}	x_{l2}		x_{ln}	$x_{l,n+1}$	…	$x_{l,n+m}$
$A_{i_{l+1}}$	-1	0					$x_{l+1,n+1}$	…	$x_{l+1,n+m}$
…	…	…		$\{x_{rs}\}$			…	…	…
A_{i_m}	-1	0					$x_{m,n+1}$	…	$x_{m,n+m}$
		Z'	Δ_1	Δ_2		Δ_n	Δ_{n+1}	…	Δ_{n+m}

Здесь для определенности принято, что первые l базисные векторы – основные (векторы исходной задачи): A_{i_1}, A_{i_2},..., A_{i_l}. Векторы $A_{i_{l+1}}$, $A_{i_{l+2}}$,..., A_{i_m} – искусственные.

Важно знать, существуют ли среди коэффициентов разложения основных векторов при искусственных базисных векторах отличные от нуля. Область этих коэффициентов выделена двойной рамкой.

<u>Случай 2b1</u>. Все $x_{rs} = 0$ ($r = \overline{l+1,\ m}$; $s = \overline{1,\ n}$), т.е. все коэффициенты в отмеченной области имеют нулевое значение.

В этом случае любой вектор исходной задачи A_j ($j = \overline{1,\ n}$) линейно выражается через основные векторы $A_{i_1}, A_{i_2}, ..., A_{i_l}$, входящие в базис оптимального решения вспомогательной задачи:

$$A_j = x_{1j} A_{i_1} + x_{2j} A_{i_2} + ... + x_{lj} A_{i_l}.$$

Таким образом, систему векторов (A_{i_1}, $A_{i_2}, ..., A_{i_l}$) можно принять в качестве базиса опорного решения исходной задачи. В этом опорном решении базисные переменные $x_{i_1}, x_{i_2}, ..., x_{i_l}$ имеют значения $x_{1j}, x_{2j}, ..., x_{lj}$, соответственно, а свободные (остальные) – нулевое значение.

Отбросив строки и столбцы, связанные с искусственными переменными, можно перейти к решению основной задачи. Для этого нужно:

- восстановить коэффициенты целевой функции;
- пересчитать значение целевой функции;
- вычислить оценки свободных векторов.

В соответствующей симплекс-таблице будет l строк – меньше, чем количество ограничений исходной задачи. Это говорит о том, что в исходной задаче имеется m-l ограничений-уравнений, которые линейно выражаются через остальные уравнения, т.е. ранг матрицы системы ограничений равен l.

Пример 1.14

Дана задача линейного программирования:

$$4x_1 + 5x_2 + 3x_3 - 7x_4 \rightarrow max$$
$$x_1 \quad + 2x_3 + x_4 = 10$$
$$x_2 + x_3 - 2x_4 = 2$$
$$4x_2 + 4x_3 - 8x_4 = 8$$
$$x_1, x_2, x_3, x_4 \geq 0.$$

Вспомогательная задача имеет вид:

$$-x_5 - x_6 \rightarrow \max$$
$$x_1 \quad + 2x_3 + x_4 \qquad = 10$$
$$x_2 + x_3 - 2x_4 + x_5 \quad = 2$$
$$4x_2 + 4x_3 - 8x_4 \quad + x_6 = 8$$
$$x_1, x_2, x_3, x_4 \geq 0$$
$$x_j \geq 0,\ (j=1\div 6).$$

Ниже представлено решение вспомогательной задачи:

			0	0	0	0	-1	-1
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
A_1	0	10	1	0	2	1	0	0
A_5	-1	2	0	**1**	1	-2	1	0
A_6	-1	8	0	4	4	-8	0	1
Табл.1		-10	0	-5	-5	10	0	0
A_1	0	10	1	0	2	1	0	0
A_2	0	2	0	1	1	-2	1	0
A_6	-1	**0**	**0**	**0**	**0**	**0**	-4	1
Табл.2		0	0	0	0	0	5	0

Как видно из таблицы, коэффициенты разложения всех основных векторов при искусственном базисном векторе A_6 имеют нулевое значение. Следовательно, можно переходить к решению исходной задачи, восстановив коэффициенты целевой функции, подсчитав ее значение и вычислив оценки свободных векторов A_3 и A_4 .

Решение основной задачи представлено ниже.

			4	5	3	-7
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_1	4	10	1	0	2	1
A_2	5	2	0	1	1	-2
Табл.1		50	0	0	10	1

Оптимальное решение исходной задачи:

$$x_{опт}=(10, 2, 0, 0);\ Z_{опт}=50.$$

Случай 2b2. $\exists\ x_{rs}\neq 0$ ($r=\overline{l+1,\ m}$; $s=\overline{1,\ n}$), т.е. среди коэффициентов в области, выделенной в оптимальной симплекс-таблице вспомогательной задачи, имеются отличные от нуля.

В этом случае получено допустимое решение исходной задачи, однако для того чтобы продолжить решение, необходимо знать, какие основные векторы в дополнение к векторам $A_{i_1}, A_{i_2}, ..., A_{i_l}$ составляют базис найденного решения.

Для того чтобы получить опорное решение исходной задачи, найти базис этого решения и коэффициенты разложения основных векторов по найденному базису, используется следующий прием.

Пусть $x_{rs}\neq 0$ ($r=\overline{l+1,\ m}$; $s=\overline{1,\ n}$). Тогда основной вектор A_s принудительно вводится в базис вместо искусственного вектора A_{i_r} и по обычным правилам пересчитывается симплекс-таблица.

Ввиду того, что $x_{r0}=0$ (оптимальное решение вспомогательной задачи – вырожденное), столбец A_0 не изменится.

Аналогичная процедура повторяется до тех пор, пока не возникнет одна из двух рассмотренных ранее ситуаций: 2a или 2b1. И в том, и в другом случае будет получено опорное решение исходной задачи и базис этого решения, состоящий только из основных векторов.

Пример 1.15

Дана задача линейного программирования:

$$-2x_1 + x_2 \quad + 6x_3 - 4x_4 \to max,$$
$$x_1 + x_2 \quad + x_3 \quad = 4$$
$$x_2 -(1/3)\, x_3 + x_4 = 2$$
$$-x_1 +2x_2 \quad - x_3 +3x_4 = 2$$
$$x_1, x_2, x_3, x_4 \geq 0.$$

Вспомогательная задача имеет вид:

$$-x_5 - x_6 - x_7 \to max$$
$$x_1 + x_2 \quad + x_3 \quad +x_5 \quad = 4$$
$$x_2 -(1/3)\, x_3 + x_4 \quad +x_6 \quad = 2$$
$$-x_1 +2x_2 \quad - x_3 +3x_4 \quad +x_7 = 2$$
$$x_j \geq 0,\ (j=1\div 7).$$

Ниже представлено решение вспомогательной задачи:

Баз	$C_{баз}$	A0	0	0	0	0	-1	-1	-1
			A1	A2	A3	A4	A5	A6	A7
A5	-1	4	1	1	1	0	1	0	0
A6	-1	2	0	1	-1/3	1	0	1	0
A7	-1	2	-1	**2**	-1	3	0	0	1
Табл.1		-8	0	-4	1/3	-4	0	0	0
A5	-1	3	**3/2**	0	3/2	-3/2	1	0	-1/2
A6	-1	1	1/2	0	1/6	-1/2	0	1	-1/2
A2	0	1	-1/2	1	-1/2	3/2	0	0	1/2
Табл.2		-4	-2	0	-5/3	2	0	0	2
A1	0	2	1	0	1	-1	2/3	0	-1/3
A6	-1	**0**	**0**	**0**	**-1/3**	**0**	-1/3	1	-1/3
A2	0	2	0	1	0	1	1/3	0	1/3
Табл.3		0	0	0	1/3	0	4/3	0	4/3

Коэффициент разложения основного вектора A_3 по базису оптимального решения вспомогательной задачи при искусственном векторе A_6 отличен от нуля (x_{23} = -1/3). Следовательно, вектор A_6 нужно вывести из базиса, а основной вектор A_3 – ввести в базис. В результате будет получена симплекс-таблица, в которой все базисные векторы – основные:

Баз	$C_{баз}$	A 0	0	0	0	0	-1	-1	-1
			A 1	A 2	A 3	A 4	A 5	A 6	A 7
A1	0	2	1	0	0	-1	-1/3	3	-4/3
A3	0	0	0	0	1	0	1	-3	1
A2	0	2	0	1	0	1	1/3	0	1/3
Табл.4		0	0	0	0	0	1	1	1

Теперь можно приступить к решению основной задачи (как в случае 2а). Для этого:

- восстанавливаются коэффициенты ЦФ;
- вычисляется значение ЦФ на полученном опорном решении;
- вычисляются оценки свободных векторов (в данном случае это единственный вектор A_4).

В результате формируется следующая симплекс-таблица:

Баз	$C_{баз}$	A 0	-2	1	6	-4
			A 1	A 2	A 3	A 4
A1	-2	2	1	0	0	-1
A3	6	0	0	0	1	0
A2	1	2	0	1	0	1
Табл. 5		-2	0	0	0	7

Ввиду того, что все оценки неотрицательны, решение, записанное в этой таблице,– оптимальное:

$$x_{опт}=(2,2,0,0); \quad Z_{опт}=-2.$$

1.2.14. Метод М-задачи

Пусть дана задача ЛП:

$$\begin{array}{l} c_1x_1 + c_2x_2 + \ldots + c_nx_n \rightarrow max \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = a_m \\ x_j \geq 0, \ j=1,2,\ldots,n, \end{array} \qquad (1.55)$$

в которой все свободные члены неотрицательны ($a_i \geq 0$, $i=1,2,\ldots,m$).

Как и в двухэтапном методе, в систему ограничений этой задачи вводятся искусственные переменные $x_{n+1}, x_{n+2},\ldots,x_{n+m}$ таким образом, чтобы получить полную систему единичных векторов. Однако целевая функция строится иначе:

$$\begin{array}{l} c_1x_1 + c_2x_2 + \ldots + c_nx_n - Mx_{n+1} - Mx_{n+2} - \ldots - Mx_{n+m} \rightarrow max \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} \qquad\qquad = a_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \quad + x_{n+2} \qquad = a_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \qquad + x_{n+m} = a_m \\ x_j \geq 0, \ j=1,2,\ldots,n+m. \end{array} \qquad (1.56)$$

Здесь M - достаточно большое число.

Задача (1.56) называется <u>*М-задачей*</u>.

Специфика M-задачи заключается в том, что для ее решения можно не придавать M конкретного значения. Действительно, при работе по симплекс-методу на каждой итерации исследуются знаки оценок свободных векторов, а не сами оценки. В результате принимается одно из трех решений:

- очередной план можно улучшить;
- очередной план – оптимальный;
- ЦФ не ограничена сверху.

Рассмотрим вопрос вычисления оценок при решении M-задачи.

Оценка свободного вектора A_j вычисляется по формуле:

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} x_{kj} - c_j,$$

где i_k ($k=1,2,...,m$) – номера векторов, входящих в базис очередного опорного решения.

Ввиду того, что среди коэффициентов ЦФ M-задачи есть коэффициенты вида -M, и M не входит в ограничения этой задачи, оценку свободного вектора A_j можно представить в виде:

$$\Delta_j = \Delta'_j M + \Delta''_j ,$$

где Δ'_j и Δ''_j не зависят от M, и, в случае достаточно большого положительного M, знак оценки Δ_j определяется знаком Δ'_j.

Этот факт позволяет не придавать M конкретного значения - просто считается что M – достаточно большое положительное число. При этом для решения M-задачи становится возможным использование основной процедуры симплекс-метода.

Единственная особенность при использовании симплекс-метода заключается в том, что вместо одной строки для хранения и вычисления оценок, используются две – одна для Δ'_j , а другая – для Δ''_j .

Это относится и к ЦФ. Учитывая, что в случае M-задачи значение ЦФ можно представить в виде:

$$Z = Z'M + Z'',$$

Z' и Z'' можно хранить и вычислять также раздельно.

Пример 1.16

Дана задача ЛП:

$$x_1+4x_2+x_3 \to max$$
$$x_1 - x_2 + x_3 = 3$$
$$2x_1 - 5x_2 - x_3 = 0$$
$$x_1, x_2, x_3 \geq 0.$$

M-задача имеет вид:

$$x_1+4x_2+x_3 - Mx_4 - Mx_5 \to max$$
$$x_1 - x_2 + x_3 + x_4 = 3$$
$$2x_1 - 5x_2 - x_3 + x_5 = 0$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0.$$

Исходное опорное решение этой задачи:

$$x=(0,0,0,3,0).$$

Ниже представлена первая симплекс-таблица, подготовленная для решения *M*-задачи.

			1	4	1	-M	-M
Баз	$C_{баз}$	A_0	A1	A2	A3	A4	A5
A4	-M	3	1	-1	1	1	0
A5	-M	0	2	-5	-1	0	1
Табл.1	Δ'_j	-3	-3	6	0	0	0
	Δ''_j	0	-1	-4	-1	0	0

Для общности в этой таблице приняты обозначения Δ'_0 и Δ''_0 вместо Z' и Z'', соответственно, т.е. значение ЦФ представлено в виде:

$$\Delta_0 = \Delta'_0 M + \Delta''_0 = -3M+0.$$

Оценки свободных векторов подсчитываются следующим образом:

$\Delta_1=\Delta'_1 M + \Delta''_1= -3M-1$ (отрицательная оценка);

$\Delta_2=\Delta'_2 M + \Delta''_2= 6M-4$ (положительная оценка);

$\Delta_3=\Delta'_3 M + \Delta''_3=-0M-1$ (отрицательная оценка);

Перед тем, как продолжить решение задачи, рассмотрим ситуации, возможные при решении *M*-задачи.

Случай 1. *M*-задача имеет оптимальное решение.

Обозначим это решение следующим образом:

$$x^{*M} = (x_1^*, x_2^*, \ldots, x_n^*, x_{n+1}^*, x_{n+2}^*, \ldots, x_{n+m}^*). \tag{1.57}$$

Принципиально важно знать, все ли искусственные переменные в этом решении имеют нулевое значение.

Случай 1а. В оптимальном решении (1.57) среди чисел $x_{n+1}^*, x_{n+2}^*, ..., x_{n+m}^*$ есть отличные от нуля. Тогда исходная задача не имеет допустимых решений.

Доказательство. На решении (1.57) ЦФ *M*-задачи принимает значение:

$$Z^{*M} = \sum_{j=1}^{n} c_j x_j^* - M \sum_{j=n+1}^{n+m} x_j^*, \text{ причем } \sum_{j=n+1}^{n+m} x_j^* > 0 .$$

Предположим от противного, что исходная задача имеет допустимое решение $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$. Тогда очевидно, что $\overline{x}^M = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n, 0, 0, ..., 0)$ – допустимое решение *M*-задачи. На этом решении ЦФ М-задачи принимает значение

$$\overline{Z}^M = \sum_{j=1}^{n} c_j \overline{x}_j .$$

Сравним Z^{*M} и $\overline{Z}^M$. Очевидно, что всегда можно найти такое значение *M*, при котором будет иметь место следующее соотношение:

$$Z^{*M} = \sum_{j=1}^{n} c_j x_j^* - M \sum_{j=n+1}^{n+m} x_j^* < \overline{Z}^M = \sum_{j=1}^{n} c_j \overline{x}_j ,$$

т.е., $\overline{x}^M = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n, 0, 0, ..., 0)$ является допустимым решением *M*-задачи, доставляющим целевой функции значение лучшее, чем оптимальное.

Это противоречие доказывает неправомерность предположения о существовании допустимого решения исходной задачи.

Случай 1b. В оптимальном решении (1.57) все искусственные переменные $x_{n+1}^*, x_{n+2}^*, ..., x_{n+m}^*$ имеют нулевое значение. Тогда решение $x^* = (x_1^*, x_2^*, ..., x_n^*)$ является оптимальным решением исходной задачи.

Доказательство. На решении (1.57) ЦФ *M*-задачи принимает значение:

$$Z^{*M} = \sum_{j=1}^{n} c_j x_j^* - M \sum_{j=n+1}^{n+m} x_j^* = \sum_{j=1}^{n} c_j x_j^* .$$

Очевидно, что $x^* = (x_1^*, x_2^*, ..., x_n^*)$ – допустимое решение исходной задачи, так как по условию все искусственные переменные $x_{n+1}^*, x_{n+2}^*, ..., x_{n+m}^*$ имеют нулевое значение.

Векторы A_j ($j \in \{1,2,...,n\}$), соответствующие положительным координатам этого решения составляют линейно-независимую систему, так как входят в базис оптимального решения *M*-задачи. Следовательно, $x^* = (x_1^*, x_2^*, ..., x_n^*)$ – опорное решение исходной задачи.

На решении X^* ЦФ исходной задачи принимает значение $Z^* = \sum_{j=1}^{n} c_j x_j^*$, т.е. $Z^{*M} = Z^*$.

Пусть теперь $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ - любое допустимое решение исходной задачи. На этом решении ЦФ исходной задачи принимает значение $\overline{Z} = \sum_{j=1}^{n} c_j \overline{x}_j$. Очевидно, что

$$\overline{x}^M = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n, 0, 0, ..., 0) \;-$$

допустимое решение *M*-задачи, причем, на этом решении ЦФ *M*-задачи принимает значение

$$\overline{Z}^M = \sum_{j=1}^{n} c_j \overline{x}_j - M \sum_{j=n+1}^{n+m} \overline{x}_j = \sum_{j=1}^{n} c_j \overline{x}_j \; .$$

Следовательно, $\overline{Z}^M = \overline{Z}$.

Ввиду того, что Z^{*M} – оптимальное значение ЦФ *M*-задачи, имеет место соотношение:

$$\overline{Z}^M = \overline{Z} \le Z^{*M} = Z^*.$$

Учитывая то обстоятельство, что $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ – произвольно взятое допустимое решение исходной задачи, и на этом решении ЦФ принимает значение, не большее, чем на решении $x^* = (x_1^*, x_2^*, ..., x_n^*)$, можно утверждать, что x^* – оптимальное решение исходной задачи, что и требовалось доказать.

Пример 1.17

Дана задача ЛП:

$$x_1+4x_2+x_3 \to max$$
$$x_1 - x_2 + x_3 = 3$$
$$2x_1 - 5x_2 - x_3 = 0$$
$$x_1, x_2, x_3 \ge 0.$$

M-задача имеет вид:

$$x_1+4x_2+x_3 - Mx_4 - Mx_5 \to max$$
$$x_1 - x_2 + x_3 + x_4 = 3$$
$$2x_1 - 5x_2 - x_3 + x_5 = 0$$
$$x_1, x_2, x_3, x_4, x_5 \ge 0.$$

Исходное опорное решение этой задачи:

$$x=(0,0,0,3,0).$$

Ниже представлена первая симплекс-таблица, подготовленная для решения *M*-задачи.

Баз	$C_{баз}$	A_0	1	4	1	-M	-M
			A1	A2	A3	A4	A5
A4	-M	3	1	-1	1	1	0
A5	-M	0	2	-5	-1	0	1
Табл.1	Δ'_j	-3	-3	6	0	0	0
	Δ''_j	0	-1	-4	-1	0	0
Баз	$C_{баз}$	A0	1	4	1	-M	-M
			A1	A2	A3	A4	A5
A	-M	3	0	3/2	3/2	1	-1/2
A1	1	0	1	-5/2	-1/2	0	1/2
Табл.2	Δ'_j	-3	0	-3/2	-3/2	0	3/2
	Δ''_j	0	0	-13/2	-3/2	0	1/2
Баз	$C_{баз}$	A0	1	4	1	-100	-100
			A1	A2	A3	A4	A5
A2	4	2	0	1	1	2/3	-1/3
A1	1	5	1	0	2	5/3	-1/3
Табл.3	Δ'_j	0	0	0	0	1	1
	Δ''_j	13	0	0	5	13/3	-5/3

Оптимальное решение *M*-задачи:

$$Z_M = 13; \; x_M = (5,2,0,0,0).$$

Оптимальное решение исходной задачи:

$$Z_{опт} = 13; \; x_{опт} = (5,2,0,0,0).$$

Наконец, последний случай, возможный при решении *M*-задачи. Этот случай оформим в виде следующего утверждения.

<u>Случай 2</u>. Если при любом значении M M-задача имеет неограниченную сверху целевую функцию, тогда и целевая функция исходной задачи не ограничена сверху на допустимом множестве.

1.2.15. Роль оценок в задаче линейного программирования

На каждом шаге симплекс-метода в соответствующей симплекс-таблице записана некоторая задача ЛП и очередное опорное решение этой задачи с единичным базисом.

Это – не исходная задача, а та, в которую она преобразовалась в процессе работы по симплекс-методу:

$$\left.\begin{aligned}
&\sum_{k=1}^{m} c_{i_k} x_{i_k} + \sum_{j\in\omega} c_j x_j \to \max \\
&x_{i_1} + \sum_{j\in\omega} x_{1j} x_j = x_{10} \\
&x_{i_2} + \sum_{j\in\omega} x_{2j} x_j = x_{20} \\
&\dots\dots\dots\dots\dots\dots\dots \\
&x_{i_m} + \sum_{j\in\omega} x_{mj} x_j = x_{m0} \\
&x_j \ge 0,\ j=\overline{1,n}
\end{aligned}\right\} \qquad (1.58)$$

Здесь

$\{ i_1, i_2, \dots, i_m \}$ – номера базисных векторов $A_{i_1}, A_{i_2}, \dots A_{i_m}$, превратившихся в задаче (1.58) в полную систему единичных векторов;

$\omega = \{ 1,2,\dots,n \} \setminus \{ i_1, i_2, \dots, i_m \}$ – номера свободных переменных.

Выразим базисные переменные x_{i_k} $(k=\overline{1,m})$ через свободные переменные:

$$x_{i_k} = x_{k0} - \sum_{j\in\omega} x_{kj} x_j . \qquad (1.59)$$

Подставим выражения (1.59) в ЦФ задачи (1.58):

$$\sum_{k=1}^{m} c_{i_k}(x_{k0} - \sum_{j\in\omega} x_{kj}x_j) + \sum_{j\in\omega} c_j x_j \Rightarrow \text{группируем} \Rightarrow$$

$$\Rightarrow \sum_{k=1}^{m} c_{i_k} x_{k0} - \sum_{j\in\omega}(\sum_{k=1}^{m} c_{i_k} x_{kj} - c_j)x_j =$$

$$= \sum_{k=1}^{m} c_{i_k} x_{k0} - \sum_{j\in\omega} \Delta_j x_j \ .$$

Теперь предельно ясно, какую свободную переменную нужно вводить в состав базисных, чтобы увеличить значение ЦФ – ту, у которой имеется отрицательная оценка. Таким образом:

оценка небазисного вектора A_j – это взятый с обратным знаком коэффициент, с которым свободная переменная x_j входит в ЦФ, если все базисные переменные выразить через свободные.

При введении свободной переменной в состав базисных эта переменная увеличивается (от нуля). Эта переменная может быть увеличена до тех пор, пока одна из базисных переменных не примет нулевое значение. Из (1.59) следует, что это будет та переменная, для которой $x_{kj} > 0$ и x_{k0}/x_{kj} имеет наименьшее значение. Именно этот факт определяет выражение (1.33) в теореме 1.2:

$$\frac{x_{r0}}{x_{rs}} = \min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\}.$$

Контрольные вопросы и задачи к разделу 1.2

1. Сформулируйте основные положения симплекс-метода для задачи на минимум целевой функции.

2. Как соотносятся понятие «базис» и «опорное решение»? Должно ли каждому базису соответствовать опорное решение, и, наоборот, каждому ли опорному решению должен соответствовать базис?

3. К чему приведет нарушение требования $\frac{x_{r0}}{x_{rs}} = \min_{x_{ks}>0}\left\{\frac{x_{k0}}{x_{ks}}\right\}$ при определении выводимого из базиса вектора?

4. Сравните метод *М*-задачи и метод вспомогательной задачи. Можно ли однозначно определить предпочтительный метод искусственного базиса перед решением задачи?

5. В какой по счету симплекс-таблице содержится базисная матрица текущего опорного решения?

6. Решите задачу графически

$$7x_1 + 10x_2 \to \max$$
$$2x_1 + 5x_2 \le 118$$
$$5x_1 + 2x_2 \le 106$$
$$x_1 \le 16$$
$$x_2 \le 20$$
$$x_{1,2} \ge 0.$$

7. После очередной итерации при решении задачи ЛП на max ЦФ симплекс-методом получена симплекс-таблица:

Баз	$C_{БАЗ}$	A_0	1	2	0	0
			A_1	A_2	A_3	A_4
A_1	1	1	1	0	-1	1
A_2	2	1	0	1	1	-1

- построить исходную задачу;
- найти оптимальное решение исходной задачи;
- найти базисную матрицу оптимального решения исходной задачи.

8. При решении задачи ЛП на максимум ЦФ получена симплекс-таблица.

Баз	$C_{БАЗ}$	A_0	3	-1	y	-2
			A_1	A_2	A_3	A_4
A_1	3	2	1	0	2	x-2
A_2	-1	1	0	1	4	x

Приняв x и y в качестве параметров, найти области значений этих параметров, при которых:

- полученное решение – оптимальное;
- ЦФ не ограничена сверху на допустимом множестве;
- решение следует продолжить.

9. Решить задачу, используя метод вспомогательной задачи

$$4x_1 + x_2 \to \max,$$
$$-5x_1 + 3x_2 = 2,$$
$$2x_1 + 7x_2 = 1,$$
$$x_1, x_2 \geq 0.$$

10. Решить задачу, используя метод М-задачи.

$$-2x_1+4x_2 -3x_3-25x_4\to max$$
$$x_1-4x_2-14x_3 \ -7x_4 =8$$
$$x_2 +4x_3 +2x_4 =2,$$
$$3x_2-12x_3-42x_4=6$$
$$x_j \geq 0,\ j=1\div 4.$$

1.3. Теория двойственности

Двойственность – фундаментальное понятие, на основе которого в линейном программировании получены важные теоретические результаты в области анализа линейных моделей, а также разработаны эффективные вычислительные процедуры.

Установлено, что с каждой задачей линейного программирования связана некоторая другая задача, которая строится по вполне определенным правилам, причем связь эта настолько тесная, что решая одну из этих задач, можно автоматически получить решение другой задачи или убедиться в неразрешимости последней. Более того, если одна из задач имеет определенную экономическую интерпретацию, то другая задача также имеет вполне определенный содержательный смысл.

1.3.1. Модели двойственных задач в линейном программировании

Без потери общности возьмем в качестве исходной задачи задачу линейного программирования в канонической форме (эту задачу далее будем называть также прямой):

$$
\begin{array}{l}
c_1x_1 + c_2x_2 + \ldots + c_jx_j + \ldots + c_nx_n \to max \\
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n = a_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n = a_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n = a_i \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mj}x_j + \ldots + a_{mn}x_n = a_m \\
x_j \geq 0, \ (j = \overline{1,n})
\end{array}
\tag{1.60}
$$

или в матрично-векторной форме:

$$
\begin{cases}
< c, x > \to \max \\
AX = A_0 \\
x \geq 0.
\end{cases}
\tag{1.61}
$$

Задачей, <u>*двойственной*</u> к (1.60), назовем задачу:

$$
\begin{array}{l}
a_1y_1 + a_2y_2 + \ldots + a_iy_i + \ldots + a_my_m \to min \\
a_{11}y_1 + a_{21}y_2 + \ldots + a_{i1}y_i + \ldots + a_{m1}y_m \geq c_1 \\
a_{12}y_1 + a_{22}y_2 + \ldots + a_{i2}y_i + \ldots + a_{m2}y_m \geq c_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{1j}y_1 + a_{2j}y_2 + \ldots + a_{ij}y_i + \ldots + a_{mj}y_m \geq c_j \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{in}y_i + \ldots + a_{mn}y_m \geq c_n \\
y_i \text{ – не ограничены в знаке } (i = 1, 2, \ldots, m)
\end{array}
\tag{1.62}
$$

или в матрично-векторной форме:

$$
\begin{cases}
< A_0^T, y > \to \min \\
A^T Y \geq C,
\end{cases}
\tag{1.63}
$$

где

$y=(y_1, y_2, \ldots, y_m)$ – вектор-строка переменных двойственной задачи;

$Y=(y_1, y_2, \ldots, y_m)^T$ – вектор-столбец переменных двойственной задачи;

A^T – транспонированная матрица коэффициентов левой части системы ограничений прямой задачи (1.60);
$C=(c_1, c_2, \ldots, c_m)^T$ – вектор-столбец, составленный из коэффициентов целевой функции прямой задачи (1.60).

Пример 1.18

Дана задача ЛП в канонической форме:

$$\begin{array}{rcrcrcrcl} 3x_1 & - & x_2 & + & 2x_3 & + & x_4 & \to & \max \\ 2x_1 & - & x_2 & + & 4x_3 & + & x_4 & = & 10 \\ -3x_1 & + & 2x_2 & + & x_3 & - & 2x_4 & = & 8 \\ 4x_1 & - & x_2 & - & 2x_3 & & & = & 4 \end{array}$$

$x_j \geq 0, \; j = \overline{1,4}.$

Двойственная к этой задаче задача имеет вид:

$$\begin{array}{l} 10y_1 + 8y_2 + 4y_3 \to \min \\ 2y_1 - 3y_2 + 4y_3 \geq 3 \\ -y_1 + 2y_2 - y_3 \geq -1 \\ 4y_1 + y_2 - 2y_3 \geq 2 \\ y_1 - 2y_2 \geq 1 \end{array}$$

y_1, y_2 – не ограничены в знаке.

Задачи (1.60) и (1.62) – взаимно-двойственные в том смысле, что, приняв в качестве прямой задачу (1.62), можно перейти к двойственной задаче, которая с точностью до обозначения переменных будет иметь вид задачи (1.60). Проиллюстрируем этот переход, используя эквивалентные преобразования, описанные в разделе 1.2.

Шаг 1. Изменение направления оптимизации и неравенств в задаче (1.63). Эквивалентная задача:

$$\begin{cases} -< A_0^T, y > \to \max \\ \quad -A^T Y \leq -C. \end{cases}$$

Шаг 2. Введение требования неотрицательности переменных путем их замены:

$$\begin{cases} -< A_0^T, (y' - y'') > \to \max \\ \quad -A^T (Y' - Y'') \leq -C \\ \quad y' \geq 0, y'' \geq 0, \end{cases}$$

где $y' = \left(y'_1, y'_2, \ldots, y'_m\right),\ y'' = \left(y''_1, y''_2, \ldots, y''_m\right).$

Шаг 3. Приведение задачи к канонической форме путем введения дополнительных переменных:

$$\begin{cases} -< A_0^T, (y' - y'') > \to \max \\ -A^T (Y' - Y'') + EZ = -C \\ y' \ge 0, y'' \ge 0, z \ge 0, \end{cases}$$

где $Z = \left(z_1, z_2, \ldots, z_n\right)^T$, $z_j \ge 0, \left(j = 1, 2, \ldots, n\right)$.

Шаг 4. Построение двойственной задачи:

$$\begin{cases} -< c, x > \to \min \\ \quad -AX \ge -A_0 \\ \quad AX \ge A_0 \\ \quad E^T X \ge 0. \end{cases}$$

Шаг 5. Изменение направления экстремизации и замена нестрогих неравенств равенствами:

$$\begin{cases} < c, x > \to \max \\ \quad AX = A_0 \\ \quad x \ge 0. \end{cases}$$

Таким образом, получена задача, имеющая вид задачи (1.61).

Задачи (1.60) и (1.62) называются несимметричной парой двойственных задач.

Пусть теперь прямая задача имеет симметричную форму:

$$\begin{cases} < c, x > \to \max \\ \quad AX \le A_0 \\ \quad x \ge 0. \end{cases} \tag{1.64}$$

Построим двойственную задачу:

Шаг 1. Приведение задачи к канонической форме:

$$\begin{cases} < c, x > \to \max \\ AX + EZ = A_0 \\ \quad x \ge 0, z \ge 0. \end{cases}$$

Шаг 2. Построение двойственной задачи:

$$\begin{cases} < A_0^T, y > \to \min \\ A^T Y \geq C \\ EY \geq 0. \end{cases}$$

Ввиду того, что последняя группа нестрогих неравенств эквивалентна требованию неотрицательности переменных, результатом приведенных преобразований является следующая задача:

$$\begin{cases} < A_0^T, y > \to \min \\ A^T Y \geq C \\ y \geq 0. \end{cases} \qquad (1.65)$$

Следует обратить внимание на то, что в двойственной задаче сохранилось требование неотрицательности переменных.

Задачи (1.64) и (1.65) называются *симметричной парой двойственных задач*.

Можно показать, что задачи (1.64) и (1.65) являются взаимно-двойственными, т.е., если принять в качестве прямой задачу (1.65) , придать ей каноническую форму и построить двойственную к ней задачу, будет получена задача, которая с точностью до обозначения переменных будет совпадать с задачей (1.64).

Рассмотрим основные, наиболее часто используемые в теории линейного программирования, модели взаимно-двойственных задач.

Прямая задача	Двойственная задача
$< c, x > \to \max$ $AX = A_0$ $x \geq 0$	$< A_0^T, y > \to \min$ $A^T Y \geq C$
$< c, x > \to \min$ $AX = A_0$ $x \geq 0$	$< A_0^T, y > \to \max$ $A^T Y \leq C$
$< c, x > \to \max$ $AX \leq A_0$ $x \geq 0$	$< A_0^T, y > \to \min$ $A^T Y \geq C$ $y \geq 0$

В самом общем случае, когда прямая задача имеет смешанную систему ограничений, содержащую уравнения, нестрогие неравенства различной направленности и к переменным предъявлены различные требования (неограниченность в знаке, неотрицательность, неположительность), двойственная задача может быть построена непосредственно по прямой задаче. При этом можно не проводить трудоемких преобразований, связанных с приданием исходной задаче канонической формы, а использовать следующие правила, сведенные в таблицы.

Прямая задача на максимум целевой функции

Прямая задача	Двойственная задача
Целевая функция: $<c,x>\to\max$	Правая часть системы ограничений: C
Правая часть системы ограничений: A_0	Целевая функция: $< A_0^T, y >\to \min$
i -е ограничение: "="	i-я переменная: y_i не ограничена в знаке
i-е ограничение: " $\le$ "	i -я переменная: $y_i \ge 0$
i -е ограничение: " $\ge$ "	i -я переменная: $y_i \le 0$
j-я переменная: x_j не ограничена в знаке	j -е ограничение: " = "
j -я переменная: $x_j \ge 0$	j -е ограничение: " $\ge$ "
j -я переменная: $x_j \le 0$	j -е ограничение: " $\le$ "

Прямая задача на минимум целевой функции

Прямая задача	Двойственная задача
Целевая функция: $<c,x>\to\min$	Правая часть системы ограничений: С
Правая часть системы ограничений: A_0	Целевая функция: $< A_0^T , y >\to \max$
i -е ограничение: " = "	i -я переменная: y_i не ограничена в знаке
i -е ограничение: " $\le$ "	i-я переменная: $y_i \le 0$
i -е ограничение: " $\ge$ "	i -я переменная: $y_i \ge 0$
j -я переменная: x_j не ограничена в знаке	j -е ограничение: " = "
j -я переменная: $x_j \ge 0$	j -е ограничение: " $\le$ "
j -я переменная: $x_j \le 0$	j -е ограничение: " $\ge$ "

1.3.2. Связь двойственных задач

Ниже приводятся некоторые теоретические выкладки, устанавливающие связь решений прямой и двойственной задачи линейного программирования.

В основном рассматривается несимметричная пара взаимнодвойственных задач (1.60) и (1.62).

ЛЕММА 1.3. (о допустимых решениях прямой и двойственной задачи).

Пусть дана прямая задача:

$$\begin{cases} < c, x >\to \max \\ AX = A_0 \\ x \ge 0 \end{cases}$$

и двойственная к ней:

$$\begin{cases} < A_0^T, y >\to \min \\ A^T Y \ge C. \end{cases}$$

Пусть далее $\overline{x} = \left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\right)$ *и* $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$ *– любые допустимые решения прямой и двойственной задачи соответственно.*

Тогда имеет место следующее соотношение:

$$< c,\overline{x} > \ \leq \ < A_0^T, \overline{y} > \qquad (1.66)$$

(*т.е. значение целевой функции прямой задачи на любом ее допустимом решении не может превысить значение целевой функции двойственной задачи на любом допустимом решении последней*).

Доказательство. Умножим левую и правую части первого ограничения-уравнения прямой задачи на $\overline{y}_1$, второго ограничения – на $\overline{y}_2$, и т.д., после чего сложим все полученные уравнения. Результат этого сложения в форме скалярного произведения векторов будет иметь вид:

$$< (A\overline{X})^T, \overline{y} >=< A_0^T, \overline{y} >,$$

или

$$< (A^T\overline{Y})^T, \overline{x} >=< A_0^T, \overline{y} >.$$

Ввиду допустимости решения $\overline{y}$, справедливо $A_0^T\overline{Y} \geq C$, откуда:

$$< c,\overline{x} > \leq < A_0^T, \overline{y} >,$$

что и требовалось доказать.

ЛЕММА 1.4 (об оптимальных решениях прямой и двойственной задачи).

Если на некоторых допустимых решениях $\overline{x} = \left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\right)$ *прямой и* $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$ *двойственной задачи нестрогое неравенство (5) обращается в строгое равенство, то* $\overline{x} = \left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\right)$ *и* $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$ *– оптимальные решения прямой и двойственной задачи соответственно.*

Доказательство. Пусть $< c,\overline{x} > = < A_0^T, \overline{y} >$, но $\overline{x}$ – не оптимальное решение. Следовательно, существует такое решение $x^0 = \left(x_1^0, x_2^0, ..., x_n^0\right)$, что $< c, x^0 > > < c,\overline{x} >$ или $< c, x^0 > > < A_0^T, \overline{y} >$, а это противоречит утверждению леммы 1.3. Следовательно, $\overline{x} = \left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\right)$ – оптимальное решение прямой задачи.

Пусть теперь $< c,\overline{x} > = < A_0^T, \overline{y} >$, но $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$ – не оптимальное решение. Следовательно, существует такое решение

$y^0 = \left(y_1^0, y_2^0, ..., y_m^0\right)$ что: $< A_0^T, y^0 > < < A_0^T, \overline{y} >$ или $< c, \overline{x} > > < A_0^T, y^0 >$, а это противоречит утверждению леммы 1.3. Следовательно, $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$ – оптимальное решение двойственной задачи.

Утверждение доказано.

ТЕОРЕМА 1.6 (первая теорема двойственности для несимметричной пары двойственных задач).

1. Если одна из задач (1.60) или (1.62) имеет оптимальное решение, то и другая задача имеет оптимальное решение, причем, оптимальные значения целевых функций этих задач совпадают.

2. Если целевая функция одной из задач не ограничена сверху (соответственно, снизу), то другая задача не имеет допустимых решений.

Доказательство. Пусть задача (1.60) решена симплекс-методом и имеет оптимальное опорное решение $x^* = \left(x_1^*, x_2^*, ..., x_n^*\right)$, и $B = \left(A_{i_1}, A_{i_2}, ..., A_{i_m}\right)$ – базис этого решения. Тогда оценки всех векторов, вычисленные на последней итерации, неотрицательные, т.е.:

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} x_{kj} - c_j = c_{баз} X_j - c_j = c_{баз} B^{-1} A_j - c_j \geq 0.$$

Здесь используются следующие обозначения:

$c_{баз} = (c_{i_1}, c_{i_2}, \ldots, c_{i_m})$ – вектор коэффициентов целевой функции;

$X_j = \left(x_{1j}, x_{2j}, ..., x_{mj}\right)^T$ – вектор-столбец коэффициентов разложения вектора A_j по базису .

Рассмотрим выражение:

$$c_{баз} B^{-1} A_j \geq c_j, \ (j = \overline{1, n}). \tag{1.67}$$

Примем обозначение:

$$(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m) = c_{баз} B^{-1}. \tag{1.68}$$

Отсюда , учитывая (1.67), имеем систему неравенств:

$$(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m)\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \geq c_j \ , \ (j = \overline{1,n}) .$$

В матричной форме эта система имеет вид: $A^T \overline{Y} \geq C,$ т.е., вектор $\overline{y}$, построенный в соответствии с выражением (1.68), удовлетворяет всем ограничениям двойственной задачи (1.62).

Найдем значение целевой функции двойственной задачи на допустимом решении $\overline{y} = \left(\overline{y}_1, \overline{y}_2, ..., \overline{y}_m\right)$:

$$< \overline{y}, A_0^T > = < c_{баз} B^{-1}, A_0^T > = < c_{баз}, (B^{-1} A_0)^T > .$$

Учитывая, что $(B^{-1} A_0)^T = \left(x_{10}, x_{20}, ..., x_{m0}\right) = X_0^T$ – вектор оптимальных значений базисных переменных задачи (1), имеет место $< c_{баз}, X_0^T > = < c, x^* >$. Следовательно:

$$< \overline{y}, A_0^T > = < c_{баз}, X_0^T > = < c, x^* > ,$$

т.е., значение целевой функции прямой задачи на решении x^0 совпадает со значением целевой функции двойственной задачи на ее допустимом решении $\overline{y}$. Следовательно, решение $\overline{y}$, в соответствии с утверждением леммы 1.4, является оптимальным. Первая часть теоремы доказана.

Пусть теперь целевая функция прямой задачи (1.60) не ограничена сверху на допустимом множестве. Предположим, от противного, что двойственная задача имеет допустимые решения и $\overline{y}$ – одно из таких решений. Пусть целевая функция двойственной задачи на решении $\overline{y}$ принимает некоторое значение $Q = < A_0^T, \overline{y} >$.

Ввиду того, что целевая функция прямой задачи (1.60) не ограничена сверху на допустимом множестве, всегда найдется такое допустимое решение $x' = (x_1', x_2', ..., x_n')$, на котором имеет место $< c, x' > > Q$, что противоречит утверждению леммы 1.3 о допустимых решениях прямой и двойственной задачи. Вторая часть теоремы доказана.

Ввиду того, что задачи (1.60) и (1.62) взаимно-двойственны, в качестве прямой задачи можно взять задачу (1.62) и повторить приведенное выше доказательство. Теорема доказана[3].

ТЕОРЕМА 1.7 (вторая теорема двойственности для несимметричной пары двойственных задач).

Пусть дана задача линейного программирования:

$$c_1x_1 + c_2x_2 + ... + c_nx_n \to \max$$

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = a_i, \ (i = \overline{1,m})$$

$$x_j \geq 0, (j = \overline{1,n})$$

и известно ее допустимое решение $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$.

Пусть далее известно допустимое решение $\overline{y} = (\overline{y}_1, \overline{y}_2, ..., \overline{y}_m)$ *двойственной задачи:*

$$a_1y_1 + a_2y_2 + ... + a_my_m \to \min$$

$$a_{1j}y_1 + a_{2j}y_2 + ... + a_{mj}y_m \geq c_j, \ (j = \overline{1,n}).$$

Тогда решения $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ *и* $\overline{y} = (\overline{y}_1, \overline{y}_2, ..., \overline{y}_m)$ *будут оптимльными решениями соответственно прямой и двойственной задачи в том и только том случае, если имеет место*

$$\overline{x}_j(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m - c_j) = 0, \ (j = \overline{1,n}). \qquad (1.69)$$

Достаточность. Очевидно, что при выполнении (1.69) будут выполняться и все равенства, полученные из этих условий:

$$\overline{x}_j(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m) = c_j\overline{x}_j, \ (j = \overline{1,n}).$$

Просуммируем эти равенства по всем *j=1,2,...,n* и перегруппируем слагаемые в левой части полученного выражения:

$$\sum_{i=1}^{m}(a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + ... + a_{in}\overline{x}_n)\overline{y}_i = \sum_{j=1}^{n} c_j\overline{x}_j .$$

Но $\overline{x}$ – допустимое решение, значит:

$$a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + ... + a_{in}\overline{x}_n = a_i \ \ (i = \overline{1,m}).$$

[3] Эта теорема справедлива и для симметричной пары двойственных задач (1.61) и (1.62) .

То есть

$$\sum_{i=1}^{m} a_i \overline{y}_i = \sum_{j=1}^{n} c_j \overline{x}_j \ .$$

Следовательно, согласно лемме 1.4 , $\overline{x}$ и $\overline{y}$ – оптимальные решения прямой и двойственной задачи соответственно (доказана достаточность).

Необходимость. Ввиду оптимальности решений $\overline{x}$ и $\overline{y}$ имеет место

$$a_1\overline{y}_1 + a_2\overline{y}_2 + \ldots + a_m\overline{y}_m = c_1\overline{x}_1 + c_2\overline{x}_2 + \ldots + c_n\overline{x}_n \ . \qquad (1.70)$$

Но $\overline{x}$ – допустимое решение, следовательно,

$$a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + \ldots + a_{in}\overline{x}_n = a_i \ , \ \ (i = \overline{1,m}\ ) \ .$$

Подставим эти выражения в (1.70) :

$$\sum_{i=1}^{m} (a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + \ldots + a_{in}\overline{x}_n)\overline{y}_i = \sum_{j=1}^{n} c_j \overline{x}_j \ .$$

После простых преобразований приходим к следующему выражению:

$$\sum_{j=1}^{n} \overline{x}_j (a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + \ldots + a_{mj}\overline{y}_m - c_j) \ = 0 \ .$$

Но $\overline{y} = (\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m)$ – допустимое решение, следовательно, ни одно из подскобочных выражений не может быть строго отрицательным.

$\overline{x}$ – также допустимое решение, т.е. $x_j \geq 0, \ (j = \overline{1,n}\ )$.

Таким образом, для любого $j = \overline{1,n}$ должно иметь место

$$\overline{x}_j (a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + \ldots + a_{mj}\overline{y}_m - c_j) = 0 \, ,$$

что и требовалось доказать.

Как обе леммы, так и обе теоремы были сформулированы для несиммтричной пары взаимно-двойственных задач. Для симметричной пары взаимно-двойственных задач леммы 1.3 и 1.4 и обе теоремы двойственности также выполняются, однако вторая теорема двойственности приобретает “симметричную” формулировку.

ТЕОРЕМА 1.8 (вторая теорема двойственности для симметричной пары двойственных задач).

Пусть дана задача линейного программирования:

$$c_1x_1 \quad + c_2x_2 \quad + \ldots + c_nx_n \to \max$$

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \le a_i,\ (i = \overline{1,m}\ )$$

$$x_j \ge 0, (j = \overline{1,n}\ )$$

и известно ее допустимое решение $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$.

Пусть далее известно допустимое решение $\overline{y} = (\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m)$ *двойственной задачи:*

$$a_1y_1 \quad + a_2y_2 \quad + \ldots + a_my_m \to \min$$

$$a_{1j}y_1 + a_{2j}y_2 + \ldots + a_{mj}y_m \ge c_j,\ (j = \overline{1,n}\ )$$

$$y_i \ge 0, \quad i = \overline{1,m}.$$

Тогда решения $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ *и* $\overline{y} = (\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m)$ *будут оптимльными решениями соответственно прямой и двойственной задачи в том и только том случае, если имеет место*

$$\begin{aligned} &\overline{x}_j(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + \ldots + a_{mj}\overline{y}_m - c_j) = 0,\ (j = \overline{1,n}\ ) \\ &\overline{y}_i(a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + \ldots + a_{in}\overline{x}_n - a_i) = 0,\ (i = \overline{1,m}\ ). \end{aligned} \tag{1.71}$$

Оставим эту теорему без доказательства. Общая схема доказательства полностью аналогична приведенной в теореме 1.7. Сформулированные в этой теореме условия (1.69) и (1.71) носят название <u>*условий дополняющей нежесткости*</u>.

Условия дополняющей нежесткости имеют четкую экономическую интерпретацияю, которая будет рассмотрена ниже.

1.3.3. Получение решения двойственной задачи по результатам решения прямой задачи

Значение первой теоремы двойственности состоит, прежде всего, в том, что решая одну из задач, можно получить решение другой задачи или убедиться в ее неразрешимости. Это обстоятельство иногда позволяет существенно облегчить решение задачи, так как предоставляется возможность выбора между прямой и двойственной задачей в пользу наименее трудоемкой.

При доказательстве первой теоремы двойственности было выведено важное соотношение:

$$y^* = c_{баз}B^{-1}, \tag{1.72}$$

где $B=\left(A_{i_1},A_{i_2},...,A_{i_m}\right)$ – базисная матрица оптимального решения прямой задачи;

$c_{баз}=(c_{i_1},c_{i_2},...,c_{i_m})$ – вектор коэффициентов целевой функции при базисных переменных;

$y^*=\left(y^*_1,y^*_2,...,y^*_m\right)$ – оптимальное решение двойственной задачи.

Как на практике используется это соотношение? Прежде всего, необходимо иметь обратную матрицу, вопрос вычисления которой затрагивался ранее.

Пусть $B^{-1}=\begin{pmatrix} \beta_{11}\cdots & \beta_{1j}\cdots & \beta_{1m} \\ \beta_{21}\cdots & \beta_{2j}\cdots & \beta_{2m} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \beta_{m1}\cdots & \beta_{mj}\cdots & \beta_{mm} \end{pmatrix}$ – матрица, обратная к базисной матрице оптимального решения прямой задачи.

Рассмотрим единичную матрицу

$$E=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}=\left(E_1,E_2,...,E_m\right),$$

где $E_j=\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$ – единичный вектор с единицей в j-й позиции ($j=1,2,...,m$).

Примем обозначение

$$X_j^e = \begin{pmatrix} e_{1j} \\ e_{2j} \\ \vdots \\ e_{mj} \end{pmatrix}$$ – матрица-столбец, составленная из коэффициентов разложения единичного вектора E_j по базису $B = \left(A_{i_1}, A_{i_2}, ..., A_{i_m}\right)$, т.е.:

$$E_j = e_{1j}A_{i_1} + e_{2j}A_{i_2} + ... + e_{mj}A_{i_m} = BX_j^e. \qquad (1.73)$$

Решаем матричное уравнение (1.7):

$$X_j^e = B^{-1}E_j = \begin{pmatrix} \beta_{11}...\beta_{1j}...\beta_{1m} \\ \beta_{21}...\beta_{2j}...\beta_{2m} \\ \dots\dots\dots\dots\dots\dots\dots \\ \beta_{m1}...\beta_{mj}...\beta_{mm} \end{pmatrix} \times \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \beta_{1j} \\ \beta_{2j} \\ \vdots \\ \beta_{mj} \end{pmatrix}.$$

Таким образом, коэффициенты разложения единичного вектора с единицей в j-й позиции совпадают с j-м столбцом обратной матрицы.

Ввиду того, что решение задачи линейного программирования симплекс-методом всегда начинается с единичной базисной матрицы, такая матрица имеется в исходной симплекс-таблице. Следовательно, в последней симплекс-таблице есть обратная матрица, и оптимальное решение двойственной задачи можно найти по формуле (1.72).

<u>Пример 1.19</u>

Пусть дана задача линейного программирования:

$$8x_1 + 2x_2 + 6x_3 \to \min$$

$$x_1 + 2x_2 + x_3 \ge 5$$

$$2x_1 - 4x_2 - 2x_3 \ge 2$$

$$x_1 \ge 4$$

$$x_2 \ge 5$$

$$x_3 \ge 2$$

x_1, x_2 не ограничены в знаке.

Для того, чтобы решить эту задачу симплекс-методом, необходимо, во-первых, придать ей каноническую форму:

$$-8x_1' + 8x_1'' - 2x_2' + 2x_2'' - 6x_3' + 6x_3'' \to \max$$

$$\begin{aligned}
x_1' - x_1'' + 2x_2' - 2x_2'' + x_3' - x_3'' - x_4 &= 5 \\
2x_1' - 2x_1'' - 4x_2' + 4x_2'' - 2x_3' + 2x_3'' - x_5 &= 2 \\
x_1' - x_1'' - x_6 &= 4 \\
x_2' - x_2'' - x_7 &= 5 \\
x_3' - x_3'' - x_8 &= 2
\end{aligned}$$

$$x_j \ge 0,\ j = \overline{1,8},$$

после чего (ввиду отсутствия в системе ограничений полного набора единичных векторов) применить метод искусственного базиса.

Для этого в систему ограничений нужно ввести пять искусственных переменных ─ по одной на каждое ограничение-уравнение. Полученная задача будет иметь размерность *5×16*, где *5* ─ это количество ограничений, а *16* ─ количество переменных.

Трудоемкость решения задачи можно существенно уменьшить, если воспользоваться результатами этого раздела. Действительно, построим задачу, двойственную к исходной.

$$5y_1 + 2y_2 + 4y_3 + 5y_4 + 2y_5 \to \max$$

$$\begin{aligned}
y_1 + 2y_2 + y_3 &= 8 \\
2y_1 - 4y_2 + y_4 &= 2 \\
y_1 - 2y_2 + y_5 &= 6
\end{aligned}$$

$$y_j \ge 0,\ j = \overline{1,5}.$$

Полученная задача имеет размерность *3 × 5*, и ее система ограничений содержит полный набор единичных векторов $A_3, A_4, A_5,$ которые можно принять в качестве базиса очевидного опорного решения $y = (0,0,8,2,6)$.

Решение двойственной задачи симплекс-методом приведено в двух таблицах:

Баз	$C_{баз}$	A_0	5	2	4	5	2
			A_1	A_2	A_3	A_4	A_5
A_3	4	8	1	2	1	0	0
A_4	5	2	2	-4	0	1	0
A_5	2	6	1	-2	0	0	1
Табл.1		54	11	-18	0	0	0
A_2	2	4	1/2	1	1/2	0	0
A_4	5	18	4	0	2	1	0
A_5	2	14	2	0	1	0	1
Табл.2		126	20	0	9	0	0

Базисная матрица оптимального решения и обратная к ней матрица определяются из первой и второй таблицы, соответственно:

$$B = (A_2, A_4, A_5) = \begin{pmatrix} 2 & 0 & 0 \\ -4 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1/2 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

откуда вычисляется вектор переменных оптимального решения исходной задачи:

$$\overline{X} = \left(\overline{x}_1, \overline{x}_2, \overline{x}_3\right) = c_{баз} B^{-1} = (2,5,2) \begin{pmatrix} 1/2 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = (13,5,2).$$

Оптимальное значение целевой функции исходной задачи совпадает (согласно утверждению леммы 1.4) со значением целевой функции двойственной задачи, оно берется из второй таблицы:

$$Z_{опт} = 126.$$

Для нахождения оптимального решения двойственной задачи можно избежать перемножения вектора $c_{баз}$ на обратную матрицу B^{-1}.

Действительно, $\Delta_j = c_{баз} B^{-1} A_j - c_j$, т.е., если B^{-1} – матрица, обратная базисной матрице оптимального решения прямой задачи, то согласно (1.72):

$$\Delta_j = y^* A_j - c_j, \text{ где } y^* = \left(y_1^*, y_2^*, ..., y_m^*\right).$$

Допустим A_j – единичный вектор с единицей в j-й позиции. Тогда:

$$\Delta_j = \left(y_1^*, y_2^*, \dots, y_m^*\right)\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} - c_j ,$$

откуда:

$$\Delta_j = y_j^* - c_j \quad \text{или} \quad y_j^* = \Delta_j + c_j .$$

Таким образом, имеем следующее правило:

Для того чтобы получить оптимальное значение j-й двойственной переменной (j=1,2,...,m), достаточно взять оптимальную симплекс-таблицу и к оценке исходного единичного вектора с единицей в j-й позиции прибавить коэффициент, с которым переменная при этом единичном векторе входит в целевую функцию.

Действительно (см. оптимальную симплекс-таблицу примера 1.19),

$$\left.\begin{aligned} x_1^* &= \Delta_3 + c_3 = 9 + 4 = 13 \\ x_2^* &= \Delta_4 + c_4 = 0 + 5 = 5 \\ x_3^* &= \Delta_5 + c_5 = 0 + 2 = 2 \end{aligned}\right\}, \text{ т.е. } x^* = \left(13, 5, 2\right) - \text{результат тот же.}$$

Итак, первый и важнейший результат теории двойственности – это возможность резкого сокращения трудоемкости решения задач ЛП.

Это можно проиллюстрировать следующим образом (рис.1.12).

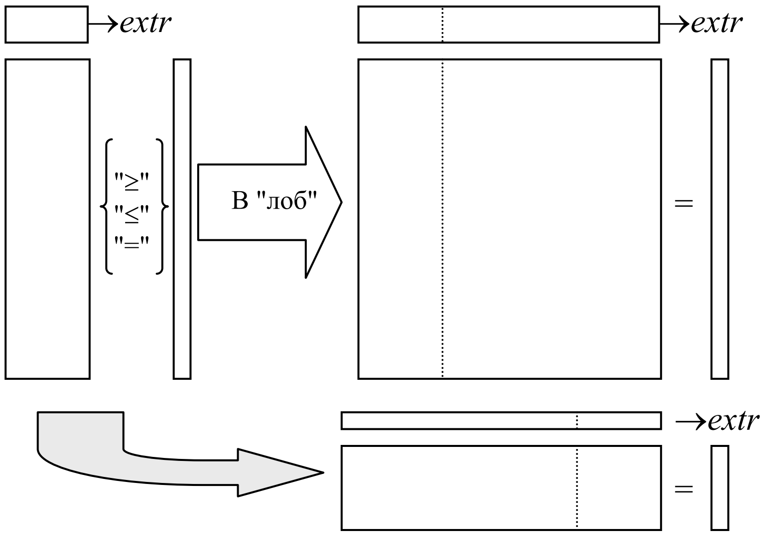


Рис.1.12. Переход к двойственной задаче

Таким образом, решается более простая двойственная задача, и по ее решению находится решение исходной задачи.

1.3.4. Экономическая интерпретация двойственности

Рассмотрим симметричную пару двойственных задач.

Прямая задача

$c_1x_1+c_2x_2+\ldots+c_jx_j+\ldots+c_nx_n\to max$

$a_{11}x_1+a_{12}x_2+\ldots+a_{1j}x_j+\ldots+a_{1n}x_n\le a_1$

$a_{21}x_1+a_{22}x_2+\ldots+a_{2j}x_j+\ldots+a_{2n}x_n\le a_2$

....

$a_{i1}x_1+a_{i2}x_2+\ldots+a_{ij}x_j+\ldots+a_{in}x_n\le a_i$

....

$a_{m1}x_1+a_{m2}x_2+\ldots+a_{mj}x_j+\ldots+a_{mn}x_n\le a_m$

$x_j\ge 0,\ (j=\overline{1,n})$

Двойственная задача

$a_1y_1+a_2y_2+\ldots+a_iy_i+\ldots+a_my_m\to min$

$a_{11}y_1+a_{21}y_2+\ldots+a_{i1}y_i+\ldots+a_{m1}y_m\ge c_1$

$a_{12}y_1+a_{22}y_2+\ldots+a_{i2}y_i+\ldots+a_{m2}y_m\ge c_2$

....

$a_{1j}y_1+a_{2j}y_2+\ldots+a_{ij}y_i+\ldots+a_{mj}y_m\ge c_j$

....

$a_{1n}y_1+a_{2n}y_2+\ldots+a_{in}y_i+\ldots+a_{mn}y_m\ge c_n$

$y_i\ge 0,\ (i=\overline{1,m})$

Прямая задача, с экономической точки зрения, может быть интерпретирована как общая задача производственного планирования с m производственными факторами и n видами выпускаемой продукции (задача Канторовича): необходимо определить план выпуска продукции, приносящий максимальный доход. При этом суммарные затраты каждого фактора не могут превышать его запас, имеющийся в наличии к началу производства. При этом коэффициенты a_{ij} интерпретируются следующим образом:

- Если a_{ij} >0, то в процессе производства единицы j-го вида продукции расходуется a_{ij} единиц i-го производственного фактора;
- Если a_{ij} <0, то в процессе производства единицы j-го вида продукции вырабатывается $\left|a_{ij}\right|$ единиц i-го производственного фактора;
- Если a_{ij} =0, то в процессе производства j-го вида продукции i-й производственный фактор не участвует.

Знак коэффициентов правой части системы ограничений может быть интерпретирован следующим образом:

- Если a_i >0, то к началу производства предприятие обладает запасом i-го производственного фактора в обеме a_i единиц;
- Если a_i <0, то к началу производства i-й фактор не только отсутствует, но и должен быть выработан в количестве, не меньшем $\left|a_i\right|$ в процессе производства;
- Если a_i =0, то i-й фактор к началу производства отсутствует, и без выработки затрачен быть не может (такой фактор можно интерпретировать как полуфабрикат).

Наконец, коэффициенты целевой функции имеют следующую интерпретацию:

- Если c_j >0, то производство единицы j-го вида продукции приносит доход в объеме c_j рублей;
- Если c_j <0, то производство единицы j-го вида продукции приносит убыток в объеме $\left|c_j\right|$ рублей;
- Если c_j =0, то производство j-го вида продукции не приносит ни прибыли, ни убытка.

Рассмотрим теперь двойственную задачу. Согласно лемме 1.3, значение целевой функции прямой задачи на любом допустимом её решении не может превышать значения целевой функции любого допустимого решения двойственной задачи:

$$\sum_{j=1}^{n} c_j \overline{x}_j \leq \sum_{i=1}^{m} a_i \overline{y}_i$$

Согласно лемме 1.4, указанное неравенство превращается в строгое равенство для оптимальных решений прямой и двойственных задач. Поскольку целевая функция прямой задачи в нашей интерпретации представляет собой суммарный доход от выпуска продукции, а размерность целевой функции двойственной задачи должна совпадать с размерностью целевой функции прямой задачи, размерность отдельной переменной двойственной задачи равна $\left[y_i\right] = руб / ед$, т.е. i-я переменная двойственной задачи представляет собой некую денежную оценку i-го производственного фактора. Очевидно, что эта оценка не имеет никакого отношения к рыночной цене фактора, поскольку подобных данных в условиях задачи нет. Скорее, речь идет о некоторой «внутренней» оценке производственного фактора, определяемой в процессе решения задачи. В таком случае, целевая функция двойственной задачи представляет собой суммарную оценку запасов производственных факторов, имеющихся к началу производства (здесь для упрощения делается предположение, что все a_i>0).

С коммерческой точки зрения, требование минимизации оценки собственных запасов может показаться несколько некорректным. Здесь следует подчеркнуть тот факт, что сама по себе поставленная задача не решает вопроса оптимального определения торговой наценки на себестоимость выпускаемой продукции, а определяет оптимальный план выпуска продукции, исходя из технологических особенностей. Само же производство представляет собой процесс преобразования факторов, имеющихся в наличии, в готовую продукцию, и экономический смысл леммы 1.3 и леммы 1.4 заключается в том, что при неоптимальном плане преобразования денежная оценка факторов превысит конечную стоимость произведенной продукции (что есть неоптимально с финансовой точки зрения). Именно в свете такой интерпретации и следует понимать направленность целевой функции двойственной задачи: в процессе оптимального производства не появляется какая-то посторонняя денежная масса, а лишь факторы преобразуются в продукцию таким образом, что их денежное выражение совпадает.

Рассмотрим ограничения двойственной задачи. Учитывая интерпретацию переменных двойственной задачи, левая часть её j-го ограничения представляет собой суммарные затраты на производство единицы продукции j-го вида в денежном выражении. Правая же часть – конечная стоимость этой единицы. Таким образом, с экономической точки зрения, опять ставится нерациональное требование: суммарные затраты на производство продукции не должны быть меньше стоимости этой продукции. Подобные ограничения опять-таки проистекают из природы двойственной задачи, цель которой – определить крайнюю оценку продукции, ниже которой оптимальное производство будет происходить в убыток. Для того чтобы понять это, рассмотрим условия дополняющей нежесткости из второй теоремы двойственности. Согласно этой теореме для симметричной пары задач, выражения

$$\overline{x}_j(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m - c_j) = 0,\ (j = \overline{1,n}\ )$$

$$\overline{y}_i(a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + ... + a_{in}\overline{x}_n - a_i) = 0,\ (i = \overline{1,m}\ )$$

будут справедливы только для оптимальных решений прямой и двойственных задач. Что мы видим в первом выражении? Если $(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m - c_j) > 0$, т.е. в оптимальном решении оценка затрат факторов превышает стоимость выпускаемой продукции, то $\overline{x}_j = 0$, т.е. такая продукция в план производства включаться не будет. Наоборот, если $\overline{x}_j > 0$, т.е. продукция j-го вида включена в оптимальный план производства, то $(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m - c_j) = 0$, т.е. оценка затрат факторов на её производство совпадает со стоимостью единицы j-го вида продукции, что полностью согласуется с указанной выше трактовкой производства как процесса преобразования факторов в продукцию без изменения суммарной денежной стоимости.

Рассмотрим второй блок условий дополняющей нежесткости. Если $(a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + ... + a_{in}\overline{x}_n - a_i) > 0$, т.е. по окончании производства остается неиспользованным некоторое количество i-го производственного фактора, то его оценка $\overline{y}_i = 0$. И наоборот, если оценка i-го фактора $\overline{y}_i > 0$, то фактор в процессе производства расходуется полностью, т.е. $(a_{i1}\overline{x}_1 + a_{i2}\overline{x}_2 + ... + a_{in}\overline{x}_n - a_i) = 0$. Эти выражения означают, что денежная ценность факторов в процессе производства полностью превращается в денежную ценность продукта, а те факторы, которые не полностью используются в процессе производства, не имеют денежной ценности.

На основании всего вышесказанного можно сформулировать окончательную интерпретацию двойственной задачи. Необходимо определить неотрицательные удельные оценки всех производственных факторов, измеряемые в тех же денежных единицах, что и доход от производимой продукции, чтобы соблюдались следующие условия:

1. Суммарная оценка всех производственных факторов, участвующих в производстве единицы продукции *j*-го вида, не должна быть меньше выходной стоимости этой продукции.
2. Из всех допустимых вариантов таких оценок должен быть выбран тот, для которого суммарная оценка всех имеющихся к началу производства запасов производственных факторов была бы минимальной.

1.3.5. Основные понятия двойственного симплекс-метода

Симплекс-метод, описанный в разделе 1.2, заключается в последовательном переходе от одного допустимого решения задачи линейного программирования к другому – лучшему, которое доставляет целевой функции большее значение. При этом используются правила, которые гарантируют получение оптимального решения или установление неразрешимости задачи через конечное количество шагов.

Пусть теперь дана каноническая задача линейного программирования:

$$\langle c,x\rangle \to \max, \tag{1.74}$$

$$AX=A_0 \tag{1.75}$$

$$x\geq 0. \tag{1.76}$$

Пусть, далее, $\overline{\overline{x}} = (\overline{\overline{x}}_1, \overline{\overline{x}}_2, ..., \overline{\overline{x}}_n)$ – одно из решений системы уравнений (1.75). При этом условия (1.76) могут не выполняться.

Предположим, также, что ненулевым компонентам этого решения соответствует линейно-независимые векторы, которые составляют базис системы A – квадратную матрицу

$$\overline{\overline{B}} = (A_{i_1}, A_{i_2}, ..., A_{i_m}).$$

Найдем коэффициенты разложения всех небазисных векторов системы A по данному базису, используя известное выражение:

$$X_j = \overline{\overline{B}}^{-1} A_j, (j = 1,2,...,n; j \neq i_k, k = 1,2,...,m).$$

Теперь вычислим оценки всех небазисных векторов:

$$\Delta_j = c_{баз} \overline{\overline{B}}^{-1} A_j - c_j, \ (j = 1,2,...,n; \ j \neq i_k, \ k = 1,2,...,m). \qquad (1.77)$$

Предположим, что все оценки (1.77) неотрицательны.

Вектор $\overline{\overline{x}} = (\overline{\overline{x}}_1, \overline{\overline{x}}_2, ..., \overline{\overline{x}}_n)$, удовлетворяющий системе ограничений (1.75), ненулевым координатам которого соответствует базис матрицы, построенной из коэффициентов левой части системы ограничений (1.75), называется <u>*псевдопланом*</u> задачи (1.74)-(1.76), если оценки всех векторов неотрицательны.

При доказательстве первой теоремы двойственности рассматривался случай неотрицательности оценок свободных векторов.

В частности, было доказано, что, если

$\Delta_j = c_{баз} \overline{\overline{B}}^{-1} A_j - c_j \geq 0$ или $c_{баз} \overline{\overline{B}}^{-1} A_j \geq c_j$, то $\overline{\overline{y}} = c_{баз} \overline{\overline{B}}^{-1}$ – допустимое решение двойственной задачи. Однако об оптимальности этого решения говорить не приходится, так как $\overline{\overline{x}} = (\overline{\overline{x}}_1, \overline{\overline{x}}_2, ..., \overline{\overline{x}}_n)$ может не удовлетворять требованию (1.76).

В этой связи псевдоплан иногда называют <u>*почти допустимым опорным решением*</u>**.**

Очевидно, что если псевдоплан удовлетворяет требованию неотрицательности переменных (1.76), то он является оптимальным решением задачи (1.74)-(1.76), а $\overline{\overline{y}} = c_{баз} \overline{\overline{B}}^{-1}$ – оптимальным решением двойственной задачи.

Каково значение целевой функции двойственной задачи на ее допустимом решении $\overline{\overline{y}} = c_{баз} \overline{\overline{B}}^{-1}$?

$$\overline{\overline{Z}}_{дв} = < A_0^T, c_{баз} \overline{\overline{B}}^{-1} > = < c_{баз}, (\overline{\overline{B}}^{-1} A_0)^T > =$$

$$= < (c_{i_1}, c_{i_2}, ..., c_{i_m}), (x_{10}, x_{20}, ..., x_{m0}) > = < c, \overline{\overline{x}} >,$$

т.е. значение целевой функции двойственной задачи на решении, соответствующем псевдоплану, совпадает со значением целевой функции прямой задачи на этом псевдоплане.

Таким образом, переход от одного псевдоплана к другому соответствует переходу от одного допустимого решения двойственной задачи к другому.

Идея двойственного симплекс-метода заключается в последовательных переходах между псевдопланами в сторону уменьшения значения целевой функции двойственной задачи.

Схема этого процесса можно приведена на рис.1.13.

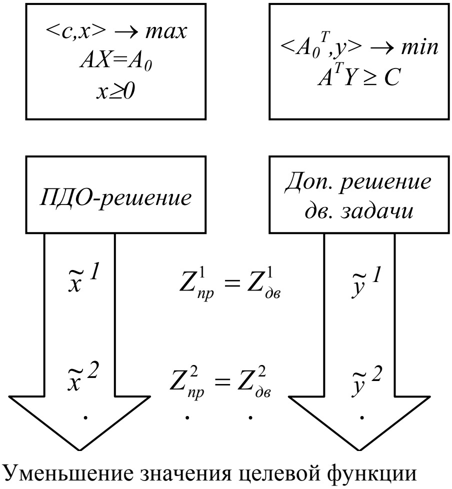


Рис.1.13. Общая схема двойственного симплекс-метода

Этот процесс продолжается до тех пор, пока координаты очередного псевдоплана станут неотрицательными или сработает признак неразрешимости задачи (будет установлен факт неограниченности снизу целевой функции двойственной задачи). В последнем случае, согласно утверждению первой теоремы двойственности, будет установлен факт отсутствия допустимых решений исходной задачи.

1.3.6. Обоснование двойственного симплекс-метода

Пусть базис некоторого псевдоплана составляют векторы

$$A_{i_1}, A_{i_2}, \dots, A_{i_{r-1}}, A_{i_r}, A_{i_{r+1}}, \dots, A_{i_m}. \qquad (1.78)$$

Заменим в этом базисе вектор A_{i_r} небазисным вектором A_s.

Согласно лемме 1.1, полученная система векторов

$$A_{i_1}, A_{i_2}, \dots, A_{i_{r-1}}, A_s, A_{i_{r+1}}, \dots, A_{i_m} \qquad (1.79)$$

будет базисом системы $A=(A_1,A_2,\dots,A_n)$, если $x_{rs} \neq 0$ – r-я координата в разложении вектора A_s по базису (1.78).

ТЕОРЕМА 1.9 (о переходе к новому псевдоплану).

Если номера выводимого из базиса (1.78) *вектора* A_{i_r} *и вводимого в базис вектора* A_s *таковы, что* $x_{rs}<0$, *и выполняются соотношения*

$$\left|\frac{\Delta_s}{x_{rs}}\right| < \left|\frac{\Delta_j}{x_{rj}}\right| \quad \text{для всех} \quad x_{rj} < 0 \quad \left(j \in \{1,2,...,n\}\right), \tag{1.80}$$

то новому базису (1.79) *будет соответствовать псевдоплан.*

Доказательство. Для того чтобы решение системы уравнений (1.75), соответствующее новому базису (1.79), являлось псевдопланом, необходимо, чтобы оценки всех небазисных векторов были неотрицательны.

Ранее было выведено выражение, связывающее оценки векторов в соседних базисах:

$$\Delta'_j = \Delta_j - \frac{x_{rj}}{x_{rs}}\Delta_s. \tag{1.81}$$

На основе этого выражения ответим на вопрос: при каких условиях новые оценки векторов будут неотрицательными?

Ввиду того, что "старому" базису (1.78) соответствует псевдоплан, имеет место $\Delta_s \geq 0$.

Рассмотрим новую оценку вектора A_{i_r}:

$$\Delta'_{i_r} = \Delta_{i_r} - \frac{x_{ri_r}}{x_{rs}}\Delta_s.$$

Здесь

$\Delta_s \geq 0$, так как старому базису (1.78) соответствует псевдоплан;

$\Delta_{i_r} = 0$, так как это оценка базисного вектора;

$x_{ri_r} = 1$, так как вектор A_{i_r} входит в старый базис.

Таким образом, требование неотрицательности оценки вектора A_{i_r} в новом базисе (1.79) эквивалентно выполнению следующего соотношения:

$$-\frac{1}{x_{rs}}\Delta_s \geq 0.$$

Очевидно, что это условие будет выполняться только в том случае, когда имеет место $x_{rs}<0$.

Условие неотрицательности оценок всех остальных векторов в новом базисе (1.76), учитывая (1.78), можно записать следующим образом:

$$\Delta_j > \frac{x_{rj}}{x_{rs}} \Delta_s. \tag{1.82}$$

Это условие автоматически выполняется для всех $x_{rj} \geq 0$, так как $\Delta_j \geq 0$ (старая оценка вектора A_j), $\Delta_s \geq 0$ и $x_{rs}<0$.

В том же случае, когда $x_{rj} < 0$, условие (1.82) выполняется только тогда, когда имеет место (1.80). Теорема доказана.

Определим условия, обеспечивающие новому псевдоплану меньшее значение целевой функции.

ТЕОРЕМА 1.10 (о переходе к лучшему псевдоплану).

Если номера выводимого из базиса (1.78) *вектора* A_{i_r} *и вводимого в базис вектора* A_s *таковы, что* $x_{rs}<0$, *и выполняются соотношения* (1.82)*, то новому псевдоплану будет соответствовать лучшее (меньшее) значение целевой функции, если будет иметь место* $x_{r0}<0$.

Доказательство. Используем известное выражение, связывающее значения целевой функции в двух последовательных базисах:

$$Z_{нов} = Z_{ст} - \frac{x_{r0}}{x_{rs}} \Delta_s. \tag{1.83}$$

Отсюда видно, что при $\Delta_s > 0$, учитывая, что обязательно должно выполняться $x_{rs}<0$, требуемое соотношение $Z_{нов} < Z_{ст}$ будет выполняться только в том случае, если $x_{r0}<0$, что и требовалось доказать.

Наконец, последний случай оформим в виде следующей теоремы.

ТЕОРЕМА 1.11 (о недопустимости задачи линейного программирования в двойственном симплекс-методе).

Пусть среди коэффициентов разложения вектора A_0 *по базису некоторого псевдоплана имеется отрицательный коэффици-*

ент x_{k0}*, но все* $x_{kj} \geq 0$*,* $k \in \{1,2,...,m\}$*,* $j = 1,2,...,n$*. Тогда задача не имеет допустимых решений.*

<u>Доказательство</u>. Рассмотрим *k*-ю строку симплекс-таблицы. В этой строке, фактически, записано следующее уравнение, которому должны удовлетворять переменные задачи:

$$x_{k0}=x_{k1}x_1+ x_{k2}x_2+\dots x_{kn}x_n. \qquad (1.84)$$

Предположим, от противного, что задача имеет допустимое решение $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$. Подставим координаты этого решения в уравнение (1.84):

$$x_{k0} = x_{k1}\bar{x}_1 + x_{k2}\bar{x}_2 + ... + x_{kn}\bar{x}_n. \qquad (1.85)$$

Ввиду того, что все $\bar{x}_j \geq 0$, $(j = \overline{1,n})$, т.к. решение $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ – допустимое и все $x_{kj} \geq 0$ по условию теоремы, в правой части (1.85) записано неотрицательное число. В левой же части записано число отрицательное, так как по условию теоремы $x_{k0} < 0$. Полученное противоречие говорит о неправомерности предположения о том, что задача имеет допустимое решение. Теорема доказана.

1.3.7. Алгоритм двойственного симплекс-метода

Пусть дана задача линейного программирования :

$$\begin{cases} < c, x > \to \max \\ AX = A_0 \\ x \geq 0, \end{cases}$$

имеющая псевдоплан $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$, и $\tilde{B} = (A_{i_1}, A_{i_2}, ..., A_{i_m})$ - базис этого псевдоплана.

Алгоритм двойственного симплекс-метода рассмотрим по шагам.

<u>Шаг 0.</u> Как и в симплекс-методе, найдем коэффициенты разложения всех (небазисных) векторов по базису $\tilde{B}$, т.е. все элементы симплекс-таблицы x_{ij} $(i = \overline{1,m};\ j = \overline{0,n})$. Запишем эти коэффициенты в первую симплекс-таблицу, после чего вычислим оценки Δ_j $(j=1,2,...,n)$ и значение целевой функции Z_0.

<u>Шаг 1</u>. Если все $x_{k0}=\tilde{x}_{i_k}\ge 0,(k=\overline{1,m})$, конец, получено оптимальное решение задачи. В противном случае выполняется следующий шаг.

<u>Шаг 2</u>. Поочередно просматриваем все элементы x_{kj}, для которых $x_{k0}<0$. Если обнаруживается, что для некоторого $x_{k0}<0$ все $x_{kj}\ge 0,(j=\overline{1,n})$ – конец, задача не имеет допустимых решений. В противном случае выполняется следующий шаг.

<u>Шаг 3.</u> Выбирается один из коэффициентов $x_{r0}<0$ $(r\in\{1,2,...,m\})$, желательно, максимальный по абсолютной величине (это, как правило, приводит к ускорению решения задачи). Далее для всех $x_{rj}<0,$ $(j\in\{1,2,...,n\})$ вычисляются отношения Δ_j/x_{rj}. Из этих отношений выбирается минимальное по абсолютной величине. Пусть это будет Δ_s/x_{rs}, что соответствует вектору A_S.

<u>Шаг 4.</u> Переходим к новому псевдоплану с новым базисом: базисный вектор A_{i_r} заменяется свободным вектором A_S. С использованием основных формул пересчитываются коэффициенты разложения всех векторов, вычисляются новые оценки свободных векторов и новое значение целевой функции. Далее выполняется шаг 1.

Работу по приведенному алгоритму рассмотрим на примере.

<u>Пример 1.20</u>

Дана задача линейного программирования в канонической форме:

$$2x_1+8x_2+x_3+2x_4-2x_5\to max$$

$$\begin{aligned}-2x_1+4x_2+x_3 &= 4\\ x_1-x_2+x_4 &= 8\\ 3x_1+4x_2-x_5 &= 8\end{aligned}$$

$$x_j\ge 0,\left(j=\overline{1,4}\right).$$

Умножим левую и правую часть третьего уравнения на "-*1*":

$$2x_1 + 8x_2 + x_3 + 2x_4 - 2x_5 \to max$$

$$-2x_1 + 4x_2 + x_3 = 4$$

$$x_1 - x_2 + x_4 = 8$$

$$-3x_1 - 4x_2 + x_5 = -8$$

$$x_j \geq 0, \ (j = \overline{1,4}).$$

В системе ограничений этой задачи имеется полный набор единичных векторов A_3, A_4 и A_5, который является базисом очевидного решения системы уравнений-ограничений $\tilde{x} = (0, 0, 4, 8, -8)$. Если это решение – псевдоплан, то можно использовать двойственный симплекс-метод.

Заполним первую симплекс-таблицу. Коэффициенты разложения всех векторов по базису (единичному) известны, поэтому требуется по известным правилам симплекс-метода лишь вычислить оценки и значение целевой функции.

Баз	$C_{баз}$	A_0	*2*	*8*	*1*	*2*	*-2*
			A_1	A_2	A_3	A_4	A_5
A_3	*1*	*4*	*-2*	*4*	*1*	*0*	*0*
A_4	*2*	*8*	*1*	*-1*	*0*	*1*	*0*
A_5	*-2*	*-8*	*-3*	*-4*	*0*	*0*	*1*
Табл.1		*36*	*4*	*2*	*0*	*0*	*0*

Как видно из таблицы, записанное в ней решение является псевдопланом, так как среди оценок векторов нет отрицательных. Решение не является оптимальном, так как в столбце A_0 имеется отрицательный элемент $x_{30} = -8$.

Признака недопустимости задачи также нет, так как среди коэффициентов третьей строки есть отрицательные $(x_{31}, x_{32} < 0)$, следовательно, решение можно продолжить. Из базиса будет выводиться вектор A_5.

Для того же, чтобы решить вопрос, какой вектор следует вводить в базис, необходимо для всех векторов, имеющие отрицательные коэффициенты в третьей строке (в нашем случае это векторы A_1 и A_2), вычислить отношения $|\Delta_1 / x_{31}| = |4/-3|$ – для вектора A_1 и $|\Delta_3 / x_{32}| = |2/-4|$ – для вектора A_2. Минимальное из этих отношений соответствует вектору A_2, следовательно, этот вектор нужно вводить в базис вместо вектора A_5.

По правилам обычного симплекс-метода, приняв в качестве ведущего элемента коэффициент $x_{32} = -4$, формируем новую (вторую) симплекс- таблицу.

Баз	*Сбаз*	*Ао*	*2*	*8*	*1*	*2*	*-2*
			A1	*A2*	*A3*	*A4*	*A5*
A3	*1*	*-4*	*-5*	*0*	*1*	*0*	*1*
A4	*2*	*10*	*7/4*	*0*	*0*	*1*	*-1/4*
A2	*8*	*2*	*3/4*	*1*	*0*	*0*	*-1/4*
Табл.2		*32*	*5/2*	*0*	*0*	*0*	*1/2*

Новый псевдоплан не является оптимальным решением, так как в столбце A_0 имеется отрицательный элемент $x_{10} = -4$. Признака недопустимости задачи также нет, так как среди коэффициентов первой строки есть отрицательные $\left(x_{11} < 0\right)$, следовательно, решение можно продолжить. Из базиса будет выводиться вектор A_3. В базис может быть введен только вектор A_1, так как это – единственный вектор, имеющий отрицательный коэффициент (-5) в третьей строке.

Новая симплекс-таблица (третья):

Баз	*Сбаз*	*Ао*	*2*	*8*	*1*	*2*	*-2*
			A1	*A2*	*A3*	*A4*	*A5*
A1	*2*	*4/5*	*1*	*0*	*-1/5*	*0*	*-1/5*
A4	*2*	*43/5*	*0*	*0*	*7/20*	*1*	*1/10*
A2	*8*	*7/5*	*0*	*1*	*3/20*	*0*	*-1/10*
Табл.3		*30*	*0*	*0*	*1/2*	*0*	*1*

Получено оптимальное решение задачи, так как очередной псевдоплан является допустимым: в столбце A_0 нет отрицательных элементов.

$$Z_{опт}=30,\ X_{опт}=(4/5,7/5,0.43/5,0).$$

1.3.8. Определение исходного псевдоплана

При использовании двойственного симплекс-метода основные вычислительные сложности связаны с определением исходного псевдоплана. Однако двойственный симплекс-метод в части основной вычислительной процедуры считается более экономным, чем обычный симплекс-метод.

Частный случай определения исходного псевдоплана

В отдельных случаях исходный псевдоплан непосредственно определяется из исходной задачи, если она имеет специальную структуру.

Пусть требуется решить задачу ЛП:

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

$$\sum_{j=1}^{n} a_{ij} x_j \le a_i,\ i = \overline{1,m}$$

$$x_j \ge 0,\ j = \overline{1,n}.$$

В этой задаче свободные члены a_i могут иметь любой знак, однако самое существенное – это неположительность всех коэффициентов целевой функции ($c_j \le 0, j=1,2,...,n$).

Придадим этой задаче каноническую форму, используя дополнительные переменные:

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

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = a_i,\ i = \overline{1,m}$$

$$x_j \ge 0,\ j = \overline{1,n+m}\ .$$

Очевидно, что вектор

$$\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_{n+m}) = (\ 0,0,...,0,a_1,a_2,...,a_m\ )$$

удовлетворяют всем ограничениям-уравнениям этой задачи.

Векторы при дополнительных переменных $A_{n+1}, A_{n+2}\ ,..., A_{n+m}$ линейно независимы и составляют полный единичный базис системы ограничений-уравнений ($B=E$).

Учитывая, что $B=B^{-1}=E$, найдем оценки всех небазисных векторов:

$$\Delta_j = c_{баз} B^{-1} A_j - c_j\ ,\ \ (j=1,2,...,n\ ).$$

Но $c_{баз}$=($0,0,...,0$), т.е., $\Delta_j = -\ c_j$, ($j=1,2,...,n$), а по условию задачи все $c_j \le 0,\ \ j=1,2,...,n$. Следовательно, все $\Delta_j\ \ge 0$, т.е. $\overline{x}$ – псевдоплан, которому соответствует допустимое решение двойственной задачи: $y = c_{баз} B^{-1}$=($0,0,...,0$).

Требование $c_j \le 0$ – это достаточно серьезное ограничение, но тем не менее класс подобных моделей весьма широк.

Пример 1.21

Дана задача ЛП:

$$\begin{array}{rcrcl} 3x_1 &+& 6x_2 &\to& min, \\ x_1 && &\ge& 1 \\ x_1 &+& 2x_2 &\le& 5 \\ 4x_1 &-& 3x_2 &\le& 10 \\ && x_{1,2} &\ge& 0 \end{array} \quad \Rightarrow \quad \begin{array}{rcrcl} -3x_1 &-& 6x_2 &\to& max \\ -x_1 && &\le& -1 \\ x_1 &+& 2x_2 &\le& 5 \\ 4x_1 &-& 3x_2 &\le& 10 \\ && x_{1,2} &\ge& 0 \end{array}$$

Приведем систему ограничений к каноническому виду:

$$\begin{array}{rcrcccl} -3x_1 &-& 6x_2 & & & \to & max \\ -x_1 & & & +x_3 & & = & -1 \\ x_1 &+& 2x_2 & & +x_4 & = & 5 \\ 4x_1 &-& 3x_2 & & +x_5 & = & 10 \end{array}$$

$$x_j \ge 0,\ j=1 \div 5.$$

Решение задачи двойственным симплекс-методом:

			-3	-6	0	0	0
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5
A_3	0	-1 $\boxed{1}$	-1 $\boxed{1}$	0 $\boxed{0}$	1 $\boxed{-1}$	0 $\boxed{0}$	0 $\boxed{0}$
A_4	0	5	1	2	0	1	0
A_5	0	10	4	-3	0	0	1
Табл.1		0	3	6	0	0	0
A_1	-3	1	1	0	-1	0	0
A_4	0	4	0	2	1	1	0
A_5	0	6	0	-3	4	0	1
Табл.2		-3	0	6	3	0	0

Итак, оптимальное решение канонической задачи:

$$Z = -3;\ X_{опт}=(\ 1,0,0,4,6\ ).$$

Оптимальное решение исходной задачи:

$$Z_{исх} = 3;\ X_{опт}=(\ 1,0\ ).$$

Задача, двойственная к канонической, имеет следующее решение:

$$Z_{дв} = -\ 3;\ \ Y=(\ 3,0,0\ ).$$

Определение исходного псевдоплана введением дополнительного ограничения

Пусть дана каноническая задача ЛП, имеющая $n+m$ переменных. Каким угодно способом придадим задаче следующую форму:

$$\sum_{j=1}^{n+m} c_j x_j \to \max$$

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = a_i,\ i = \overline{1,m}$$

$$x_j \geq 0,\ j = \overline{1, n+m}.$$

Здесь свободные члены a_i ($i=1,2,...,m$) могут иметь любой знак.

Очевидно, что вектор

$$\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_{n+m}) = (\ 0,0,...,0,a_1,a_2,...,a_m\ )$$

удовлетворяет всем ограничениям-уравнениям этой задачи, причем переменным $\overline{x}_{n+1}, \overline{x}_{n+2}, ..., \overline{x}_{n+m}$, имеющим значения $a_1,a_2,...,a_m$, соответствует полная система единичных (линейно-независимых) векторов: $A_{n+1},A_{n+2},...,A_{n+m}$.

По сути, задаче в такой форме соответствует вполне определенная симплекс-таблица:

Баз	$C_{баз}$	A_0	c_1	…	c_k	…	c_n	c_{n+1}	…	c_{n+m}
			A_1	…	A_k	…	A_n	A_{n+1}	…	A_{n+m}
A_{n+1}	c_{n+1}	x_{10}	x_{11}	…	x_{1k}	…	x_{1n}	*1*	…	*0*
⋮	⋮	⋮	⋮		⋮		⋮	⋮		⋮
A_{n+m}	c_{n+m}	x_{m0}	x_{m1}	…	x_{mk}	…	x_{mn}	*0*	…	*1*
		Δ_0	Δ_1	…	Δ_k	…	Δ_n	*0*	…	*0*

Следует обратить внимание на тот факт, что оценки векторов могут иметь любые знаки и, главное, среди коэффициентов разложения вектора A_0 могут быть отрицательные. То есть, записанное в таблице решение – это не псевдоплан и не опорное решение.

Введем в состав ограничений дополнительное ограничение, имеющее вид:

$$\sum_{j=1}^{n} x_j + x_{n+m+1} = M\ ,$$

где x_{n+m+1} – дополнительная переменная, а M – достаточно большое положительное число. Это число подбирается таким образом, чтобы полупространство $x_1+x_2+...+x_n \leq M$ полностью включало в себя все допустимое множество задачи. Это множество (D), учитывая что все $x_{n+i} \geq 0$, можно представить в виде:

$$\sum_{j=1}^{n} a_{ij} x_j \leq a_i,\ i = \overline{1,m}$$

$$x_j \geq 0,\ j = \overline{1,n}.$$

Графически, подбор значения M можно проиллюстрировать следующим образом (рис.1.14):

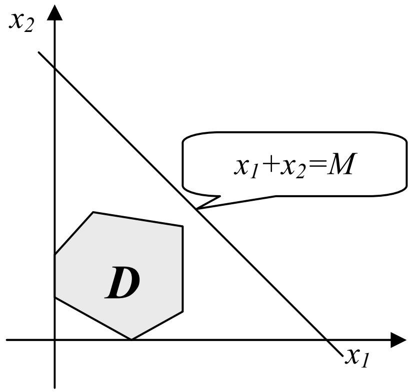


Рис.1.14. Иллюстрация подбора достаточно большого числа M

Введение нового ограничения приводит к следующему изменению в симплекс-таблице:

1 в базис вводится единичный вектор $A_{n+m+1}=(0,0,...,0,1)^T$ с единицей в $m+1$-й позиции;
2 все базисные векторы $A_{n+1},A_{n+2},...,A_{n+m}$ дополняются нулевой $m+1$-й координатой;
3 все свободные векторы $A_1,A_2,...,A_n$ дополняются единичной $m+1$-й координатой ($x_{m+1,j}=1$).

Ввиду того, что $c_{n+m+1}=0$, оценки векторов не изменяются.

Новой задаче соответствует следующая симплекс-таблица:

Баз	$C_{баз}$	A_0	c_1		c_k		c_n	c_{n+1}		c_{n+m}	0
			A_1		A_k		A_n	A_{n+1}		A_{n+m}	A_{n+m+1}
A_{n+1}	c_{n+1}	x_{10}	x_{11}		x_{1k}		x_{1n}	1		0	0
...											
A_{n+m}	c_{n+m}	x_{m0}	x_{m1}		x_{mk}		x_{mn}	0		1	0
A_{n+m+1}	0	M	1		**1**		1	0		0	1
		Δ_0	Δ_1		Δ_k		Δ_n	0		0	0

Выведем из базиса вектор A_{n+m+1} и заменим его некоторым небазисным вектором A_k ($k \in [1,2,...,n]$). Подсчитаем новые оценки свободных векторов A_j ($j \in [1,2,...,n]$):

$$\Delta'_j = \Delta_j - \frac{x_{m+1,j}}{x_{m+1,k}} \Delta_k = \Delta_j - \Delta_k \, .$$

Очевидно, что для того чтобы все новые оценки Δ'_j стали неотрицательными, в базис нужно вводить вектор с наименьшей из отрицательных оценок.

При выполнении этого условия новому базису будет соответствовать псевдоплан: $(x'_{10}, x'_{20}, ..., x'_{m+1,0})$.

Пересчитав по основным формулам все элементы, имеем исходную симплекс-таблицу для работы по двойственному симплекс-методу.

Столбец A_{n+m+1} в последующих вычислениях игнорировать.

Число M находится непосредственно из условий задачи или решением вспомогательной задачи:

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

$$\sum_{j=1}^{n} a_{ij} x_j \le a_i, \; i = \overline{1,m}$$

$$x_j \ge 0, j = \overline{1,n}.$$

Если исходная задача имеет хотя бы одно ограничение – нестрогое неравенство вида

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \le a_i \text{, где } a_{ij} > 0; \; a_i > 0,$$

то число M можно определить более простым способом.

Приведем эти неравенства к виду:

$$\frac{x_1}{\dfrac{a_i}{a_{i1}}} + \frac{x_2}{\dfrac{a_i}{a_{i2}}} + \ldots + \frac{x_n}{\dfrac{a_i}{a_{in}}} \le 1 .$$

Очевидно, что если взять

$$M = \max_{j=\overline{1,n}} \left\{ \frac{a_i}{a_{ij}} \right\},$$

то ограничение

$$x_1 + x_2 + \ldots + x_n \le M$$

не сократит область допустимых решений.

Этот подход в двумерном случае можно проиллюстрировать следующим образом.

Допустим в системе ограничений имеется ограничение вида:

$a_{i1}x_1 + a_{i2}x_2 \le a_i$, где $a_{ij} > 0$; $a_i > 0$ (рис.1.15).

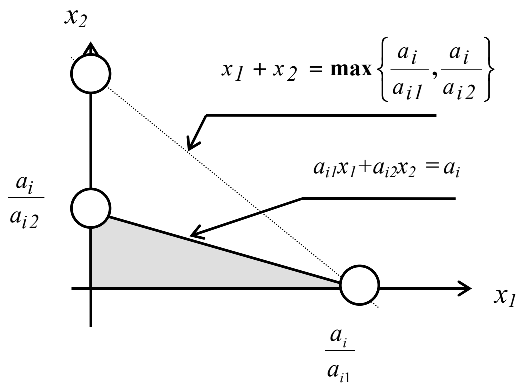


Рис.1.15. Выбор числа М для двумерного случая

Таким образом, нужно взять минимальный положительный элемент (пусть это будет a_{is}) и принять $M = a_i / a_{is}$.

Пример 1.22

Найти исходный псевдоплан следующей задачи ЛП:

$$8x_1 + 5x_2 - 2x_3 - 4x_4 + 5x_5 + x_6 + 3x_7 \rightarrow \max$$

$$x_1 - 2x_2 + 2x_3 + 4x_4 - 4x_5 + x_6 = -2$$

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + x_7 = 1$$

$$x_j \geq 0,\ j=1 \div 7.$$

В этой задаче, учитывая что базисная переменная $x_7 \geq 0$, второе ограничение-уравнение можно представить в виде:

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 \leq 1.$$

Все параметры этого ограничения положительные. Находим минимальный коэффициент $\min\{ a_{2j} \}=a_{21}=1$, откуда определяем $M= max\{ a_2 /a_{2j} \}= 1$.

Дополнительное ограничение имеет вид

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 1.$$

Это ограничение вводим в состав ограничений исходной задачи и придаем ему форму равенства (путем введения дополнительной переменной x_8

$$8x_1 + 5x_2 - 2x_3 - 4x_4 + 5x_5 + x_6 + 3x_7 \rightarrow \max$$

$$x_1 - 2x_2 + x_3 + x_4 - 4x_5 + x_6 = -2$$

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + x_7 = 1$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_8 = 1$$

$$x_j \geq 0,\ j=1 \div 8.$$

Ниже представлена симплекс-таблица, соответствующая этой задаче.

			8	5	-2	-4	5	1	3	0
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A6	1	-2	1	-2	2	4	-4	1	0	0
A7	3	1	1	2	3	4	5	0	1	0
A8	0	1	**1**	1	1	1	1	0	0	1
Табл.1		1	-4	-1	13	20	6	0	0	0

Минимальную по абсолютной величине оценку имеет вектор A_1 . Вводим этот вектор в базис, а выводим дополнительный вектор A_8 . Имеем следующую симплекс-таблицу:

			8	5	-2	-4	5	1	3	0
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_6	1	-3	0	-3	1	3	-5	1	0	-1
A_7	3	0	0	1	2	3	4	0	1	-1
A_1	8	1	1	1	1	1	1	0	0	1
Табл.2		5	0	3	17	24	10	0	0	4

Все оценки неотрицательны, а в столбце A_0 имеется отрицательный элемент. Следовательно, получен псевдоплан задачи

$$\tilde{x} = (1,0,0,0,0,-3,0,0).$$

Контрольные вопросы и задачи к разделу 1.3

1. Покажите взаимодвойственность следующей пары задач:

$$< c, x > \to \min, \quad AX = A_0, \quad x \ge 0.$$

$$< A_0^T, y > \to \max, \quad A^T Y \le C.$$

2. Рассмотрим следующую задачу линейного программирования. Пусть для функционирования атомной электростанции применяется n видов топлива. Существует m характеристик этих видов топлива. Пусть a_{ij} – i-я характеристика j-го вида топлива, a_i – минимально допустимое требование по i-й характеристике для электростанции в целом. Пусть c_j – стоимость одной тонны топлива вида j. Необходимо определить план закупки топлива для электростанции, минимальный по затратам при условии обеспечения выполнения минимальных требований по всем характеристикам. Постройте модель линейного программирования, соответствующую описанной задаче, постройте двойственную модель и приведите её экономическую интерпретацию.

3. Сформулируйте основные положения двойственного симплекс-метода для задачи на минимум целевой функции.

4. Чему соответствует псевдоплан при графической интерпретации задачи линейного программирования?

5. Почему задача получения исходного псевдоплана не всегда имеет решение?

6. Постройте задачу, двойственную к заданной

$$x_1 - 2x_2 + 4x_3 - x_4 \to \min$$

$$x_1 + x_2 + 2x_3 + x_4 \ge 10$$

$$-x_1 + 3x_2 + 2x_3 + 5x_4 = 6$$

$$4x_1 - x_2 - 3x_3 + 8x_4 \le 15$$

$$x_{1,3,4} \ge 0$$

x_2 – *не ограничена в знаке.*

7. Решите задачу симплекс-методом и получите оптимальное решение задачи, двойственной к заданной

$$6x_1 + 6x_2 \to \max$$
$$2x_1 + 4x_2 \le 102$$
$$4x_1 + 3x_2 \le 114$$
$$x_1 \le 18$$
$$x_2 \le 20$$
$$x_{1,2} \ge 0.$$

8. После очередной (не первой) итерации решения задачи ЛП (на максимум) двойственным симплекс-методом получена следующая симплекс-таблица. Найдите все элементы базисной матрицы оптимального решения этой задачи и обратной к ней матрицы.

Баз	$C_{баз}$	A_0	*-1*	*-18*	*1*	*2*
			A_1	A_2	A_3	A_4
A_1	*-1*	*2*	*1*	*-2*	*-1*	*0*
A_4	*2*	*-5*	*0*	*-4*	*2*	*1*

9. Задача ЛП на максимум решается двойственным симплекс-методом. Найти области параметров x и y, в которых:

- получено оптимальное решение задачи;
- задача не имеет решения;
- решение следует продолжить.

Баз	$C_{баз}$	A_0	*-1*	*-18*	*1*	*2*
			A_1	A_2	A_3	A_4
A_1	*-1*	*2*	*1*	*-2*	*-1*	*0*
A_4	*2*	x	*0*	y	*2*	*1*

10. Получите исходный псевдоплан и решите задачу двойственным симплекс-методом

$$
\begin{aligned}
&9x_1 + 6x_2 \to \max \\
&3x_1 + 4x_2 \le 125 \\
&4x_1 + 3x_2 \le 120 \\
&x_1 \le 18 \\
&x_2 \le 23 \\
&x_{1,2} \ge 0
\end{aligned}
$$

1.4. Вопросы вычислительной эффективности в симплекс-методе

1.4.1. Обоснование модифицированного симплекс-метода

Теория двойственности предоставила мощный аппарат для решения емких с точки зрения вычислений задач (рис.1.16):

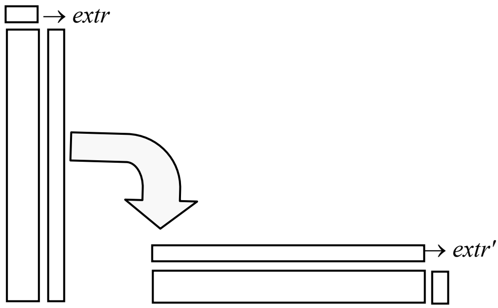


Рис.1.16. Решение двойственной задачи вместо прямой для ускорения вычислений

Тем не менее, существуют иные резервы повышения эффективности.

Пример 1.23

$$5y_1 + 14y_2 \to min$$

1.	$-y_1$				≤ 23
2.				y_2	≥ 4
3.	$-y_1$	$-$	2	y_2	≤ 5
4.	$-4y_1$	$-$	13	y_2	≤ 1
5.	$3y_1$	$+$	11	y_2	≥ 2
6.	$-y_1$	$-$	3	y_2	≤ 2
7.	$2y_1$	$+$	7	y_2	≥ 1
8.	$-2y_1$	$-$	3	y_2	≥ 10
9.	$-5y_1$	$-$	15	y_2	≤ 10
10	$-4y_1$	$-$	9	y_2	≤ 18

y_1, y_2 – не ограничены в знаке.

Если решать задачу "в лоб", то общее количество переменных составит:

- (2+2) – наложение на переменные требования неотрицательности;
- 10 – количество дополнительных переменных для перехода от нестрогих неравенств к равенствам;
- 4 – количество искусственных переменных для получения полного набора единичных векторов.

Всего – 18 переменных.

Таким образом, размерность задачи будет (10×18).

Мы же поступим по-другому:

1. изменим неравенства смысла "≤" на противоположные ("≥");
2. перейдем к двойственной задаче.

$$-23x_1+4x_2, -5x_3 - x_4 + 2x_5 - 2x_6 +x_7 +10x_8 - 10x_9 - 18x_{10} \to max$$

$$x_1 + x_3 + 4x_4 + 3x_5 +x_6 +2x_7 - 2x_8 + 5x_9 + 4x_{10} = 5$$

$$x_2 +2x_3 +13x_4+11x_5+3x_6+7x_7 - 3x_8 +15x_9 + 9x_{10} =14$$

$$x_j \ge 0, \quad (j=1..10).$$

Размерность этой задачи составляет (2×10).

Итак, трудоемкость решения задач линейного программирования существенным образом зависит от соотношения количества ограничений m (или количества строк в матрице системы ограничений) и количества неизвестных n (или количества столбцов в этой матрице). Наиболее технологичными являются задачи, у которых $n>>m$, так как для решения подобных задач требуется значительно меньшее количество дополнительных (и искусственных) переменных, необходимых для придания им канонической формы и получения единичной базисной матрицы исходного решения.

В этой связи использование результатов теории двойственности позволяет резко повысить эффективность решения задач линейного программирования. Действительно, вместо решения задачи , у которой $m>>n$, всегда можно перейти к двойственной задаче с $n>>m$ и, решив последнюю, найти решение прямой задачи, как это показано в разделе 1.3.3.

В этом разделе рассматривается модифицированный симплекс-метод, ориентированный на эффективное решение именно тех задач, у которых $n>>m$. Этот метод в настоящее время является основой многих коммерческих пакетов линейного программирования.

Рассмотрим задачу линейного программирования:

$$
\begin{gathered}
< c, x > \to \max \\
A_1 x_1 + A_2 x_2 + \ldots + A_n x_n = A_0 \qquad (1.86) \\
x \geq 0
\end{gathered}
$$

Пусть известно некоторое опорное решение задачи $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ и базис этого решения $B = (A_{i_1}, A_{i_2}, \ldots, A_{i_m})$.

В тех случаях, когда количество неизвестных n существенно превышает количество ограничений m, основной объем вычислений приходится на пересчет коэффициентов разложения небазисных векторов при переходе от одного опорного решения к другому– от одного базиса к другому, смежному с ним.

						c_s		c_j	
Баз	$C_{баз}$	A_0				A_s		A_j	
A_{i_1}	c_{i_1}	x_{10}				x_{1s}		x_{1j}	
A_{i_2}	c_{i_2}	x_{20}		B^{-1}		x_{2s}		x_{2j}	
A_{i_m}	c_{i_m}	x_{m0}				x_{ms}		x_{mj}	
		Δ_0				Δ_s		Δ_j	

Зададимся вопросом, с какой целью осуществляется этот пересчет? Это делается только для того, чтобы вычислить оценки небазисных векторов с использованием известного выражения:

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} x_{kj} - c_j \qquad (1.87)$$

или

$$\Delta_j = c_{баз} X_j - c_j, \tag{1.88}$$

где $c_{баз} = (c_{i_1}, c_{i_2}, ..., c_{i_m})$ – вектор коэффициентов целевой функции при базисных переменных задачи; $X_j = (x_{1j}, x_{2j}, ..., x_{mj})^T$ – вектор-столбец коэффициентов разложения вектора A_j по базису.

В дальнейшем будем считать, что $\Delta_0 = c_{баз} X_0 - c_0$ – значение целевой функции ($c_0 = 0$).

Учитывая, что $X_j = B^{-1} A_j$, где B^{-1} – обратная базисная матрица, выражение (1.88) можно переписать следующим образом

$$\Delta_j = c_{баз} B^{-1} A_j - c_j. \tag{1.89}$$

Если знать обратную базисную матрицу вычисление оценок можно проводить по следующей схеме: сначала вычислить вектор

$$\Lambda = (\lambda_1, \lambda_2, ..., \lambda_m) = c_{баз} B^{-1}, \tag{1.90}$$

после чего – вектор оценок:

$$\Delta_j = \Lambda A_j - c_j, \ (j = 0, 1, 2, ..., m). \tag{1.91}$$

Вектор $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ называется *вектором симплексных множителей*.

В соответствии с приведенной схемой, для вычисления оценок не требуется знать координаты разложения соответствующих векторов по базису – требуется лишь знать обратную матрицу.

Учитывая тот факт, что решение начинается с единичной базисной матрицы, достаточно, начиная с первой итерации симплекс-метода, хранить разложения всех исходных единичных векторов по текущему базису. В этом случае в каждой симплекс-таблице будет находиться обратная матрица.

Для вычисления же обратных матриц в соседних базисах можно использовать основные формулы обычного симплекс-метода.

Пусть, например, из базиса выводится вектор A_{i_r}. Вместо него в базис вводится вектор A_s. Тогда элементы новой обратной матрицы вычисляются по формулам:

$$\beta'_{ij} = \beta_{ij} - \frac{\beta_{rj}}{x_{rs}} x_{is} \, (i \neq r); \quad \beta'_{rj} = \frac{\beta_{rj}}{x_{rs}}.$$

Таким образом, для вычисления новой обратной матрицы необходимо иметь только коэффициенты разложения вводимого в очередной базис вектора. В данном случае это

$$X_s = \left(x_{1s}, x_{2s}, \ldots, x_{ms}\right)^T .$$

Итак, зная обратную базисную матрицу очередного опорного решения, можно, используя приведенную выше схему вычислений (1.90),(1.91), определить оценки всех небазисных векторов.

Если все оценки неотрицательны $\Delta_j \geq 0$ $(j = 1, 2, \ldots, n)$, процесс закончен: получено оптимальное решение задачи, координаты которого определяются коэффициентами разложения вектора A_0.

Новая ситуация возникает в том случае, когда среди оценок Δ_j $(j = 1, 2, \ldots, n)$ есть отрицательные оценки.

В обычном симплекс-методе для каждой отрицательной оценки $\Delta_j < 0$ проверяется признак неограниченности сверху целевой функции: проверяются знаки всех коэффициентов x_{kj}. Если обнаруживается, что среди этих коэффициентов нет ни одного строго положительного, принимается решение о неразрешимости задачи. То есть для принятия этого решения нужно иметь разложения всех векторов с отрицательной оценкой по базису. В модифицированном же симплекс-методе коэффициенты x_{kj} не известны, поэтому используется несколько упрощенная схема проверки на неограниченность целевой функции.

Выберем одну из отрицательных оценок (максимальную по абсолютной величине или, что используется чаще, первую обнаруженную отрицательную оценку). Пусть это будет $\Delta_S < 0$.

Найдем коэффициенты разложения вектора A_S по текущему базису (соответствующая процедура называется <u>*генерацией столбца*</u>):

$$X_s = \begin{pmatrix} x_{1s} \\ x_{2s} \\ \vdots \\ x_{ms} \end{pmatrix} = B^{-1} A_s .$$

Если все $x_{ks} \le 0$ $(k=1,2,...,m)$, целевая функция не ограничена сверху на допустимом множестве. В противном случае по правилам симплекс-метода определяется вектор, который должен быть выведен из базиса: коэффициенты разложения вектора A_0 известны; известны и коэффициенты разложения вектора A_S.

Новизна ситуации заключается в том, что факт неограниченности сверху целевой функции на данной итерации может быть и не установлен, так как проводится проверка только одного вектора A_S. Однако, рано или поздно, этот факт обязательно обнаружится.

1.4.2. Алгоритм модифицированного симплекс-метода

Дана задача линейного программирования:

$$< c, x > \to \max$$

$$A_1x_1 + A_2x_2 + ... + A_nx_n = A_0$$

$$x \ge 0.$$

Известно некоторое опорное решение задачи $x^* = (x_1^*, x_2^*, ..., x_n^*)$ и базис этого решения $B = (A_{i_1}, A_{i_2}, ..., A_{i_m})$.

Алгоритм модифицированного симплекс-метода рассмотрим по шагам.

<u>Шаг 1</u>. Вычисляется матрица B^{-1} – обратная базисная матрица. Ввиду того, что в качестве исходного опорного решения обычно принимается решение с единичным базисом, это вычисление не производится, так как в этом случае $B^{-1}=E=B$.

Вычисляются коэффициенты разложения вектора A_0 по базису B: $X_0=(x_{10},x_{20},...,x_{m0})^T$.

<u>Шаг 2</u>. С использованием выражений $\Lambda=(\lambda_1,\lambda_2,...,\lambda_m)=c_{баз}B^{-1}$ и $\Delta_j = \Lambda A_j - c_j$, $(j=0,1,2,...,n)$ вычисляются оценки Δ_j $(j=0,1,2,...,n)$, где Δ_0 – значение целевой функции.

<u>Шаг 3.</u> Если все $\Delta_j \ge 0$ $(j=1,2,...,n)$, то получено оптимальное решение задачи. Это решение представлено коэффициентами разложения вектора A_0 по базису B. Известно также оптимальное значение целевой функции – Δ_0.

<u>Шаг 4.</u> Если среди оценок Δ_j есть оценка $\Delta_S < 0$, то в качестве претендента на введение в базис принимается вектор A_S. Обычно это первый вектор, для которого вычисленная оценка отрицательная.

Шаг 5. Определяются коэффициенты разложения вектора A_S по базису B. Для этого используется выражение:

$$X_s = (x_{1s}, x_{2s}, ..., x_{ms}) = B^{-1} A_s .$$

Если все $x_{ks} \le 0$ $(k=1,2,...,m)$, то конец: задача не имеет решения, так как ее целевая функция не ограничена сверху на допустимом множестве. В противном случае для всех $x_{ks} > 0$ $(k=1,2,...,m)$ вычисляются отношения x_{k0} / x_{ks} и из этих отношений выбирается минимальное. Пусть

$$\min_{x_{ks}>0} \left\{ \frac{x_{k0}}{x_{ks}} \right\} = \frac{x_{r0}}{x_{rs}} ,$$

т.е., из базиса следует выводить вектор A_{i_r}.

Шаг 6. В базис B вместо вектора A_{i_r} вводится вектор A_S. По основным формулам пересчитываются элементы новой обратной матрицы, а также коэффициенты разложения вектора A_0 по новому базису и новое значение целевой функции. Далее выполняется шаг 2[4].

Пример 1.24

Решить задачу модифицированным симплекс-методом:

$$-23x_1+4x_2 -5x_3 - x_4 + 2x_5 - 2x_6 +x_7 +10x_8 - 10x_9 - 18x_{10} \to max$$

$$x_1 + x_3 + 4x_4 + 3x_5 +x_6 +2x_7 - 2x_8 + 5x_9 + 4x_{10} = 5$$

$$x_2 +2x_3+13x_4+11x_5+3x_6+7x_7 - 3x_8 +15x_9 + 9x_{10} =14$$

$$x_j \ge 0, \quad (j = \overline{1,10}).$$

В этой задаче имеется единичный базис $B = (A_1, A_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E = (E_1, E_2)$, которому соответствует исходное опорное решение задачи $x=(5,14,0,0,0,0,0,0,0,0)$.

Построим вспомогательную таблицу, в которую запишем параметры задачи – векторы-столбцы A_j $(j=1,2,...,10)$ и коэффициенты целевой функции (строка таблицы C).

[4] Новые симплексные множители можно вычислять на этом же шаге, используя основные формулы – правило "крест-накрест".

Вспомогательная таблица

№	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	$\Lambda^{(1)}$	$\Lambda^{(2)}$
1	1	0	1	4	3	1	2	-2	5	4	-23	-13
2	0	1	2	13	11	3	7	-3	15	9	4	4
C	-23	4	-5	-1	2	-2	1	10	-10	-18		
$\Delta^{(1)}$	0	0	-10	-39	-27	-9	-19	24	-45	-38		
$\Delta^{(2)}$	10	0	0	1	3	1	1	4	5	2		

В эту же таблицу на каждой итерации будем записывать координаты векторов симплексных множителей – в дополнительных столбцах таблицы, помеченных символами $\Lambda^{(t)}$, где t – номер итерации ($t=1,2,...$); и вычисленные оценки векторов – в дополнительных строках таблицы, помеченных символами $\Delta^{(t)}$.

Заполним первую основную таблицу. Сначала в этой таблице заполняются столбцы *БАЗ*, $C_{баз}$, A_0, и E_1, E_2 – столбцы обратной матрицы исходного опорного решения. Далее, как в симплекс-методе, вычисляется значение целевой функции ($\Delta_0= -59$).

Затем вычисляются симплексные множители – заполняется строка Λ, при этом для каждого столбца E_1 и E_2 производится такое же скалярное произведение, как и при вычислении значения целевой функции.

Вектор вычисленных симплексных множителей $\Lambda=(-23,4)$ переносится в столбец $\Lambda^{(1)}$ вспомогательной таблицы. В этой же таблице производится последовательное вычисление оценок небазисных векторов – до получения первой отрицательной оценки: заполняется строка $\Delta^{(1)}$.

В данном примере такой оценкой является оценка вектора[5] A_3

$$\Delta_3 = -23 * 1 + 4 * (-2) - (-5) = - 10.$$

Основные таблицы

			1	2	
Баз	$C_{баз}$	A_0	E_1	E_2	A_3
A_1	-23	5 $\boxed{5}$	1 $\boxed{1}$	0 $\boxed{0}$	1 $\boxed{1}$
A_2	4	14	0	1	2
Табл .1		-59 Λ	-23	4	Δ_3=-10

[5] Оценки остальных векторов приведены в таблице только в качестве справки, их вычисление не требуется.

A₃	-5	5	1	0
A₂	4	4	-2	1
Табл.2		-9 Λ	-13	4

Координаты вектора $A_3 = (1,2)$ из вспомогательной таблицы записываются в верхнюю часть первой основной таблицы.

Вычисляются коэффициенты разложения вектора A_3 по базису (генерация столбца):

$$X_3 = (x_{13}, x_{23})^T = B^{-1}A_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Следует отметить, что это – тривиальный случай, так как обратная матрица – единичная.

В таблицу записывается вычисленная оценка вектора A_3 (Δ_3=-*10*).

С использованием правил симплекс-метода по основным формулам вычисляются коэффициенты разложения вектора A_0 по новому базису B=(A_3, A_2), новое значение целевой функции и симплексные множители. Эти данные записываются в табл. 2.

Вектор вычисленных симплексных множителей Λ=(-13, 4) переносится в столбец $\Lambda^{(2)}$ вспомогательной таблицы. В этой же таблице производится последовательное вычисление оценок небазисных векторов – до получения первой отрицательной оценки: заполняется строка $\Delta^{(2)}$.

Все оценки неотрицательные, т.е. получено оптимальное решение задачи:

$$X_{опт} =(0,4,5,0,0,0,0,0,0,0);\ Z_{опт} =-9.$$

Пример 1.25

$$\begin{cases} -80y_1 - 180y_2 - 300y_3 - 80y_4 \to \max \\ \qquad\qquad y_2 + 3y_3 + y_4 - y_5 = 2 \\ y_1 + 2y_2 + 2y_3 - y_6 = 3 \\ \qquad y_j \geq 0,\ (j = \overline{1,6}). \end{cases}$$

Эта задача имеет очевидное исходное опорное решение с единичной базисной матрицей:

$$B = \left(A_4, A_1\right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};\ Y_{исх}= (3,0,0,2,0,0);\ Z= -80*3 -80*2 = -400.$$

При этом $B^{-1}=(E_1,E_2)=E=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Построим вспомогательную таблицу, в которую запишем параметры задачи – векторы-столбцы A_j ($j=1,2,...,6$) и коэффициенты целевой функции (строка таблицы C).

Вспомогательная таблица

№	A_1	A_2	A_3	A_4	A_5	A_6	$\Lambda^{(1)}$	$\Lambda^{(2)}$	$\Lambda^{(3)}$
1	*0*	*1*	*3*	*1*	*-1*	*0*	*-80*	*-80*	*-60*
2	*1*	*2*	*2*	*0*	*0*	*-1*	*-80*	*-50*	*-60*
C	*-80*	*-180*	*-300*	*-80*	*0*	*0*			
$\Delta^{(1)}$	*0*	*-60*	...	...	...	...			
$\Delta^{(2)}$	*30*	*0*	*-40*	...	...	...			
$\Delta^{(3)}$	*20*	*0*	*0*	*20*	*60*	*60*			

В эту же таблицу на каждой итерации будем записывать координаты векторов симплексных множителей – в дополнительных столбцах $\Lambda^{(1)}$, $\Lambda^{(2)}$,... . В строках $\Delta^{(1)}$, $\Delta^{(2)}$,... будем записывать вычисленные оценки векторов.

Заполним первую таблицу.

			1	*2*	$\leftarrow A_2$ $X_2=B^{-1}A_2$ $\downarrow$
БАЗ	$C_{баз}$	A_0	E_1	E_2	A_2
A_4	*-80*	*2*	*1*	*0*	*1*
A_1	*-80*	*3* $\boxed{3/2}$	*0* $\boxed{0}$	*1* $\boxed{1/2}$	***2*** $\boxed{\mathbf{1}}$
Табл.1		*-400*	*-80*	*-80*	*-60*
		$\uparrow\Delta_0$	$\uparrow\lambda_1$	$\uparrow\lambda_2$	$\uparrow\Delta_2$

Таблица заполняется в следующей последовательности.

Заполняются столбцы *Баз*, $C_{баз}$, A_0, и E_1, E_2 – столбцы обратной матрицы исходного опорного решения.

Вычисляется значение целевой функции (как в обычном симплекс-методе): $\Delta_0=-80\times2+(-80\times3)=-400$.

Вычисляются симплексные множители – координаты вектора $\Lambda^{(1)}=(\lambda_1, \lambda_2)$: $\lambda_1=-80\times1+(-80\times0)=-80$; $\lambda_2=-80\times0+(-80\times1)=-80$.

Вектор симплекс-множителей $\Lambda^{(1)}=(-80,-80)$ переносится в столбец $\Lambda^{(1)}$ вспомогательной таблицы. В этой же таблице производится последовательное вычисление оценок небазисных векторов до получения первой отрицательной оценки – заполняется строка $\Delta^{(1)}$.

В рассматриваемом примере отрицательную оценку имеет вектор A_2: $\Delta_2=-80\times1+(-80)\times2-(-180)=-60$. Остальные оценки не вычисляются.

Координаты вектора $A_2=(1,2)^T$ из вспомогательной таблицы переносятся в верхнюю часть основной таблицы (табл.1), после чего производится "генерация столбца" – нахождение коэффициентов разложения вектора A_2 по базису $B=(A_4,A_1)$:

$$X_2=(x_{12},x_{22})=B^{-1}A_2=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\times\begin{pmatrix}1\\ 2\end{pmatrix}=\begin{pmatrix}1\\ 2\end{pmatrix}.$$

В таблицу записывается вычисленная оценка вектора A_2 : $\Delta_2=-60$.

С использованием правил обычного симплекс-метода по основным формулам (правило "крест-накрест") вычисляются коэффициенты разложения вектора A_0 по новому базису $B=(A_4,A_2)$, новая обратная базисная матрица, новое значение целевой функции и новый вектор симплексных множителей.

Все данные записываются в новую таблицу (табл.2). Вычисленный вектор симплексных множителей $\Lambda^{(2)}=(-80,-50)$ переносится в столбец $\Lambda^{(2)}$ вспомогательной таблицы. Определяется вектор с отрицательной оценкой A_3: $\Delta_3=-40$ и т.д.

			3	2	
Баз	*Сбаз*	*A0*	*E1*	*E2*	*A3*
A4	*-80*	*1/2* $\boxed{1/4}$	*1* $\boxed{1/2}$	*-1/2* $\boxed{-1/4}$	**2** $\boxed{\mathbf{1}}$
A2	*-180*	*3/2*	*0*	*1/2*	*1*
Табл.2		*-310*	*-80*	*-50*	*-40*

Баз	*Сбаз*	*A0*	*E1*	*E2*	
A3	*-300*	*1/4*	*1/2*	*-1/4*	
A2	*-180*	*5/4*	*-1/2*	*3/4*	
Табл.3		*-300*	*-60*	*-60*	

На третьей итерации оценки всех векторов становятся неотрицательными. Получено оптимальное решение задачи:

$$Y_{опт}=(0,5/4,1/4,0);\ \ Z_{опт}=-300.$$

По третьей таблице можно определить решение задачи, двойственной к исходной задаче.

Решенная задача:

$$\begin{cases}-80y_1-180y_2-300y_3-80y_4\to\max\\ \qquad\quad y_2+3y_3+y_4-y_5=2\\ y_1+2y_2+2y_3-y_6=3\\ y_j\ge 0,\ (j=\overline{1,6}).\end{cases}$$

Двойственная задача:

$$2x_1 + 3x_2 \to min$$
$$x_2 \geq -80$$
$$x_1 + 2x_2 \geq -180$$
$$3x_1 + 2x_2 \geq -300$$
$$x_1 \qquad \geq -80$$
$$-x_1 \qquad \geq 0$$
$$-x_2 \geq 0$$
x_1, x_2 – *не ограничены в знаке.*

Оптимальное решение этой задачи:

$$X_{опт} = \Lambda^{(3)} = c_{баз}B^{-1} = (-60, -60); \;\; Z_{опт} = -300.$$

1.4.3. Мультипликативное представление обратной матрицы

Существующие в настоящее время программные пакеты позволяют решать задачи ЛП очень большой размерности.

Так, если имеется оперативная память достаточного объема, эти пакеты позволяют решать задачи, содержащие от 8000 до 16000 строк в матрице ограничений. Если же учесть, что обычно в моделях ЛП столбцов больше, чем строк, число элементов матрицы ограничений в таких задачах колеблется от 106 млн до 108 млн.

Решение подобных задач оказывается возможным благодаря тому, что большинство элементов матрицы – нулевые. Количество же ненулевых элементов обычно не превышает *0.5%* от общего количества.

Для решения задач такой размерности разработаны специальные процедуры управления данными.

Как уже говорилось, современные пакеты программ в основном используют модифицированный симплекс-метод.

При этом данные программы ориентированы на *мультипликативную форму представления обратной матрицы*.

Здесь используется алгебра матриц, позволяющая вычислять элементы новой обратной матрицы по известной предыдущей обратной матрице, при условии, что соответствующие базисные матрицы – соседние.

Пусть $B^{-1} = (\beta_{ij})$ – обратная матрица, соответствующая некоторому базису $B = (A_1, A_2, \ldots, A_r, \ldots, A_m)$.

Пусть далее вектор A_r выводится из базиса и вместо него в базис вводится вектор A_s . То есть новая базисная матрица имеет вид:

$$B_{нов} = (A_1, A_2, \ldots, A_{r-1}, A_s, A_{r+1}, \ldots, A_m).$$

Примем обозначение:

$$X_s = \begin{pmatrix} x_{1s} \\ x_{2s} \\ \vdots \\ x_{rs} \\ \vdots \\ x_{ms} \end{pmatrix},$$

где x_{ij} - i-я координата в разложении вектора A_S по старому базису B. Соответственно, x_{rs} – ведущий элемент.

Рассмотрим произведение двух матриц:

$$\begin{array}{c} 1 \\[12pt] 2 \\[12pt] \dots \\[12pt] i \\[12pt] \dots \\[12pt] r \\[12pt] \dots \\[12pt] m \end{array}
\begin{Vmatrix}
1 & 0 & \dots & 0 & \dots & -\dfrac{x_{1s}}{x_{rs}} & \dots & 0 \\
0 & 1 & \dots & 0 & \dots & -\dfrac{x_{2s}}{x_{rs}} & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & 1 & \dots & -\dfrac{x_{is}}{x_{rs}} & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & 0 & \dots & \dfrac{1}{x_{rs}} & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & 0 & \dots & -\dfrac{x_{ms}}{x_{rs}} & \dots & 1
\end{Vmatrix}
\times
\begin{Vmatrix}
\beta_{11} & \beta_{12} & \dots & \beta_{1j} & \dots & \beta_{1s} & \dots & \beta_{1m} \\
\beta_{22} & \beta_{22} & \dots & \beta_{2j} & \dots & \beta_{2s} & \dots & \beta_{2m} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\beta_{i1} & \beta_{i2} & \dots & \beta_{ij} & \dots & \beta_{is} & \dots & \beta_{im} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\beta_{r1} & \beta_{r2} & \dots & \beta_{rj} & \dots & \beta_{rs} & \dots & \beta_{rm} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\beta_{m1} & \beta_{m2} & \dots & \beta_{mj} & \dots & \beta_{ms} & \dots & \beta_{mm}
\end{Vmatrix} =$$

$$\begin{array}{cccccccc} 1 & 2 & \dots & i & \dots & r & \dots & m \end{array}$$

$$
= \begin{array}{c||ccc||}
 & \dots & \dots & \dots \\
i & \dots & \beta_{ij} - \dfrac{\beta_{rj}}{x_{rs}} x_{is} & \dots \\
\dots & \dots & \dots & \dots \\
r & \dots & \dfrac{\beta_{rj}}{x_{rs}} & \dots \\
\dots & \dots & \dots & \dots \\
\multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{j} & \multicolumn{1}{c}{}
\end{array}
\quad = (\beta'_{ij}) = B^{-1}_{нов.}
$$

Новая обратная матрица

Такое произведение автоматически реализует основные формулы.

Для левой ("почти единичной") матрицы примем обозначение $\tilde{E}$ – это <u>*элементарная матрица*</u>.

Имеем соотношение:

$$B^{-1}_{нов} = \tilde{E} B^{-1}_{стар}.$$

Рассмотрим основные моменты работы с использованием мультипликативной формы представления обратной матрицы.

По сути это тот же модифицированный симплекс-метод с отдельными особенностями.

Предполагаем, что известно исходное опорное решение с единичной базисной матрицей:

$$B = B^{-1} = E.$$

<u>**Цикл 1**</u>.

По обычному для МСМ правилу вычисляем вектор симплексных множителей:

$$\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) = c_{баз} B^{-1} = c_{баз} E = c_{баз},$$

а далее – оценки:

$$\Delta_j = \Lambda A_j - c_j \;\; (j = 1, 2, \dots, n).$$

Выбираем $\Delta_{s_1} < 0$ и ищем разложение вектора A_{s_1} :

$$X_{s_1} = B^{-1} A_{s_1} = E \; A_{s_1} = A_{s_1}.$$

Если все $x_{is_1 \le 0}$, ЦФ не ограничена сверху на допустимом множестве (*i=1,2,...,m*). В противном случае из базиса выводим вектор A_{r_1} такой, что:

$$\frac{x_{r_1 0}}{x_{r_1 s_1}} = \min_{x_{ks_1}>0}\left\{\frac{x_{k0}}{x_{ks_1}}\right\}.$$

Новая базисная матрица имеет вид

$$B_1^{-1} = \tilde{E}_1 B^{-1} = \tilde{E}_1 E = \tilde{E}_1,$$

где $\tilde{E}_1$ – первая элементарная матрица.

$$\tilde{E}_1 = \begin{array}{c} \left\| \begin{array}{cccccc} 1 & 0 & \dots & -\dfrac{x_{1s_1}}{x_{r_1 s_1}} & \dots & 0 \\ 0 & 1 & \dots & -\dfrac{x_{2s1}}{x_{r_1 s_1}} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dfrac{1}{x_{r_1 s_1}} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\dfrac{x_{ms1}}{x_{r_1 s1}} & \dots & 1 \end{array} \right\| \\ \textit{r_1 - й} \\ \textit{столбец} \end{array} \begin{array}{c} 1 \\ \\ 2 \\ \\ \\ \textit{r_1-я строка} \\ \\ m \\ \\ \\ \end{array}$$

Для хранения этой матрицы в памяти ЭВМ нужно:

1 знать единственный неединичный вектор этой матрицы и номер позиции, которую этот вектор занимает в матрице (в данном случае – это r_1, $1 \le r_1 \le m$);

2 кроме того, при сильной разреженности матрицы (много нулей) запоминаются только отличные от нуля элементы неединичного вектора-столбца и данные о позициях этих элементов.

Цикл 2.

Вычисляется вектор симплексных множителей:

$$\Lambda = c_{баз} B_1^{-1} = c_{баз} \tilde{E}_1,$$

а далее – оценки Δ_j ($j=1,2,...,n$). Если обнаруживается $\Delta_{s_2} \leq 0$, ищется разложение вектора A_{s_2}:

$$X_{s_2} = B_1^{-1} A_{s_2} = \tilde{E}_1 A_{s_2},$$

Если не обнаруживается факт неограниченности ЦФ, определяется вектор A_{r_2}, который нужно выводить из базиса.

Новая базисная матрица имеет вид:

$$B_2^{-1} = \tilde{E}_2 B_1^{-1} = \tilde{E}_2 \tilde{E}_1,$$

где $\tilde{E}_2$ – вторая элементарная матрица.

$$\begin{array}{cl}
\tilde{E}_2 = \left\| \begin{array}{cccccc}
1 & 0 & \dots & -\dfrac{x_{1s_2}}{x_{r_2 s_2}} & \dots & 0 \\
0 & 1 & \dots & -\dfrac{x_{2s2}}{x_{r_2 s_2}} & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & -\dfrac{1}{x_{r_2 s_2}} & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & -\dfrac{x_{ms2}}{x_{r_2 s2}} & \dots & 1
\end{array} \right\| & \begin{array}{l} 1 \\ \\ 2 \\ \\ \\ r_2\text{-я строка} \\ \\ \\ m \end{array} \\
\qquad\qquad\qquad r_2 \text{ - й столбец} &
\end{array}$$

Теперь нужно знать неединичный вектор этой матрицы и номер позиции, которую он занимает.

Непосредственно в памяти ЭВМ матрица B_2^{-1} не хранится. Хранятся две элементарные матрицы, точнее, информация о неединичных столбцах этих матриц – $\tilde{E}_1$ и $\tilde{E}_2$.

<u>Цикл k. (Основные моменты)</u>

Вычисление симплексных множителей:

$$\Lambda = c_{баз} B_{k-1}^{-1} \Rightarrow \quad \Lambda = (...((c_{баз} \tilde{E}_{k-1}) \tilde{E}_{k-2})...) \tilde{E}_1.$$

Генерация столбца:

$$X_{s_k} = B_{k-1}^{-1} A_{s_k} \quad \Rightarrow X_{s_k} = \tilde{E}_{k-1} (...\tilde{E}_2 (\tilde{E}_1 A_{s_k})...).$$

Преобразование обратной матрицы:

$$B_k^{-1} = \tilde{E}_k B_{k-1}^{-1} \Rightarrow B_k^{-1} = \tilde{E}_k \times \tilde{E}_{k-1} \times ... \times \tilde{E}_1.$$

1.4.4. Блочное программирование

Итак, мы достаточно подробно изучили симплекс-метод, двойственный симплекс-метод, а также модифицированный симплекс-метод.

Что можно сказать об этих методах? Прежде всего, это – *<u>универсальные</u>*, *<u>конечные</u>* методы.

Конечность – это гарантия решения задачи ЛП за конечное количество шагов (пусть и очень большое).

Универсальность означает, что данные методы практически не учитывают особенности строения матрицы коэффициентов ограничений. Главное, что задача относится к классу задач ЛП:

$$\langle c,x \rangle \to \max$$
$$AX = A_0$$
$$x \geq 0.$$

И вот здесь мы сталкиваемся с фундаментальным законом математического программирования: за универсальность нужно платить. Иными словами, конкретную задачу лучше, эффективнее решать методом, ориентированным на особенности именно этой задачи, чем методом, полностью игнорирующим любые особенности.

Например, в нелинейном программировании вообще нет универсальных методов. Практически каждый метод (за исключением, разве что, полного перебора) предъявляет определенные требования как к целевой функции, так и к функциям, определяющим множество допустимых решений[6].

---

[6] В этой связи следует отметить, что важнейшая составляющая процесса решения произвольной задачи нелинейного программирования – это анализ задачи на предмет определения возможности использования той или иной группы методов (например, выпуклый анализ)

Но вернемся к линейному программированию. Некоторые задачи ЛП большой размерности имеют такие структурные особенности, которые позволяют найти их оптимальное решение путем *декомпозиции* (или разложения) исходной задачи на ряд подзадач меньшей размерности. Причем эти подзадачи решаются практически независимо друг от друга. Основной эффект при таком подходе связан с тем, что существенно легче решить много небольших задач, чем одну большую. В математическом программировании это связано с полиномиальной и даже экспоненциальной зависимостью трудоемкости решения задач от их размерности.

Преимущество методов, использующих декомпозицию исходной задачи на подзадачи меньшей размерности, заключается в том, что с помощью этих методов оказывается возможным решение задач такой размерности, которые не удается решить другими методами из-за практически невыполнимого объема вычислений.

Ситуации, приводящие к постановке подобных задач, встречаются довольно часто.

Например, это – задачи планирования производственной деятельности предприятий, входящих в объединение. Хотя каждое из таких предприятий может иметь свои независимые ограничения, производственные программы отдельных предприятий обычно согласовывают на верхнем уровне управления из-за объективно существующих ограничений (связанных, например, с общим объемом финансовых ресурсов, которыми располагает объединение).

При этом возникает задача ЛП, обладающая структурой, которую в схематичной форме можно представить следующим образом (рис.1.17).

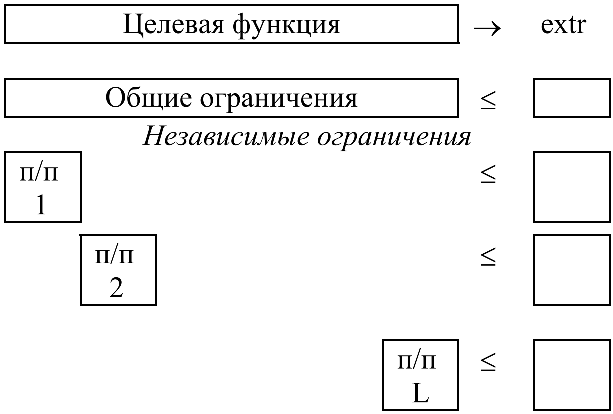


Рис.1.17. Задача с блочно-диагональной структурой и блоком-связкой

Независимые ограничения имеют блочно-диагональную структуру. При отсутствии общих ограничений различные виды производств могут рассматриваться, как совершенно независимые.

Поэтому, если общие ограничения, образующие *блок-связку*, отсутствуют, то проблем с декомпозицией не возникает (рис.1.18):

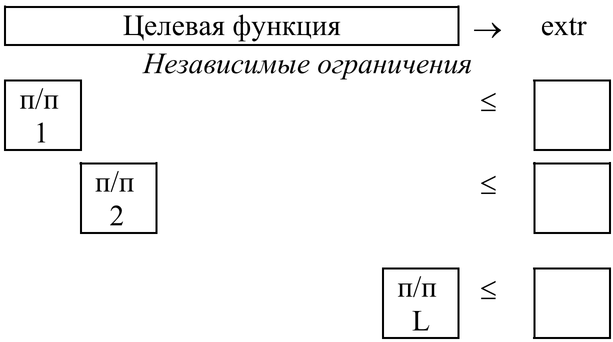


Рис.1.18 Задача только с блочно-диагональной структурой

Решить подобную задачу можно, разделив ее на L независимых задач, решить каждую из этих задач и объединить полученные оптимальные решения.

При этом $Z_{опт} = \sum_{i=\overline{1,L}} Z^{i}_{опт}$

Наличие же блока-связки, в общем случае, приводит к тому, что $Z_{опт} \neq \sum_{i=\overline{1,L}} Z^{i}_{опт}$. Дело в том, что, как правило, оптимизация какой-либо одной частной задачи ущемляет возможность оптимизации другой задачи.

Теория блочного программирования ориентирована на решение методом декомпозиции задач, имеющих как общие, так и независимые ограничения.

Класс задач с подобной структурой достаточно широк. В частности, к этому классу очень просто сводятся задачи, имеющие матрицу ограничений вида (рис.1.19):

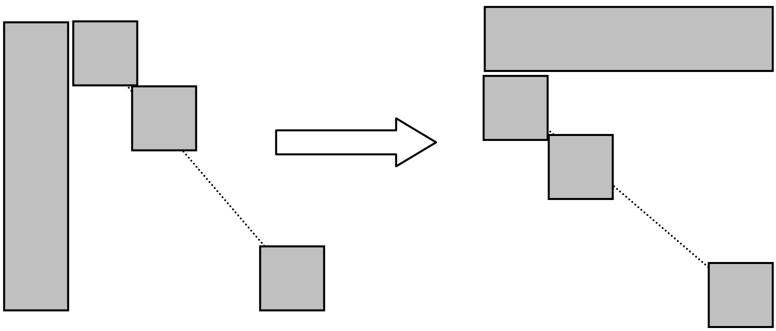


Рис.1.19. Использование двойственности для сведения задачи к блочно-диагональному виду с блоком-связкой

Ясно, что двойственная задача имеет блочно-диагональную структуру с блоком-связкой.

Более того, при сильной разреженности матрицы ограничений можно (небезуспешно) попытаться путем перестановки строк и столбцов придать произвольной задаче подобную структуру.

1.4.5. Метод декомпозиции Данцига-Вулфа: построение координирующей задачи

Рассмотрим некоторую задачу ЛП, имеющую блочно-диагональную структуру.

Пусть задача имеет n переменных, пронумерованных числами натурального ряда: $\{1,2,...,n\}$.

Множество индексов (номеров) переменных разобьем на L подмножеств по принципу принадлежности соответствующих переменных к 1-му, 2-му, ... , L-му диагональному блоку.

$\omega_1=\{1, 2, \dots ,q_1\}$,
$\omega_2=\{ q_1+1, q_1+2, \dots ,q_2\}$,
....
$\omega_\nu=\{ q_{\nu-1}+1, q_{\nu-1}+2, \dots ,q_\nu\}$,
....
$\omega_L=\{ q_{L-1}+1, q_{L-1}+2, \dots ,q_L\}$.

Здесь $q_L=n$, $q_0=0$.

Такое разбиение позволяет записать задачу с блочно-диагональной структурой в следующем виде.

$$\sum_{j\in\omega_1} c_j x_j + \sum_{j\in\omega_2} c_j x_j + \dots + \sum_{j\in\omega_\nu} c_j x_j + \dots + \sum_{j\in\omega_{L1}} c_j x_j \to \max$$

$$\sum_{j\in\omega_1} A_j x_j + \sum_{j\in\omega_2} A_j x_j + \dots + \sum_{j\in\omega_\nu} A_j x_j + \dots + \sum_{j\in\omega_L} A_j x_j = B_0$$

$$\sum_{j\in\omega_1} D_j x_j \qquad\qquad\qquad\qquad\qquad\qquad = B_1$$

$$\sum_{j\in\omega_2} D_j x_j \qquad\qquad\qquad\qquad = B_2$$

....

$$\sum_{j\in\omega_\nu} D_j x_j \qquad\qquad = B_\nu$$

....

$$\sum_{j\in\omega_L} D_j x_j = B_L$$

$$x_j \ge 0, (j=\overline{1,n})$$

Здесь

A_j – r_0-мерный вектор-столбец коэффициентов при переменной x_j из блока-связки;

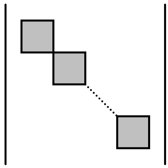
D_j – вектор-столбец коэффициентов при переменной x_j , принадлежащий диагональному блоку ν (если $j\in\omega_\nu$). Его размерность – r_ν ;

B_ν – r_ν-мерный вектор-столбец свободных членов в ν-м диагональном блоке ($\nu=1\div L$).

Размерность этой задачи $(\sum_{k=0}^{L} r_k) \times n$. Здесь $\sum_{k=0}^{L} r_k$ – общее количество ограничений.

Будем считать, что в задаче имеется один блок-связка $AX=B_0$ и один-единственный диагональный блок $DX=B^*$.

Здесь $B^* = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_L \end{pmatrix}$, а D – матрица, имеющая блочно-диагональную структуру. Эта матрица состоит из L диагональных блоков.

D=


Теперь задачу можно записать следующим образом:

$$\sum_{j=1}^{n} c_j x_j \to \max$$
$$AX = B_0$$
$$DX = B^*$$
$$x \geq 0.$$

Для простоты положим, что $\{x \mid DX = B^*,\ x \geq 0\}$ – ограниченное не пустое множество. Кроме того, пусть:

$$\{x \mid AX = B_0, DX = B^*,\ x \geq 0\} \neq \varnothing.$$

Как известно, любую точку ограниченного выпуклого множества можно представить, как выпуклую линейную комбинацию экстремальных точек (вершин).

То есть, если $\widehat{x}^1$, $\widehat{x}^2$, ... , $\widehat{x}^k$ – координаты вершин множества $\{x \mid DX = B^*,\ x \geq 0\}$, то это множество состоит из всех возможных точек вида:

$$\sum_{l=1}^{k} \beta_l \widehat{x}^l \text{, где } \beta_l \geq 0 \ (l = \overline{1,k}\,) \text{ и } \sum_{l=1}^{k} \beta_l = 1. \qquad (1.92)$$

Очевидно, что любая точка, представленная в виде (1.92), удовлетворяет всем ограничениям $DX = B^*$, $x \geq 0$ (рис. 1.20).

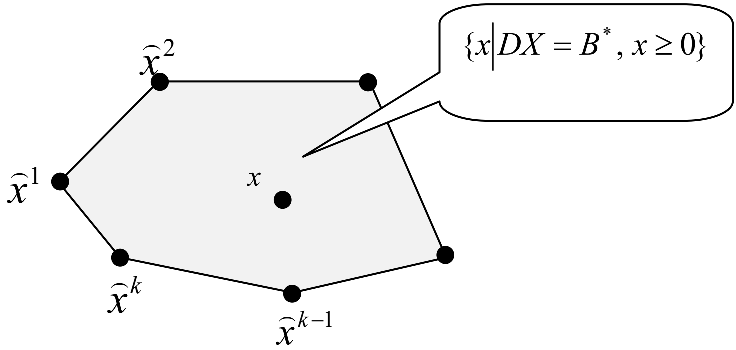


Рис.1.20. Иллюстрация множества $DX = B^*$

Здесь $\widehat{x}^l = (x_1^l, x_2^l, ..., x_n^l)$.

Это обстоятельство позволяет видоизменить исходную задачу таким образом, что из рассмотрения будут исключены все ограничения диагонального блока.

Примем обозначения:

$c = (c_1, c_2, ..., c_n)$ – вектор коэффициентов ЦФ;

A=(A_1,A_2, ... ,A_n) – матрица, составленная из векторов блока-связки.

Теперь задача приобретает следующий вид:

$$\begin{cases} \sum_{l=1}^{k} c\widehat{x}^l \beta_l \to \max \\ \left.\sum_{l=1}^{k} A\widehat{x}^l \beta_l = B_0 \right\} r_0 \text{ ограничений} \\ \sum_{l=1}^{k} \beta_l = 1 \\ \beta_l \geq 0, \quad (l = \overline{1,k}) \end{cases} \qquad (1.93)$$

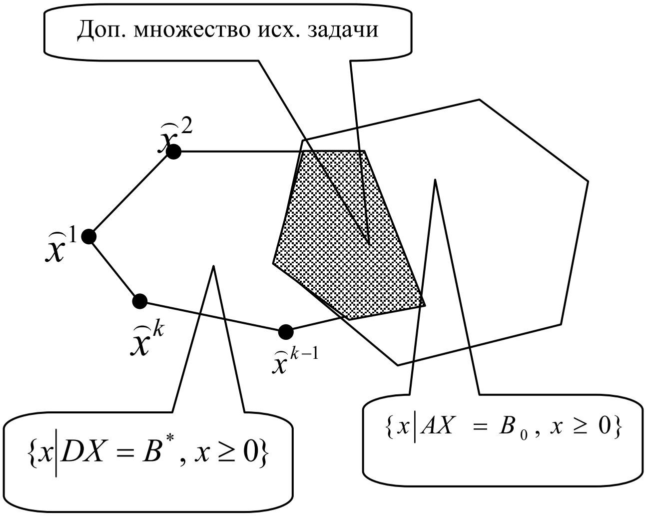


Рис.1.21 Допустимое множество исходной задачи

Как видно, все ограничения, составляющие диагональный блок $DX = B^*$, исчезли. А таких ограничений $\sum_{\lambda=1}^{L} r_\lambda$!

Добавилось же только одно – $\sum_{l=1}^{k} \beta_l = 1$.

Но в новой задаче появились новые переменные $\beta_l \geq 0$ $(l = \overline{1,k})$. Их количество – количество всех вершин.

Там не менее ясно, что найдя значения этих переменных, можно найти и решение исходной задачи:

$$X = \sum_{l=1}^{k} \beta_l \widehat{x}^l .$$

Задачу, соответствующую этой новой формулировке (1.93) будем называть *<u>главной,</u>* или, что то же самое, *<u>координирующей задачей</u>*.

Упростим обозначения:

$$c\widehat{x}^l = c_l\,;\; P_l = \begin{pmatrix} A\widehat{x}^l \\ 1 \end{pmatrix};\; P_0 = \begin{pmatrix} B_0 \\ 1 \end{pmatrix} \Bigg\} \, r_0 + 1.$$

Пришли к следующей задаче:

$$\begin{cases} \sum_{l=1}^{k} c_l \beta_l \to \max \\ \sum_{l=1}^{k} P_l \beta_l = P_0 \\ \beta_l \geq 0, \quad (l = \overline{1,k}). \end{cases} \tag{1.94}$$

<u>Пример 1.26</u>

$$\begin{aligned} &2x_1+3x_2\to\max \\ &x_1+x_2 \leq 15 \\ &-3x_1+x_2 \leq 9 \\ &4x_1 - x_2 \leq 8 \\ &x_{1,2} \geq 0. \end{aligned}$$

Будем считать, что блок-связка в этой задаче представлен единственным (первым) ограничением. Диагональный блок – вторым и третьим ограничениями. Т.е.

$$2x_1 + 3x_2 \to \max, \quad \Rightarrow <c,x> \to \max : \; c=(2,3)$$

$$x_1 + x_2 \leq 15 \quad \Rightarrow AX = B_0 \quad : \; A=(1,1); \; B_0=(15)$$

$$\left.\begin{aligned} -3x_1 + x_2 &\leq 9 \\ 4x_1 - x_2 &\leq 8 \end{aligned}\right\} \Rightarrow DX = B^* \quad : \begin{Vmatrix} -3 & 1 \\ 4 & -1 \end{Vmatrix} \times \begin{Vmatrix} x_1 \\ x_2 \end{Vmatrix} = \begin{Vmatrix} 9 \\ 8 \end{Vmatrix}$$

$$x_{1,2} \geq 0 \quad \Rightarrow \quad x \geq 0.$$

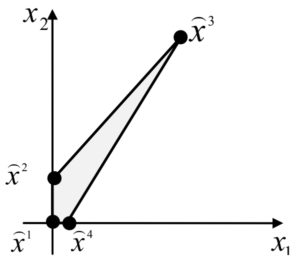
Множество $DX = B^*$, $\boldsymbol{x} \geq 0$ имеет 4 вершины:

$\widehat{x}^1 = (0,0)$ $c_1=((2,3),(0,0))=0$

$\widehat{x}^2 = (0,9)$ $c_2=((2,3),(0,9))=27$

$\widehat{x}^3 = (17,60)$ $c_3=((2,3),(17,60))=214$

$\widehat{x}^4 = (2,0)$ $c_4=((2,3),(2,0))=4$



$$P_0 = \begin{pmatrix} B_0 \\ 1 \end{pmatrix} = \begin{pmatrix} 15 \\ 1 \end{pmatrix}.$$

$$P_l = \begin{pmatrix} A\hat{x}^l \\ 1 \end{pmatrix}: \; P_1 = \begin{pmatrix} (1,1)\begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \quad P_2 = \begin{pmatrix} (1,1)\begin{pmatrix} 0 \\ 9 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 9 \\ 1 \end{pmatrix};$$

$$P_3 = \begin{pmatrix} (1,1)\begin{pmatrix} 17 \\ 60 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 77 \\ 1 \end{pmatrix}; \qquad P_4 = \begin{pmatrix} (1,1)\begin{pmatrix} 2 \\ 0 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Координирующая задача имеет вид:

$$\begin{gathered} 0\beta_1 + 27\beta_2 + 214\beta_3 + 4\beta_4 \to \max \\ 0\beta_1 + 9\beta_2 + 77\beta_3 + 2\beta_4 \le 15 \\ \beta_1 + \beta_2 + \beta_3 + \beta_4 = 1 \\ \beta_j \ge 0, \quad (j = \overline{1,4}). \end{gathered}$$

Какие можно сделать выводы?

Если сравнить исходную и главную задачи, можно заметить, что исходная задача имеет 3 ограничения и 2 переменные, а главная – 2 ограничения и 4 переменные. В принципе, уменьшение количества ограничений – это хорошо, так как не раз подчеркивалось, что методы ЛП весьма чувствительны к именно к количеству ограничений.

Но количество переменных возросло. Потребовалось вычислить координаты всех вершин многоугольника, соответствующего диагональному блоку.

Рассматриваемый метод декомпозиции[7] ориентирован на решение задач очень большой размерности.

Естественно, что для вычисления координат всех вершин требуется нереализуемая мощность ЭВМ. Вулф и Данциг – авторы метода декомпозиции – доказали, что решение главной задачи можно искать не зная ни всех вершин многогранника, определенного диагональным блоком, ни количества этих вершин.

Достаточно знать лишь один (любой) опорный план главной задачи, а также базисную и обратную к ней матрицы соответствующего опорного решения.

При этом (что самое важное в методе) координирующая задача в явном виде нигде не записывается[8].

Остановимся на этом методе подробнее.

1.4.6. Обоснование метода декомпозиции Данцига-Вулфа

Итак, представим себе главную задачу:

$$\left.\begin{aligned} &\sum_{l=1}^{k} c_l \beta_l \to \max \\ &\sum_{l=1}^{k} P_l \beta_l = P_0 \\ &\beta_l \ge 0, \quad (l = \overline{1,k}) \end{aligned}\right\} \text{Здесь } c_l = c\widehat{x}^l;\ P_l = \begin{pmatrix} A\widehat{x}^l \\ 1 \end{pmatrix};$$

$$P_0 = \begin{pmatrix} B_0 \\ 1 \end{pmatrix} \left.\vphantom{\begin{pmatrix} B_0 \\ 1 \end{pmatrix}}\right\} r_0 + 1$$

Пусть:

B – базисная матрица некоторого опорного решения этой задачи; B^{-1}– обратная матрица; $C_{баз}$ – вектор коэффициентов ЦФ при базисных переменных.

В соответствии с вычислительной схемой модифицированного симплекс-метода[9] (МСМ) текущее решение является оптималь-

---

[7] Непосредственно о декомпозиции речь пока не идет.

[8] Достаточно представить (мысленно), что эта задача есть (есть не в памяти ЭВМ, а вообще есть!).

[9] Почему МСМ? Дело в том, что именно в рамках схемы МСМ можно не знать коэффициентов разложения тех векторов, чьи оценки нужно вычислить.

ным, если для всех небазисных векторов $P_l \quad (l=\overline{1,k})$ оценки этих векторов не отрицательны:

$$\Delta_l = c_{баз}B^{-1}P_l - c_l = c_{баз}B^{-1}P_l - c\hat{x}^l \geq 0. \tag{1.95}$$

Начнем анализ этого выражения на предмет исследования возможности его практического использования.

Рассмотрим произведение $B^{-1}P_l$:[10]

$$B^{-1}P_l = \left[\begin{array}{c|c} \begin{matrix} R_0 \\ матрица \\ размерностью \\ (r_0+1)\times r_0 \\ r_0\ столбцов \end{matrix} & \begin{matrix} V \\ \\ (r_0+1)-й \\ столбец \end{matrix} \end{array}\right] \times \begin{bmatrix} A\hat{X}^l \\ \\ 1 \end{bmatrix} = \tag{1.96}$$

$$= R_0 A\hat{x}^l + V$$

Подставим (1.96) в (1.95):

$$\Delta_l = C_{баз}(R_0 A\hat{x}^l + V) - C\hat{x}^l \ \text{ – оценка вектора } P_l = \begin{pmatrix} A\hat{x}^l \\ 1 \end{pmatrix}$$

или

$$\Delta_l = (C_{баз}R_0 A - C)\hat{x}^l + C_{баз}V. \tag{1.97}$$

Известно, что если решение не оптимальное, вектор P_l, которому соответствует отрицательная оценка (желательно, максимальная по абсолютной величине), должен быть включен в состав базисных векторов.

Проблема же в том, что Δ_l нельзя вычислить до тех пор, пока не известны все элементы вектора $P_l = \begin{pmatrix} A\hat{x}^l \\ 1 \end{pmatrix}$.

В свою очередь, численное определение этих элементов невозможно до тех пор, пока не известна соответствующая экстремальная точка $\hat{x}$.

То есть нужно сначала определить эту экстремальную точку. А как ее определить, если мы знаем только обратную матрицу

[10] $B^{-1}P_l$ -это вектор коэффициентов разложения вектора P_l по базису.

главной задачи B^{-1} и $C_{баз}$. Сама же задача задана неявно. Ее просто нет!

Идея метода Данцига-Вулфа заключается в том, чтобы *заменить поиск отрицательной оценки путем последовательного перебора свободных векторов текущего опорного решения направленным поиском только одной экстремальной точки, которой соответствует минимальная оценка связанного с этой точкой вектора главной задачи.*

Если окажется, что найденная таким образом минимальная оценка неотрицательна, задача решена. Точнее, решена главная задача, а следовательно, и исходная.

В противном случае ясно, какой вектор следует вводить в базис: этот вектор будет полностью определен координатами найденной экстремальной точки.

Итак, нужно найти экстремальную точку, которой соответствует вектор главной задачи, имеющий минимальную оценку.

Это можно сделать, решив следующую задачу:

$$\Delta \to \min$$

$$DX = B^*$$

$$x \geq 0.$$

Здесь следует обратить внимание на то, что индекс номера вершины отсутствует. Этот индекс просто неизвестен, так как соответствующую вершину еще нужно найти.

В качестве переменных этой задачи выступают координаты искомой вершины.

Мы предположили, что множество, определенное ограничениями $DX = B^*, x \geq 0$, ограничено. Следовательно, минимальное значение Δ также ограничено и должно соответствовать некоторой экстремальной точке. Ее-то и нужно найти.

Уже было показано (1.97), что:

$$\Delta_l = (C_{баз} R_0 A - C)\hat{x} + C_{баз} V.$$

Теперь, опустив индекс l, имеем:

$$\Delta = (C_{баз} R_0 A - C)X + C_{баз} V. \tag{1.98}$$

Ввиду того, что $C_{баз} V$ – постоянная величина, не зависящая от X, задаче ЛП можно придать следующий вид:

$$z=(C_{баз}R_0A-C)X\to\min$$
$$DX=B^* \qquad (1.99)$$
$$x\geq 0.$$

Назовем эту задачу <u>*частной*</u> (или <u>*локальной*</u>).

Допустим, мы решили эту задачу[11], и z^* – оптимальное значение ЦФ. Отсюда узнаем и минимальную оценку:

$$\Delta_{\min}=z^*+C_{баз}V$$

и, что самое главное, и сам свободный вектор, который обладает данной оценкой. Действительно, мы знаем координаты экстремальной точки $\widehat{x}^l$, на которой ЦФ локальной задачи имеет значение z^***.** Следовательно, можно сформировать вектор главной задачи, соответствующий этой точке:

$$P_l=\begin{pmatrix}A\widehat{x}^l\\1\end{pmatrix}.$$

Здесь l – это не номер вершины, а номер итерации, на которой данная вершина найдена.

Теперь, если $\Delta_{\min}=z^*+C_{баз}V\geq 0$, то главная задача решена. Исследуемое на текущей итерации решение – оптимальное. Можно восстановить решение исходной задачи. Делается это следующим образом.

Пусть σ^*– множество номеров базисных переменных главной задачи.

Тогда, по найденным координатам соответствующих экстремальных точек $\widehat{x}^j$ $(j\in\sigma^*)$ можно восстановить решение исходной задачи:

$$X_{опт}=\sum_{j\in\sigma^*}\beta_j\widehat{x}^j.$$

[11] Другого, в предположении, что допустимое множество ограничено, не может быть.

Если $\Delta_{\min} < 0$, то вектор $P_l = \begin{pmatrix} A\widehat{x}^l \\ 1 \end{pmatrix}$ нужно ввести в базис нового опорного решения главной задачи. Делается это по правилам МСМ.

Следует обратить внимание, что используются общие принципы обычного симплекс-метода: последовательный переход от одного опорного решения координирующей задачи к другому, лучшему. Но при каждом таком переходе решается локальная задача для поиска вектора с минимальной оценкой. При этом, на каждой итерации формируется частная ЦФ локальной задачи[12].

Смысл метода декомпозиции заключается в том, что при решении локальной задачи:

$$z = (C_{баз} R_0 A - C)X \to \min$$

$$DX = B^*$$

$$x \geq 0$$

мы сталкиваемся с чисто диагональной структурой ее матрицы ограничений (без блока-связки).

Система ограничений $DX = B^*$ – это сокращенная запись следующей системы ограничений:

$$\begin{array}{lllll}
\sum\limits_{j \in \omega_1} D_j x_j & & & & = B_1 \\
& \sum\limits_{j \in \omega_2} D_j x_j & & & = B_2 \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \\
& & \sum\limits_{j \in \omega_\nu} D_j x_j & & = B_\nu \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \\
& & & \sum\limits_{j \in \omega_L} D_j x_j & = B_L
\end{array}$$

Такая структура допускает простое разделение задачи на L подзадач меньшей размерности. Сборка же решения осуществляется следующим образом.

[12] Коэффициенты ЦФ локальной задачи определяются через обратную матрицу очередного опорного решения главной задачи.

Пусть $z'_j\,(j=\overline{1,L})$ – оптимальное значение ЦФ j-й подзадачи, а $(x'_{q_{j-1}+1}, x'_{q_{j-1}+2}, ..., x'_{q_j})$ – ее оптимальное решение.

Тогда оптимальное решение локальной задачи определяется:

$$x_{опт} = (x'_1, ..., x'_{q_1}, x'_{q_1+1}, ..., x'_{q_j}, ..., x'_{q_{L-1}+1}, ..., x'_{q_n}),$$

$$Z_{опт} = \sum_{j=\overline{1,L}} z'_j .$$

1.4.7. Алгоритм метода декомпозиции Данцига-Вулфа

Дана задача ЛП, ограничения которой составляют блок-связку и ряд диагональных блоков.

При решении координирующей задачи будем считать, что имеем один диагональный блок. При решении же частных задач этот блок будет разделяться на соответствующие диагональные блоки.

Т.е. задача имеет вид:

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

$$AX = B_0$$

$$DX = B^*$$

$$x \geq 0 -$$

сохраняем предположение, что допустимое множество задачи не пусто и ЦФ на нем ограничена сверху.

Шаг 1. "Построить" координирующую задачу. Конечно же, условно: мы не в состоянии этого сделать. Однако структуру этой задачи мы можем представить следующим образом:

$$\begin{cases} \sum_{l=1}^{?} C\widehat{x}^l \beta_l \to \max \\ \sum_{l=1}^{?} A\widehat{x}^l \beta_l = B_0 \\ \sum_{l=1}^{?} \beta_l = 1 \\ \beta_l \geq 0, \quad (l = \overline{1,?}). \end{cases}$$

Условность этой записи связана еще с одним обстоятельством. Дело в том, что в исходной формулировке ограничения задачи могут быть представлены и уравнениями-ограничениями и нестрогими неравенствами любого направления. Работа же по методу декомпозиции начинается с известного опорного решения координирующей задачи, имеющего, как правило, единичный базис. Для того чтобы получить такое решение, при сведении координирующей задачи к каноническому виду (с полным набором единичных векторов) в нее вводятся дополнительные и/или искусственные переменные. Это привносит некоторую специфику при реализации метода декомпозиции. При этом координирующую задачу схематически можно представить так (рис.1.22):

$$\begin{cases} \sum\limits_{l=1}^{?} C\widehat{X}^{l}\beta_l + \text{XXXXXXX} \to \max, \\ \sum\limits_{l=1}^{?} A\widehat{X}^{l}\beta_l + \text{XXXXXXX} = B_0, \\ \sum\limits_{l=1}^{?} \beta_l + \qquad \text{XXXXXXX} = 1, \\ \beta_l \ge 0, \ (l = \overline{1,?}) \ \text{XXXXXXX}. \end{cases}$$

Область системы ограничений, по которой строится исходное опорное решение

Рис.1.22. Схематическое представление координирующей задачи

Переменные β_l, $(l = \overline{1,?})$ будем называть *<u>основными переменными</u>* задачи. Все остальные – дополнительные и/или искусственные - *<u>вспомогательными</u>*.

<u>Шаг 2</u>. Найти исходное опорное решение координирующей задачи. Обычно для этого используется метод *M*-задачи: в выделенной области ограничений соответствующие векторы составляют полный единичный базис. Здесь важно учесть, что теперь методом декомпозиции будет решаться координирующая задача, имеющая вид *М*-задачи! То есть исходное опорное решение, а также некоторые промежуточные решения будут содержать искусственные переменные в составе базисных[13].

Чисто технически, будем использовать следующий прием. Допустим, имеется базис некоторого опорного решения:

$$B = (P_{i_1}, P_{i_2}, ..., P_{r_0+1}),$$

где $\boldsymbol{r_0}+1$ - количество ограничений-уравнений в координирующей задаче.

Как уже отмечалось, среди базисных векторов могут быть как основные, так и вспомогательные.

Опорному решению поставим в соответствие массив из $\boldsymbol{r_0}+1$ элементов $Q = (q_1, q_2, ..., q_{r_0+1})$ таких, что:

$$q_k = \begin{cases} \widehat{X}^l, \textit{если вектор } P_{i_k} \textit{ занимает } k\textit{-ю позицию в базисной мат}- \\ \textit{рице и является основным}; \\ 0, \textit{ если вектор } P_{i_k} - \textit{вспомогательный.} \end{cases}$$

БАЗ	Q			
	1	2	...	n
P_{i_1}	q_1			
P_{i_2}	q_2			
P_{i_k}	q_k			
$P_{i_{r_0}+1}$	q_{r_0+1}			

[13] Для каждого опорного решения координирующей задачи необходимо будет запоминать не только значения базисных переменных, но и координаты вершин, связанных с основными переменными β, если эти переменные входят в состав базисных. Проблема в том, что номеров у этих номеров переменных просто нет и быть не может. Поэтому при обозначении этих переменных будем указывать номер итерации, на которой соответствующая переменная вошла в базис.

Здесь 0=(0,0,...,0) – n-мерный вектор.

Итак, имеем опорное решение:

B – базисная матрица опорного решения;

B^{-1} – обратная матрица;

$C_{баз}$ – вектор коэффициентов ЦФ при базисных переменных.

Вычисляем $X(P_0)=B^{-1}P_0$ и элементы массива $Q=(q_1,q_2,...,q_{r_0+1})$. Заметим, что для исходного опорного решения все элементы этого массива – нулевые, так как мы еще не знаем ни одной вершины, и базисная матрица состоит из одних вспомогательных векторов.

Полагаем $l=1$ (номер первой итерации).

Шаг 3. Формируем частную задачу:

$$Z=\left(C_{баз}R_0A-C\right)X\to\min$$

$$DX=B^*$$

$$x\geq 0.$$

Решаем эту задачу (если D имеет блочно-диагональную структуру, эта задача разделяется на подзадачи). Собираем решение.

z^* – оптимальное значение ЦФ;

$\widehat{x}^l$ – вектор координат экстремальной точки.

Вычисляем оценку: $\Delta_{\min}=z^*+C_{баз}V=\Delta_l$.

Если $\Delta_l\geq 0$, переходим к шагу 6. В противном случае выполняем следующий шаг.

Шаг 4. Находим вектор P_l, которому соответствует найденная отрицательная оценка:

$$P_l=\begin{pmatrix}A\widehat{x}^l\\1\end{pmatrix}.$$

Шаг 5.

Ищем разложение этого вектора:

$$X(P_l)=B^{-1}P_l.$$

По обычному для симплекс-метода правилу находим вектор, который нужно вывести из базиса:

$$\min_{x_{kl}>0}\left\{\frac{x_{k0}}{x_{kl}}\right\}=\frac{x_{r0}}{x_{rl}}.$$

Корректируем вектор $c_{баз}$. Если вводимый вектор – основной, принимаем $c_{i_r}=c\widehat{x}^l$.

Корректируем массив $Q=(q_1,q_2,...,q_{r_0+1})$: ***r***-му элементу q_r ставим в соответствие вектор координат найденной вершины $\widehat{x}^l$ или (*0,0,...,0*) в зависимости от того, является ли вводимая в состав базисных переменная основной или вспомогательной.

Формируем новую обратную матрицу $B^{-1}_{нов}$ (известно разложение вектора P_l по старому базису, известен ведущий элемент):

$$B^{-1}_{нов}=\tilde{E}B^{-1},$$

где $\tilde{E}=\left\|\begin{matrix}1 & 0 & \cdots & -\dfrac{x_{1s}}{x_{rs}} & \cdots & 0\\ 0 & 1 & \cdots & -\dfrac{x_{2s}}{x_{rs}} & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & 0\\ 0 & 0 & \cdots & \dfrac{1}{x_{rs}} & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots & \cdots & 0\\ 0 & 0 & \cdots & -\dfrac{x_{r_0+1,s}}{x_{rs}} & \cdots & 1\end{matrix}\right\|$ – элементарная матрица.

Находим новое разложение вектора P_0 по базису:

$$X(P_0)=B^{-1}_{нов}P_0.$$

Положив $l=l+1$, переходим к шагу 3.

<u>Шаг 6</u>. Все оценки у основных векторов неотрицательны. Необходимо проверить оценки вспомогательных векторов.

Вычисление этих оценок проводим по обычным правилам:

$$\Delta_j=C_{баз}B^{-1}A_j-C_j.$$

Если обнаруживается вспомогательный вектор с отрицательной оценкой (A_s), переходим к шагу 5. При этом в качестве P_l принимаем A_s. В противном случае выполняем следующий шаг.

Шаг 7. Получено оптимальное решение координирующей задачи:

$X(P_0)=(x_{10},x_{20},...,x_{r_0+1,0})$ – значения базисных переменных.

Известен массив $Q=(q_1,q_2,...,q_{r_0+1})$.

Восстанавливается решение исходной задачи:

$X_{опт}=\sum_{i=1}^{r_0+1} q_i x_{i0}$. Это соответствует $X_{опт}=\sum_{i=1}^{k} \beta_i \widehat{x}^i$. Конец.

Пример 1.27

Необходимо решить задачу линейного программирования, состоящую из двух диагональных блоков.

$$
\begin{array}{lllll}
7x_1 & +15x_2 & +7x_3 & +10x_4 & \to \max \\
3x_1 & +5x_2 & +x_3 & +x_4 & =2 \\
x_1 & +x_2 & & & \le 2 \\
x_1 & +2x_2 & & & \le 1 \\
 & & x_3 & +x_4 & \le 3 \\
 & & x_3 & +2x_4 & \le 1
\end{array}
$$

$x_j \ge 0 (j=1..4).$

Координирующая задача состоит из двух ограничений. Используя метод M-задачи, построим исходную таблицу модифицированного симплекс-метода координирующей задачи (искусственные векторы назовем I_1 и I_2) . Учитывая особенности метода декомпозиции, дополним таблицу элементами массива Q и опустим элементы вектора симплексных множителей (за ненадобностью).

Баз	*$C_{баз}$*	*A_0*	*E_1*	*E_2*	*Q*			
					1	*2*	*3*	*4*
I_1	*-M*	*2*	*1*	*0*	*0*	*0*	*0*	*0*
I_2	*-M*	*1*	*0*	*1*	*0*	*0*	*0*	*0*
Табл.1		*-3M*						

Используя выражение (4), определим коэффициенты целевой функции частной задачи:

$$\begin{pmatrix}-M & -M\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}3 & 5 & 1 & 1\end{pmatrix}-\begin{pmatrix}7 & 15 & 7 & 10\end{pmatrix}=$$

$$=\begin{pmatrix}-3M-7 & -5M-15 & -M-7 & -M-10\end{pmatrix}.$$

Теперь можно построить саму частную задачу:

$$\begin{array}{lllll}
(-3M-7)x_1 & +(-5M-15)x_2 & +(-M-7)x_3 & +(-M-10)x_4 & \to \min \\
x_1 & +x_2 & & & \le 2 \\
x_1 & +2x_2 & & & \le 1 \\
 & & x_3 & +x_4 & \le 3 \\
 & & x_3 & +2x_4 & \le 1
\end{array}$$

$x_j \ge 0 (j = 1..4).$

За счет диагональной структуры, частная задача поддается декомпозиции на две независимые подзадачи:

$$\begin{array}{ll}
(-3M-7)x_1 + (-5M-15)x_2 \to \min & (-M-7)x_3 + (-M-10)x_4 \to \min \\
x_1 + x_2 \le 2 & x_3 + x_4 \le 3 \\
x_1 + 2x_2 \le 1 & x_3 + 2x_4 \le 1 \\
x_j \ge 0 (j = 1..4) & x_j \ge 0 (j = 1..4).
\end{array}$$

Решим первую подзадачу простым симплекс-методом (следует заметить, что в этом примере мы будем придерживаться нумерации векторов с единицы, в порядке следования соответствующих им переменных в задаче, т.е. номер вектора не будет совпадать с индексом переменной):

Баз	$C_{баз}$	A_0	*-3M-7*	*-5M-15*	*0*	*0*
			A_1	A_2	A_3	A_4
A_3	*0*	*2*	*1*	*1*	*1*	*0*
A_4	*0*	*1*	*1*	*2*	*0*	*1*
Табл. 1		*0*	*3M+7*	*5M+15*	*0*	*0*
A_3	*0*	*1*	*0*	*-1*	*1*	*-1*
A_1	*-3M-7*	*1*	*1*	*2*	*0*	*1*
Табл. 2		*-3M-7*	*0*	*-M+1*	*0*	*-3M-7*

Решение задачи: $x^1 = (x_1^1, x_2^1) = (1,0); z^1 = -3M - 7.$

Теперь решим вторую задачу:

Баз	$C_{баз}$	A_0	$-M-7$	$-M-10$	0	0
			A_1	A_2	A_3	A_4
A_3	0	3	1	1	1	0
A_4	0	1	1	2	0	1
Табл. 1		0	$M+7$	$M+10$	0	0
A_3	0	2	0	-1	1	-1
A_1	$-M-7$	1	1	2	0	1
Табл. 2		$-M-7$	0	$-M-4$	0	$-M-7$

Решение задачи: $x^2 = (x_3^2, x_4^2) = (1,0); z^2 = -M - 7.$

Теперь объединим решения обеих задач в один вектор, чтобы получить полное решение частной задачи: $x^* = (x_1^1, x_2^1, x_3^2, x_4^2) = (1,0,1,0); z^* = z^1 + z^2 = -4M - 14$. По сути дела, вектор $P_1 = \begin{pmatrix} Ax^* \\ 1 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 3 & 5 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$ – это вектор координирующей задачи, обладающий минимальной оценкой. Вычислим теперь её значение:

$$\Delta_{\min} = z^* + C_{баз}V = -4M - 14 + \begin{pmatrix} -M & -M \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -5M - 14 < 0.$$

Коэффициент целевой функции перед вектором P_1 равен

$$c^* = cx^* = \begin{pmatrix} 7 & 15 & 7 & 10 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = 14.$$

Поскольку минимальная оценка отрицательна, вектор P_1 следует ввести в базис:

		P_1	4	1		Q			
Баз	$C_{баз}$	A_0	E_1	E_2	14				
					$B^{-1}P_1$	1	2	3	4
I_1	$-M$	2	1	0	4	0	0	0	0
I_2	$-M$	1	0	1	1	0	0	0	0
Табл.1		$-3M$							

					Q			
Баз	$C_{баз}$	A_0	E_1	E_2	*1*	*2*	*3*	*4*
P_1	*14*	*1/2*	*1/4*	*0*	*1*	*0*	*1*	*0*
I_2	*-M*	*1/2*	*-1/4*	*1*	*0*	*0*	*0*	*0*
Табл.2		*-3M*						

Вычисляем коэффициенты целевой функции частной задачи:

$$\begin{pmatrix}14 & -M\end{pmatrix}\begin{pmatrix}1/4 \\ -1/4\end{pmatrix}\begin{pmatrix}3 & 5 & 1 & 1\end{pmatrix}-\begin{pmatrix}7 & 15 & 7 & 10\end{pmatrix}=$$

$$=\begin{pmatrix}\frac{3}{4}M+\frac{7}{2} & \frac{5}{4}M+\frac{5}{2} & \frac{M}{4}-\frac{7}{2} & \frac{M}{4}-\frac{13}{2}\end{pmatrix}.$$

Учитывая, что в частной задаче ведется поиск минимума целевой функции, а также тот факт, что среди всех коэффициентов целевой функции присутствует M с положительным коэффициентом, частная задача имеет очевидное минимальное решение:

$x^* = (0,0,0,0); z^* = 0$, которое соответствует вектору $P_2 = \begin{pmatrix}\begin{pmatrix}3 & 5 & 1 & 1\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\end{pmatrix} \\ 1\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$ координирующей задачи. Его коэффициент целевой функции $c^* = \begin{pmatrix}7 & 15 & 7 & 10\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\end{pmatrix} = 0$, оценка $\Delta_{\min} = 0 + \begin{pmatrix}14 & -M\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = -M < 0$. Введем этот вектор в базис координирующей задачи:

		P_2	0	1		Q			
Баз	$C_{баз}$	A_0	E_1	E_2	0				
					$B^{-1}P_2$	1	2	3	4
P_1	14	1/2	1/4	0	0	1	0	1	0
I_2	-M	1/2	-1/4	1	1	0	0	0	0
Табл.2		-3M							

					Q			
Баз	$C_{баз}$	A_0	E_1	E_2	1	2	3	4
P_1	14	1/2	1/4	0	1	0	1	0
P_2	0	1/2	-1/4	1	0	0	0	0
Табл.3		7						

Для данной симплекс-таблицы коэффициенты целевой функции частной задачи следующие:

$$(14 \quad 0)\begin{pmatrix} 1/4 \\ -1/4 \end{pmatrix}(3 \quad 5 \quad 1 \quad 1)-(7 \quad 15 \quad 7 \quad 10)=\begin{pmatrix} \frac{7}{2} & \frac{5}{2} & -\frac{7}{2} & -\frac{13}{2} \end{pmatrix}.$$

Частная задача поддается декомпозиции на две задачи. Первая задача имеет очевидное оптимальное решение $x^1 = (0,0); z^1 = 0$.

Решим вторую частную задачу.

Баз	$C_{баз}$	A_0	-7/2	-13/2	0	0
			A_3	A_2	A_3	A_4
A_3	0	3	1	1	1	0
A_4	0	1	1	2	0	1
Табл. 1		0	7/2	13/2	0	0
A_3	0	2	0	-1	1	-1
A_1	-7/2	1	1	2	0	1
Табл. 2		-7/2	0	-1/2	0	-7/2

Оптимальное решение задачи $x^2 = (1,0); z^2 = -7/2$. Таким образом, оптимальное решение частной задачи $x^* = (0,0,1,0); z^* = -7/2$. Это решение соответствует вектору

$$P_3 = \begin{pmatrix} \begin{pmatrix} 3 & 5 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$ координирующей задачи,

$$c^* = \begin{pmatrix} 7 & 15 & 7 & 10 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = 7$$, оценка

$\Delta_{\min} = -7/2 + \begin{pmatrix} 14 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -7/2 < 0$. Произведем шаг симплекс-метода координирующей задачи:

		P_3	*1*	*1*		*Q*			
БАЗ	$C_{баз}$	A_0	E_1	E_2	*7*				
					$B^{-1}P_3$	*1*	*2*	*3*	*4*
P_1	*14*	*1/2*	*1/4*	*0*	*1/4*	*1*	*0*	*1*	*0*
P_2	*0*	*1/2*	*-1/4*	*1*	*3/4*	*0*	*0*	*0*	*0*
Табл.3		*7*							

					Q			
БАЗ	$C_{баз}$	A_0	E_1	E_2	*1*	*2*	*3*	*4*
P_1	*14*	*1/3*	*1/3*	*-1/3*	*1*	*0*	*1*	*0*
P_3	*7*	*2/3*	*-1/3*	*4/3*	*0*	*0*	*1*	*0*
Табл.4		*28/3*						

Вычислим значение коэффициентов целевой функции частной задачи

$$\begin{pmatrix} 14 & 7 \end{pmatrix} \begin{pmatrix} 1/3 \\ -1/3 \end{pmatrix} \begin{pmatrix} 3 & 5 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 7 & 15 & 7 & 10 \end{pmatrix} = \begin{pmatrix} 0 & -10/3 & -14/3 & -\frac{23}{3} \end{pmatrix}$$

Решим первую подзадачу:

Баз	$C_{баз}$	A_0	*0*	*-10/3*	*0*	*0*
			A_1	A_2	A_3	A_4
A_3	*0*	*2*	*1*	*1*	*1*	*0*
A_4	*0*	*1*	*1*	*2*	*0*	*1*
Табл. 1		*0*	*0*	*10/3*	*0*	*0*
A_3	*0*	*3/2*	*1/2*	*0*	*1*	*-1*
A_2	*-10/3*	*1/2*	*1/2*	*1*	*0*	*1*
Табл. 2		*-5/3*	*-5/3*	*0*	*0*	*-10/3*

Оптимальное решение задачи $x^1 = (0, \frac{1}{2}); z^1 = -\frac{5}{3}$. Теперь решим вторую подзадачу:

Баз	$C_{баз}$	A_0	*-14/3*	*-23/3*	*0*	*0*
			A_3	A_2	A_3	A_4
A_3	*0*	*3*	*1*	*1*	*1*	*0*
A_4	*0*	*1*	*1*	*2*	*0*	*1*
Табл. 1		*0*	*14/3*	*23/3*	*0*	*0*
A_3	*0*	*2*	*0*	*-1*	*1*	*-1*
A_1	*-14/3*	*1*	*1*	*2*	*0*	*1*
Табл. 2		*-14/3*	*0*	*-5/3*	*0*	*-14/3*

Оптимальное решение задачи $x^2 = (1, 0); z^2 = -\frac{14}{3}$. Таким образом, оптимальное решение всей частной задачи $x^* = (0, \frac{1}{2}, 1, 0); z^* = -\frac{19}{3}$, что соответствует вектору координирующей задачи $P_4 = \begin{pmatrix} \begin{pmatrix} 3 & 5 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{2} \\ 1 \\ 0 \end{pmatrix} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{7}{2} \\ 1 \end{pmatrix}$,

$$c^* = \begin{pmatrix} 7 & 15 & 7 & 10 \end{pmatrix} \begin{pmatrix} 0 \\ 1/2 \\ 1 \\ 0 \end{pmatrix} = \frac{29}{2},$$

оценка $\Delta_{\min} = -19/3 + \begin{pmatrix} 14 & 7 \end{pmatrix} \begin{pmatrix} -1/3 \\ 4/3 \end{pmatrix} = -5/3 < 0$. Вводим в базис координирующей задачи вектор P_4 :

		P_4	7/2	1		Q			
БАЗ	$C_{баз}$	A_0	E_1	E_2	29/2				
					$B^{-1}P_4$	1	2	3	4
P_1	14	1/3	1/3	-1/3	5/6	1	0	1	0
P_3	7	2/3	-1/3	4/3	1/6	0	0	1	0
Табл.3		28/3							

					Q			
БАЗ	$C_{баз}$	A_0	E_1	E_2	1	2	3	4
P_4	29/2	2/5	2/5	-2/5	0	1/2	1	0
P_3	7	3/5	-2/5	7/5	0	0	1	0
Табл.4		10						

Для данной симплекс-таблицы коэффициенты целевой функции частной задачи следующие:

$$\begin{pmatrix} 29/2 & 7 \end{pmatrix} \begin{pmatrix} 2/5 \\ -2/5 \end{pmatrix} \begin{pmatrix} 3 & 5 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 7 & 15 & 7 & 10 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -4 & -7 \end{pmatrix}.$$

Первая подзадача имеет очевидное оптимальное решение $x^1 = (0,0); z^1 = 0$. Решим вторую подзадачу.

Баз	$C_{баз}$	A_0	-4	-7	0	0
			A_3	A_2	A_3	A_4
A_3	0	3	1	1	1	0
A_4	0	1	1	2	0	1
Табл. 1		0	-4	-7	0	0
A_3	0	2	0	-1	1	-1
A_1	-4	1	1	2	0	1
Табл. 2		-4	0	-1	0	-4

Оптимальное решение подзадачи $x^2 = (1,0); z^2 = -4$. Оптимальное решение частной задачи $x^* = (0,0,1,0); z^* = -4$. Минимальная оценка векторов координирующей задачи $\Delta_{\min} = -4 + \begin{pmatrix} 29/2 & 7 \end{pmatrix} \begin{pmatrix} -2/5 \\ 7/5 \end{pmatrix} = 0$. Это означает, что в последней симплекс-таблице записано оптимальное решение координирующей задачи $\beta = (2/5 \quad 3/5)$, и соответствующие координаты вершин блочно-диагонального множества, определяющие базис оптимального решения (в структуре Q) – $\hat{X}_1 = (0, 1/2, 1, 0)$ и $\hat{X}_2 = (0, 0, 1, 0)$. Восстанавливаем оптимальное решение исходной задачи:

$$X_{опт} = \sum_{i=1}^{k} \beta_i \widehat{X}^i = \frac{2}{5}\begin{pmatrix} 0 & 1/2 & 1 & 0 \end{pmatrix} + \frac{3}{5}\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{5} & 1 & 0 \end{pmatrix}.$$

Значение целевой функции оптимального решения исходной задачи совпадает со значением целевой функции координирующей задачи и равно 10.

1.4.8. Анализ чувствительности линейных моделей

Возможности, предоставляемые теорией линейного программирования, не ограничиваются лишь получением оптимальных значений управляющих переменных. Решение задач линейного программирования должно обеспечивать пользователя динамической информацией.

Дело в том, что как только условия, в соответствии с которыми была построена модель задачи, изменяются, информация, ассоциированная со статическим решением, теряет свою актуальность.

Параметрическое программирование, как раздел линейного программирования, представляет собой метод исследования изменений оптимального решения задачи линейного программирования, обусловленных заранее задаваемыми непрерывными изменениями параметров линейной модели.

Очень ценные результаты по анализу изменений оптимального решения задачи линейного программирования, вызванных изменениями параметров этой задачи, связаны с теорией двойственности.

В разделе 1.3.3 было показано, что в процессе решения задачи единичная матрица, записанная в первой симплекс-таблице, преобразуется в обратную базисную матрицу оптимального решения задачи.

Первая таблица ⇒

Баз	$C_{баз}$	A_0	....	
			....	***E*** **– единичная базисная матрица**

....

Оптимальная таблица ⇒

Баз	$C_{баз}$	A_0		
A_{i_1} A_{i_2} ... A_{i_m}	c_{i_1} c_{i_2} ... c_{i_m}	x_{10} x_{20} ... x_{m0}		$\boldsymbol{B^{-1}}$ **– обратная матрица**

Это обстоятельство иногда позволяет не решать заново задачу в случае изменения ее параметров.

Действительно, рассмотрим известное выражение :

$$X_0=B^{-1}A_0. \tag{1.100}$$

Очевидно, что это выражение можно использовать для анализа влияния изменений в правой части системы ограничений на допустимость решения, связанного с последним базисом. На оптимальность же полученного решения изменение коэффициентов вектора A_0 повлиять не может, так как признаком оптимальности является неотрицательность оценок небазисных векторов, которые определяются по формуле:

$$\Delta_j= c_{баз}\, B^{-1}A_j - c_j. \tag{1.101}$$

Таким образом, если оказалось, что изменение коэффициентов вектора A_0 не привело к недопустимости нового решения (среди коэффициентов вектора X_0, вычисленного по формуле (1.100), нет отрицательных), X_0 является оптимальным решением задачи с новым вектором A_0. Этому решению будет соответствовать новое значение целевой функции, которое можно вычислить по формуле:

$$Z_0=c_{баз}X_0= c_{баз}B^{-1}A_{0.} \qquad (1.102)$$

Если же среди коэффициентов вектора X_0, вычисленного по формуле (1.100), имеются отрицательные, то достаточно:

- обновить столбец A_0 последней симплекс-таблицы;
- записать в нее новое значение целевой функции;
- продолжить решение задачи двойственным симплекс-методом (решение X_0 является псевдопланом, так как оценки векторов не изменились – остались неотрицательными, а в столбце A_0 появились отрицательные коэффициенты).

Рассмотрим теперь выражение (1.101). Это выражение можно использовать:

1. Для анализа влияния изменений коэффициентов целевой функции на оптимальность полученного решения;
2. Для анализа влияния изменений коэффициентов левой части системы ограничений на оптимальность полученного решения (векторов A_j, $j=1,2,...n$).

Следует, однако, отметить, что во втором случае есть определенные проблемы. Так, например, если A_j входит в оптимальный базис, то исследование на чувствительность просто провести нельзя, так как для этого нужно пересчитать обратную матрицу – проще решить задачу заново. Поэтому этот случай, как правило, не рассматривается при исследовании задач линейного программирования на чувствительность.

При анализе влияния изменений коэффициентов целевой функции на оптимальность полученного решения анализ проводится по следующей схеме:

- Пересчитываются оценки всех векторов с использованием выражения (1.101);

- Если оценки остаются неотрицательными, то X_0 – оптимальное решение задачи. Этому решению соответствует новое значение целевой функции, вычисленное по формуле (1.102);
- Если среди новых оценок появилась отрицательная, то все новые оценки записываются в симплекс-таблицу, по формуле (1.102) вычисляется новое значение целевой функции, обновляется столбец $C_{баз}$, и с использованием симплекс-метода продолжается решение задачи.

1.4.9. Общая схема параметрического анализа задач линейного программирования

В параметрическом программировании, в основном, решаются следующие задачи.

Задача А. Дана задача линейного программирования:

$$< c, x > \to \max$$

$$\left.\begin{array}{r} AX = A_0(t) \\ x \geq 0 \end{array}\right\} D(t), \tag{1.103}$$

где $D(t)$ – множество допустимых решений задачи; $A_0(t) = (a_1(t), a_2(t), ..., a_m(t))^T$ – вектор-функция.

Необходимо определить функцию

$$\Phi(t) = \max_{x \in D(t)} < c, x >$$

Задача В. Дана задача линейного программирования:

$$< c(t), x > \to \max$$

$$\left.\begin{array}{r} AX = A_0 \\ x \geq 0 \end{array}\right\} D. \tag{1.104}$$

где D – множество допустимых решений задачи; $c(t) = \left(c_1(t), c_2(t), ..., c_n(t)\right)$ – вектор-функция.

Необходимо определить функцию

$$\Phi(t) = \max_{x \in D} < c(t), x > .$$

Функции, с помощью которых задается изменение параметров задач, не обязательно должны быть линейными. Вид этих функций не имеет отношения к тем условиям, которые определяют принадлежность задач к классу задач линейного программирования. Преимущество использования линейных функций состоит лишь в том, что вычисления являются менее трудоемкими.

Следует отметить, что любую нелинейную функцию одной переменной всегда можно аппроксимировать кусочно-линейной функцией.

При параметрическом анализе параметр t рассматривается как время и анализ всегда начинается с момента $t=0$ (считается, что $t \geq 0$).

Общая схема анализа заключается в следующем.

1. Принимается $t=t_0=0$ и решается задача линейного программирования (1.103) или (1.104).
2. С помощью условий допустимости (1.100) и/или оптимальности (1.101) находится момент времени $t=t_1$ такой, что в интервале $[t_0, t_1]$ решение остается оптимальным и/или допустимым.

В случае **задачи А** для определения этого момента используется следующая последовательность действий:

- вычисляется вектор-функция коэффициентов разложения вектора A_0 по базису:

$$B^{-1}A_0(t) = X_0(t) = \begin{pmatrix} x_{10}(t) \\ x_{20}(t) \\ \vdots \\ x_{m0}(t) \end{pmatrix};$$

- решается система неравенств:

$$\begin{cases} x_{10}(t) \geq 0, \\ x_{20}(t) \geq 0, \\ \quad\vdots \\ x_{m0}(t) \geq 0 \end{cases}$$, откуда находится t_1 ;

определяется закон изменения оптимального значения целевой функции в интервале $[t_0, t_1]$:

$$\Phi(t) = Z_0(t) = c_{баз}X_0(t) = c_{баз}B^{-1}A_0(t).$$

Для **задачи В** последовательность действий такова:

- вычисляется вектор-функция оценок:

$\Delta(t)=(\Delta_1(t), \Delta_2(t),\ldots,\Delta_n(t))$, где $\Delta_j = c_{баз}(t)B^{-1}A_j - c_j(t)$, $j=1,2,\ldots,n$;

- решается система неравенств:

$$\begin{cases} \Delta_1(t) \ge 0, \\ \Delta_2(t) \ge 0, \\ \quad\vdots \\ \Delta_n(t) \ge 0. \end{cases}$$

Определяется закон изменения оптимального значения целевой функции в интервале $[t_0, t_1]$:

$$\Phi(t) = Z_0(t) = c_{баз} X_0 .$$

При $t > t_1$ решение становится недопустимым (**задача А**) или неоптимальным (**задача В**).

3. Принимается $t = t_1$, корректируется оптимальная симплекс-таблица, и полученная задача решается двойственным симплекс-методом (**задача А**) или обычным симплекс-методом (**задача В**). Определяется момент времени $t = t_2$ такой, что в интервале $[t_1, t_2]$ решение остается оптимальным и/или допустимым.

Этот процесс продолжается до тех пор, пока не будет определен момент времени $t = t_k$ такой, что при $t > t_k$ задача становится неразрешимой или остается неизменным предыдущее решение.

На этом параметрическое исследование задачи заканчивается.

1.4.10. Примеры анализа чувствительности коэффициентов целевой функции и правой части системы ограничений

Рассмотрим задачу **В** на следующем примере.

<u>Пример 1.28</u>

Пусть требуется провести параметрический анализ следующей задачи линейного программирования:

$$\begin{array}{l} (3-6t)x_1 + (2-2t)x_2 + (5+5t)x_3 \to max, \\ x_1 \quad + 2x_2 \quad +x_3 + x_4 \quad = 40, \\ 3x_1 \qquad 2x_3 \qquad +x_5 \quad = 60, \\ x_1 \quad +4x_2 \qquad\qquad +x_6 = 30\,, \\ x_j \ge 0,\ (j=1,2,\ldots,6). \end{array}$$

Принимаем $t = t_0 = 0$ и решаем эту задачу симплекс-методом.

Исходное опорное решение $\overline{x} = (0,0,0,40,60,30)$ имеет единичный базис.

Ниже представлена последняя симплекс-таблица, содержащая оптимальное решение задачи. Обратная базисная матрица выделена. Это ─ та же таблица, что и в предыдущем разделе, так как при $t = t_0 = 0$ соответствующие задачи совпадают.

Баз	*Сбаз*	A_o	*3*	*2*	*5*	*0*	*0*	*0*
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	*2*	*5*	*-1/4*	*1*	*0*	*1/2*	*-1/4*	*0*
A_3	*5*	*30*	*3/2*	*0*	*1*	*0*	*1/2*	*0*
A_6	*0*	*10*	*2*	*0*	*0*	*-2*	*1*	*1*
Табл.1	*(t=0)*	*160*	*4*	*0*	*0*	*1*	*2*	*0*

Определим первое критическое значение параметра $t=t_1$ такое, что в интервале $[t_0,t_1]$ решение остается оптимальным.

Проделаем это в три приема, воспользовавшись аппаратом модифицированного симплекс-метода:

1. Найдем вектор-функцию симплексных множителей:

$$\Lambda(t) = c_{баз}(t)B^{-1} = (2-2t,\, 5+5t,\, 0)\times\begin{bmatrix} 1/2 & -1/4 & 0 \\ 0 & 1/2 & 0 \\ -2 & 1 & 1 \end{bmatrix} =$$

$$= (1-t,\, 2+3t,\, 0).$$

2. Найдем оценки свободных векторов A_1, A_4, A_5, как функции параметра t:

$$\Delta_1(t) = \Lambda(t)A_1 - c_1(t) = (1-t,\, 2+3t,\, 0)\begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} - (3-6t) =$$

$$= 4 + 14t\ ;$$

$$\Delta_4(t) = \Lambda(t)A_4 - c_4 = (1-t,\, 2+3t,\, 0)\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - 0 = 1 - t\ ;$$

$$\Delta_5(t) = \Lambda(t)A_5 - c_5 = (1-t,\, 2+3t,\, 0)\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - 0 = 2 + 3t\ .$$

3. Найдем критическое значение параметра t такое, что в интервале $t\in[0,t_1]$ выполняется система неравенств:

$$\begin{cases} 4 + 14t \ge 0, \\ 1 - t \ge 0, \\ 2 + 3t \ge 0. \end{cases}$$

Решение этой системы неравенств: $t_1=1$. В интервале $t\in[0,t_1]$ решение $x^0=(0,5,30,0,0,10)$ остается оптимальным.

Оптимальное же значение целевой функции изменяется по закону:

$$Z_0(t) = c_{баз}(t)X_0 = (2-2t\ ,\ 5+5t\ ,\ 0)\times\begin{bmatrix}5\\30\\10\end{bmatrix} = 160+140t\ .$$

При $t>1$ второе неравенство перестает выполняться: оценка вектора A_4 становится отрицательной, т.е. вектор A_4 нужно ввести в базис.

Подготовим переход к новому базису. Для этого в табл. *1 (t=0)* внесем необходимые изменения.

Положим $t=1$ и соответственно изменим коэффициенты целевой функции, ее значение, используя выражение для целевой функции, а также оценки свободных векторов (оценки этих векторов, как функции параметра t известны).

В результате получим первую симплекс-таблицу *(t=1)* для продолжения решения задачи:

Баз	$C_{баз}$	A_o	*-3*	*0*	*10*	*0*	*0*	*0*
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	*0*	*5*	*-1/4*	*1*	*0*	***1/2***	*-1/4*	*0*
A_3	*10*	*30*	*3/2*	*0*	*1*	*0*	*1/2*	*0*
A_6	*0*	*10*	*2*	*0*	*0*	*-2*	*1*	*1*
Табл.1	*(t=1)*	*300*	*18*	*0*	*0*	***0***	*5*	*0*
A4	*0*	*10*	*-1/2*	*2*	*0*	***1***	*-1/2*	*0*
A_3	*10*	*30*	*3/2*	*0*	*1*	*0*	*1/2*	*0*
A_6	*0*	*30*	*1*	*4*	*0*	*0*	*0*	*1*
Табл.2	*(t=1)*	*300*	*18*	*0*	*0*	***0***	*5*	*0*

Решение задачи представлено двумя симплекс-таблицами. Следует отметить весьма важную особенность приведенного решения.

Во-первых, это оценка вектора A_4 в первой таблице.

Отрицательный знак при нуле подчеркивает то обстоятельство, что при любом, сколь угодно малом увеличении параметра t решение становится неоптимальным.

Во-вторых, при переходе ко второй таблице не изменились ни оценки векторов, ни значение целевой функции. Имеет место случай <u>*альтернативного базиса*</u>.

Этот случай имеет простую геометрическую интерпретацию: при $t=1$ линия уровня "ложится" на ребро, соединяющее две смежные вершины допустимого множества, и любой точке этого ребра соответствует одно и то же значение целевой функции (рис.1.23).

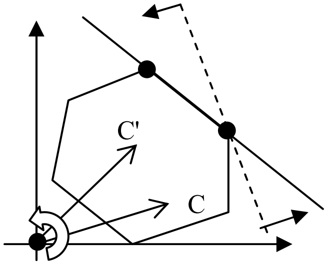


Рис.1.23. Случай альтернативного базиса

Найдем новое критическое значение параметра t такое, что в интервале $[t_1,t_2]$ решение остается оптимальным. Проделаем это, как и на предыдущей итерации, в три приема:

1. Вычислим вектор-функцию симплексных множителей:

$$\Lambda(t) = c_{баз}(t)B^{-1} = (0,\, 5+5t,\, 0) \times \begin{pmatrix} 1 & -1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, \frac{5+5t}{2}, 0);$$

2. Найдем оценки свободных векторов, как функции параметра:

$$\Delta_1(t) = \Lambda(t)A_1 - c_1(t) = (0, \frac{5+5t}{2}, 0)\begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} - (3-6t) = \frac{9+27t}{2};$$

$$\Delta_2(t) = \Lambda(t)A_2 - c_2(t) = (0, \frac{5+5t}{2}, 0)\begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix} - (2-2t) = -2+2t;$$

$$\Delta_5(t) = \Lambda(t)A_5 - c_5(t) = (0, \frac{5+5t}{2}, 0)\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - 0 = \frac{5+5t}{2}.$$

3. Найдем критическое значение параметра $t=t_2$ такое, что в интервале $[t_1,t_2]$ выполняется система неравенств:

$$\begin{cases} \dfrac{9+27t}{2} \ge 0; \\ -2+2t \ge 0; \\ \dfrac{5+5t}{2} \ge 0. \end{cases}$$

Очевидно, что при любых $t \ge 1$ эти неравенства будут выполняться, следовательно, решение, записанное во второй симплекс-таблице, $x^0=(0,0,30,10,0,30)$ будет оставаться оптимальным.

В полуоткрытом интервале $t \ge 1$ значение целевой функции будет изменяться по закону:

$$Z_0(t) = c_{баз}(t)X_0 = (0,5+5t,0)\times\begin{bmatrix}10\\30\\30\end{bmatrix} = 150+150t.$$

Параметрический анализ закончен. Его результаты сведем в таблицу.

t	Оптимальное решение						$Z_0(t)$
	x_1	x_2	x_3	x_4	x_5	x_6	
$0 \le t \le 1$	*0*	*5*	*30*	*0*	*0*	*10*	*160+140t*
$t \ge 1$	*0*	*0*	*30*	*10*	*0*	*30*	*150+150t*

Рассмотрим задачу **А** из предыдущего раздела на конкретном примере. Пусть требуется провести параметрический анализ следующей задачи ЛП.

<u>Пример 1.29</u>

$$\begin{cases} 2x_1 + 3x_2 \to \max, \\ x_2 + x_3 = 80, \\ x_1 + 2x_2 + x_4 = 180, \\ 3x_1 + 2x_2 + x_5 = 300 + t, \\ x_1 + x_6 = 80, \\ x_j \ge 0,\ j = \overline{1,6}. \end{cases}$$

Принимаем $t=0$ и решаем задачу симплекс-методом. Решение задачи для приведено в последней симплекс-таблице.

Баз	$C_{баз}$	A_0	*2*	*3*	*0*	*0*	*0*	*0*
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	*3*	*60*	*0*	*1*	*0*	*3/4*	*-1/4*	*0*
A_1	*2*	*60*	*1*	*0*	*0*	*-1/2*	*1/2*	*0*
A_3	*0*	*20*	*0*	*0*	*1*	*-3/4*	*1/4*	*0*
A_6	*0*	*20*	*0*	*0*	*0*	*1/2*	*-1/2*	*1*
Табл.4(t=0)		*300*	*0*	*0*	*0*	*5/4*	*1/4*	*0*

Обратная матрица оптимального решения выделена.

Найдем вектор-функцию разложения вектора A_0 по базису:

$$B^{-1}A_0(t)=\begin{bmatrix}0 & 3/4 & -1/4 & 0\\ 0 & -1/2 & 1/2 & 0\\ 1 & -3/4 & 1/4 & 0\\ 0 & 1/2 & -1/2 & 1\end{bmatrix}*\begin{bmatrix}80\\ 180\\ 300+t\\ 80\end{bmatrix}=\begin{bmatrix}60-t/4\\ 60+t/2\\ 20+t/4\\ 20-t/2\end{bmatrix}.$$

Решим систему неравенств:

$$\begin{cases}60-t/4 \ge 0,\\ 60+t/2 \ge 0,\\ 20+t/4 \ge 0,\\ 20-t/2 \ge 0.\end{cases}$$

Отсюда находим первое критическое значение параметра $t=t_1$ такое, что в интервале $[0, t_1]$ решение остается допустимым.

При $t>40$ последнее неравенство не выполняется, следовательно, $t_1=40$. В интервале $[0,40]$ закон изменения оптимального значения целевой функции определяется следующим образом:

$$Z_0(t)=c_{баз}X_0(t)=(3,2,0,0)\begin{bmatrix}60-t/4\\ 60+t/2\\ 20+t/4\\ 20-t/2\end{bmatrix}=3*(60-t/4)+2*(60+t/2)=$$

$$=300+t/4\ .$$

Заменим в оптимальной симплекс-таблице столбец A_0.

Для этого при $t=t_1=40$ пересчитываем столбец A_0 последней симплекс-таблицы:

$$\left.\begin{vmatrix}60-t/4\\ 60+t/2\\ 20+t/4\\ 20-t/2\end{vmatrix}\right._{t=40}=\begin{vmatrix}50\\ 80\\ 30\\ -0\end{vmatrix}$$

Вычислим новое значение целевой функции:

$$Z_{(t=40)}=300+10=310.$$

Переписываем последнюю симплекс-таблицу с новыми значениями коэффициентов столбца A_0 .

При $t>40$ четвертая координата в разложении вектора A_0 становится отрицательной, поэтому для поиска оптимального решения задачи используется двойственный симплекс-метод.

Баз	$C_{баз}$	A_0	*2*	*3*	*0*	*0*	*0*	*0*
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	*3*	*50*	*0*	*1*	*0*	*3/4*	*-1/4*	*0*
A_1	*2*	*80*	*1*	*0*	*0*	*-1/2*	*1/2*	*0*
A_3	*0*	*30*	*0*	*0*	*1*	*-3/4*	*1/4*	*0*
A_6	*0*	***-0***	*0*	*0*	*0*	*1/2*	***-1/2***	*1*
t=40		*310*	*0*	*0*	*0*	*5/4*	*1/4*	*0*
A_2	*3*	*50*	*0*	*1*	*0*	*1/2*	*0*	*-1/2*
A_1	*2*	*80*	*1*	*0*	*0*	*0*	*0*	*1*
A_3	*0*	*30*	*0*	*0*	*1*	*-1/2*	*0*	*1/2*
A_5	*0*	*0*	*0*	*0*	*0*	*-1*	*1*	*-2*
t=40		*310*	*0*	*0*	*0*	*3/2*	*0*	*1/2*

Решение задачи при *t=40*.

Найдем второе критическое значение параметра $t=t_2$ такое, что в интервале $[40,t_2]$ полученное решение остается допустимым и оптимальным:

$$B^{-1} * A_0(t) = \begin{bmatrix} 0 & 1/2 & 0 & -1/2 \\ 0 & 0 & 0 & 1 \\ 1 & -1/2 & 0 & 1/2 \\ 0 & -1 & 1 & -2 \end{bmatrix} * \begin{bmatrix} 80 \\ 180 \\ 300+t \\ 80 \end{bmatrix} = \begin{bmatrix} 50 \\ 80 \\ 30 \\ -40+t \end{bmatrix}.$$

Очевидно, что при всех $t > 40$ решение будет оставаться допустимым и оптимальным. При этом целевая функция будет иметь постоянное значение: $Z(t)=310$.

$$Z_0(t) = (3,2,0,0)\begin{bmatrix} 50 \\ 80 \\ 30 \\ -40+t \end{bmatrix} = 3*50 + 2*80 = 150 + 160 = 310.$$

Параметрический анализ закончен. Сведем его результаты в таблицу:

	Оптимальное решение						
Интервал	x_1	x_2	x_3	x_4	x_5	x_6	*Z(t)*
$0 \le t \le 40$	*60+t/2*	*60-t/4*	*20+t/4*	*0*	*0*	*20-t/2*	*300+t/4*
$t > 40$	*80*	*50*	*30*	*0*	*-40+t*	*0*	*310*

Графическая иллюстрация

$2x_1 + 3x_2 \to \max,$

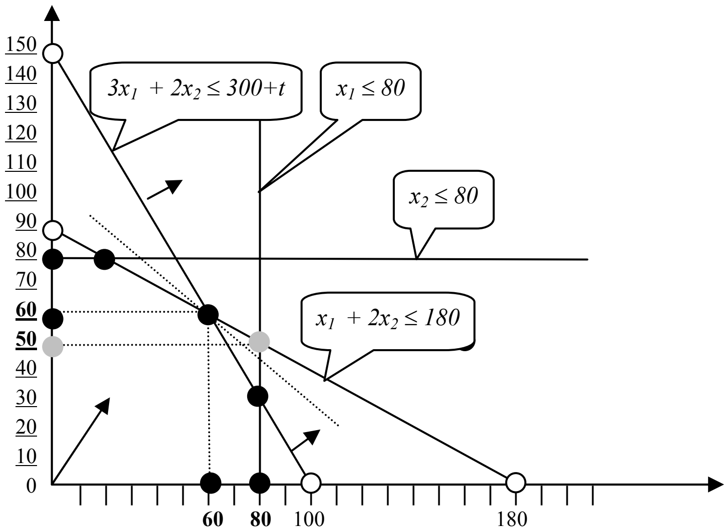
$x_2 \le 80,$ *огр.1*

$x_1 + 2x_2 \le 180,$ *огр.2*

$3x_1 + 2x_2 \le 300+t,$ *огр.3*

$x_1 \le 80,$ *огр.4*

$x_1, x_2 \ge 0.$



1.4.11. Проблемы накопления ошибок в симплекс-методе

В заключение этого раздела скажем несколько слов об основных проблемах реализации симплекс-метода.

Различные варианты МСМ отличаются, в основном, оптимизацией за счет разреженности матриц. При этом достигается не только уменьшение необходимого объема памяти, но и повышение скорости вычислений.

Значение последнего фактора не ограничивается только тем, что сокращается время решения задач ЛП.

Дело в том, что сокращение объема вычислений обеспечивает большую эффективность и в плане численной устойчивости решений.

Проблема устойчивости, связанная с накоплением ошибок, пожалуй, самая острая в симплекс-методе.

Если число итераций велико по сравнению с m, базисная матрица должна периодически обращаться заново. Необходимость пересчета обратной базисной матрицы возникает тогда, когда устанавливается факт ухудшения точности. Здесь используются специальные приемы.

Например, можно вычислить оценку любого базисного вектора – она должна быть нулевой. Можно вычислить разложение любого базисного вектора – должен получиться единичный вектор.

Контрольные вопросы и задачи к разделу 1.4

1. Составьте схему алгоритма модифицированного симплекс-метода, сопряженного с методом M-задачи для поиска исходного опорного решения.

2. Проанализируйте случай неограниченности целевой функции на допустимом множестве в методе декомпозиции Данцига-Вулфа. Каким образом возникновение этого случая должно обрабатываться в вычислительной схеме алгоритма?

3. Предложите действия при параметрическом анализе, если в момент времени t_0 при решении задачи срабатывает признак неограниченности целевой функции.

4. Предложите действия при параметрическом анализе, если в момент времени t_0 невозможно найти ни одного допустимого решения.

5. Возможен ли одновременный параметрический анализ коэффициентов целевой функции и правых частей системы ограничений?

6. Решить задачу модифицированным симплекс-методом

$$
\begin{aligned}
&9x_1 + 10x_2 \to \max \\
&3x_1 + 4x_2 \le 109 \\
&4x_1 + 4x_2 \le 124 \\
&x_1 \le 19 \\
&x_2 \le 19 \\
&x_{1,2} \ge 0.
\end{aligned}
$$

7. Дана задача ЛП:

$$-100x_1 - 260x_2 - 280x_3 - 110x_4 \to max$$
$$x_1 \quad +2x_2 \quad +x_3 \qquad\qquad -x_5 \quad = 1$$
$$x_2 \quad +2x_3 \quad +x_4 \qquad -x_6 = 4$$
$$x_j \geq 0 \ (j=1 \div 6).$$

На очередном шаге решения этой задачи МСМ получена таблица:

Баз	$C_{баз}$	A_0	E_1	E_2
A_3	-280	1	1	0
A_4	-110	2	-2	1
Табл.3				

Найти оптимальное решение этой и двойственной к ней задачи.

8. Решить задачу методом декомпозиции Данцига-Вулфа:

$$\begin{array}{lllll} x_1 & +2x_2 & +3x_3 & +x_4 & \to \max \\ 2x_1 & +3x_2 & +3x_3 & +3x_4 & = 5 \\ x_1 & +x_2 & & & \leq 3 \\ x_1 & +2x_2 & & & \leq 1 \\ & & x_3 & +2x_4 & \leq 2 \\ & & 2x_3 & +x_4 & \leq 5 \end{array}$$
$$x_j \geq 0 (j = 1..4).$$

9. Дана задача ЛП:

$$2x_1 - 4x_2 - 3x_3 + x_4 \to \max,$$
$$x_1 - 3x_2 + x_3 \qquad = 5,$$
$$2x_1 - 2x_2 \qquad + x_4 = 20$$
$$x_j \geq 0, \ (j=1 \div 4).$$

Определить зависимость оптимального решения задачи от коэффициента c_3 целевой функции.

10. Дана задача ЛП:

$$\begin{aligned} &2x_1 - 4x_2 - 3x_3 + x_4 \to \max, \\ &x_1 - 3x_2 + x_3 \qquad = 5, \\ &2x_1 - 2x_2 \qquad + x_4 = 20 \\ &x_j \geq 0, \ \ (j = 1 \div 4). \end{aligned}$$

Определить зависимость оптимального решения задачи от коэффициента a_1 целевой функции.

1.5. Дробно-линейное программирование

Рассмотрим следующую прикладную задачу. Для выполнения n различных работ могут быть использованы рабочие m квалификационных групп.

При выполнении i-й группой рабочих j-й работы выработка в единицу времени составляет c_{ij} единиц ($i=1\div m; j=1\div n$).

Общий фонд времени, в течение которого i-я группа рабочих может быть занята выполнением работ, не превышает b_i единиц времени, а j-я работа должна быть выполнена в объеме не менее a_j единиц.

Необходимо составить такой план выполнения работ, который обеспечивает максимальную производительность.

$$\text{Производительность} = \frac{\sum \textit{объем работ}}{\sum \textit{затраты времени}}.$$

Построим модель. Пусть x_{ij} – время, затрачиваемое i-й группой рабочих для выполнения j-й работы ($i=1\div m; j=1\div n$).

Тогда при плане $\{ x_{ij} \}$ общий объем работ (Z_1) составит:

$$Z_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} .$$

Общие затраты времени (Z_2) на выполнение этого объема работ определяются следующим образом:

$$Z_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} .$$

При плане $\{ x_{ij} \}$ общая производительность всех работ составит:

$$\frac{Z_1}{Z_2} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}}.$$

Сформулируем ограничения задачи.

j-я работа должна быть выполнена в объеме не менее a_j единиц. Следовательно, должно иметь место:

$$\sum_{i=1}^{m} c_{ij}x_{ij} \geq a_j \quad (j = \overline{1,n}).$$

i-я группа рабочих может быть занята выполнением работ не более b_i единиц времени:

$$\sum_{j=1}^{n} x_{ij} \leq b_i \quad (i = \overline{1,m}).$$

Последнее, естественное, ограничение – это требование неотрицательности переменных: $x_{ij} \geq 0$ ($i=1 \div m; j=1 \div n$).

Окончательно модель приобретает вид:

$$\frac{Z_1}{Z_2} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}} \rightarrow \max$$

$$\sum_{i=1}^{m} c_{ij}x_{ij} \geq a_j, \quad (j = \overline{1,n})$$

$$\sum_{j=1}^{n} x_{ij} \leq b_i, \quad (i = \overline{1,m})$$

$$x_{ij} \geq 0, \ (i=1 \div m; j=1 \div n).$$

В практике планирования с использованием математических моделей оптимизационных задач подобные нелинейные задачи встречаются довольно часто. Они составляют целый класс задач математического программирования – класс задач *дробно-линейного (ДЛП)* или *гиперболического программирования*. В общей постановке задача ДЛП имеет вид:

$$Z=\frac{\sum_{j=1}^{n} c_j x_j}{\sum_{j=1}^{n} d_j x_j} \to \max \qquad (1.105)$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_i \quad (i=\overline{1,m}) \qquad (1.106)$$

$$x_j \geq 0 \ (j=1 \div n\ ). \qquad (1.107)$$

При этом предполагается, что $\sum_{j=1}^{n} d_j x_j \geq 0$ и, кроме того, $\sum_{j=1}^{n} d_j x_j \neq 0$ в области неотрицательных решений системы уравнений (1.106). Заметим, что условие $\sum_{j=1}^{n} d_j x_j > 0$ в этой области не нарушает общности задачи, так как в противном случае знак минус всегда можно отнести к числителю.

Для этой задачи характерны следующие свойства. Как и в случае задачи ЛП, своего максимального значения ЦФ (1.105) достигает в одной из вершин выпуклого многогранника, определяемого ограничениями (1.106) и (1.107) (естественно, при условии, что задача имеет решение). Если же ЦФ принимает максимальное значение более, чем в одной вершине, то она достигает это значение в любой точке, являющейся выпуклой линейной комбинацией данных вершин.

Эти свойства хорошо иллюстрируются путем геометрической интерпретации задачи ДЛП.

1.5.1. Геометрическая интерпретация задачи дробно-линейного программирования

Рассмотрим случай двух переменных:

$$Z=\frac{c_1 x_1 + c_2 x_2}{d_1 x_1 + d_2 x_2} \to \max$$

$$a_{i1} x_1 + a_{i2} x_2 \leq a_i \ \ (i=1,2,\dots,m)$$

$$x_1,\ x_2 \geq 0$$

Будем считать, что в области допустимых решений (D) имеет место $d_1x_1+ d_2x_2 \neq 0$.

Для того чтобы найти решение задачи, сначала построим многогранник решений, определенный ограничениями задачи (рис.1.24).

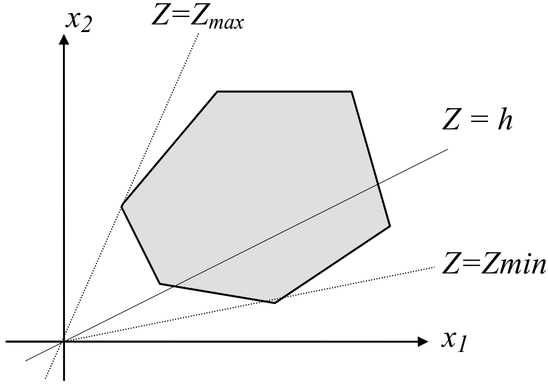


Рис.1.24. Геометрическая интерпретация задачи ДЛП

Положим значение ЦФ равным некоторому числу h. То есть ЦФ будет принимать одно и то же значение во всех точках прямой

$\dfrac{c_1x_1 + c_2x_2}{d_1x_1 + d_2x_2} = h$ или

$$(c_1-d_1h)x_1+(c_2-d_2h)x_2=0. \qquad (1.108)$$

Очевидно, что эта прямая проходит через начало координат. Для того чтобы найти допустимые решения, на которых ЦФ принимает значение h, прямая должна иметь общие точки с многоугольником.

Начнем увеличивать параметр h. Увеличение этого параметра приведет к вращению прямой (1.108) вокруг начала координат либо по, либо против часовой стрелки, в зависимости от сочетания параметров c_j, d_j ($j=1,2$).

Из геометрических соображений ясно, что если допустимое множество ограничено, при некотором значении $h=h^*$ прямая (1.108) станет опорной к допустимому множеству. При этом в точке (точках) касания будет достигнуто искомое оптимальное решение.

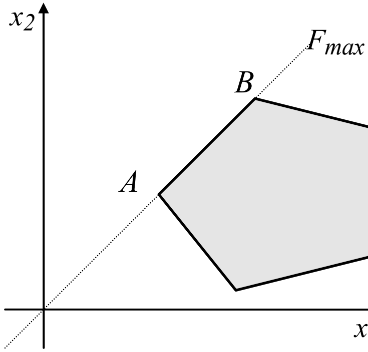


Рис.1.25. Случай бесконечного количества оптимальных решений

На рис. 1.25 представлен случай, когда максимум ЦФ достигается в любой точке отрезка $[A,B]$.

Представляет интерес случай, когда допустимое множество не ограничено. Здесь возможны следующие ситуации.

А) Допустимое множество не ограничено, однако существуют вершины, в которых ЦФ принимает соответственно максимальное и минимальное значение (рис.1.26).

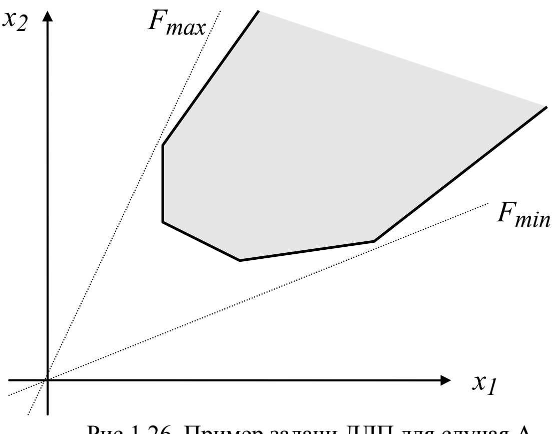


Рис.1.26. Пример задачи ДЛП для случая А

В) Допустимое множество не ограничено, и один из экстремумов не достигается. Например, минимум достигается в одной из вершин, а максимум не достигается вообще (рис.1.27).

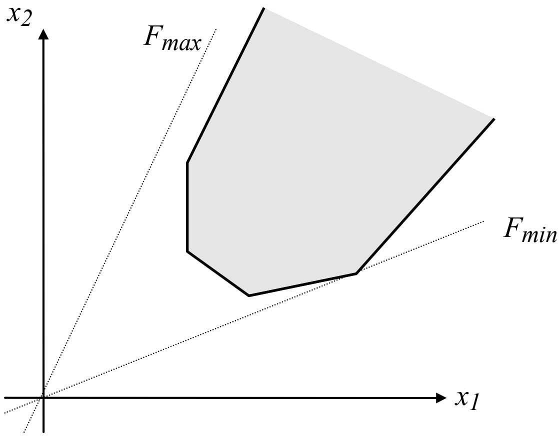


Рис.1.27. Случай асимптотического максимума

Это – случай *асимптотического максимума*.[14]

В принципе, возможна ситуация, когда имеет место и асимптотический максимум и асимптотический минимум (рис.1.28).

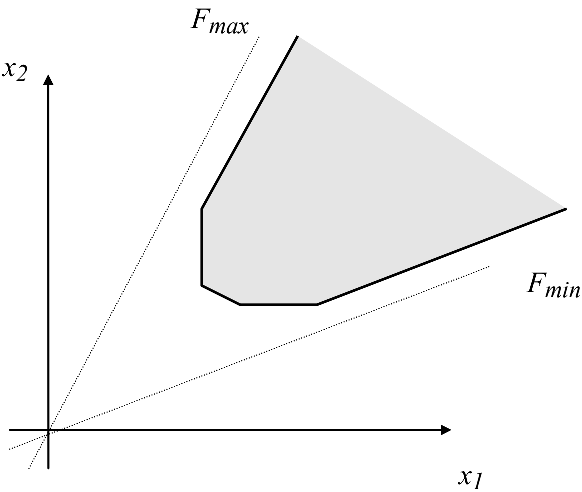


Рис.1.28. Случай асимптотического максимума и минимума

[14] Будет большой ошибкой, если этот случай считать случаем неограниченности ЦФ сверху. Здесь ЦФ ограничена сверху, но граница эта недостижима!

1.5.2. Сведение задачи ДЛП к задаче ЛП

Пусть дана задача ДЛП:

$$Z=\frac{\sum_{j=1}^{n} c_j x_j}{\sum_{j=1}^{n} d_j x_j} \to \max \qquad (1.109)$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_i, \quad (i=\overline{1,m}) \qquad (1.110)$$

$$x_j \geq 0, \ (j=1 \div n). \qquad (1.111)$$

Кроме того, предполагается, что в области неотрицательных решений системы уравнений (1.110) имеет место $\sum_{j=1}^{n} d_j x_j > 0$.

Это предположение требует, чтобы имело место

$\{x \left| \sum_{j=1}^{n} d_j x_j = 0 ; \sum_{j=1}^{n} a_{ij} x_j = a_i, i=\overline{1,m} ; x_j \geq 0, j=\overline{1,n} \right.\} = \varnothing$,

где $x=(x_1, x_2, ..., x_n)$.

Примем обозначение:

$$y_0 = \frac{1}{\sum_{j=1}^{n} d_j x_j}. \qquad (1.112)$$

Кроме того, введем новые переменные:

$$y_j = y_0 x_j \ (j=1 \div n) \qquad (1.113)$$

или $x_j = \frac{y_j}{y_0}$.

Из (1.112) имеем $y_0 \sum_{j=1}^{n} d_j x_j = 1$.

Подставим в это выражение $x_j = \frac{y_j}{y_0}$. Получим $\sum_{j=1}^{n} d_j y_j = 1$.

Теперь исходная задача приобретает следующий вид:

$$F^* = \sum_{j=1}^{n} c_j y_j \to \max$$

$$\sum_{j=1}^{n} a_{ij} y_j - a_i y_0 = 0, \quad (i = \overline{1,m})$$

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

$$y_j \geq 0 \ \ (j = \overline{1,n}); \quad y_0 \geq 0.$$

Это – задача линейного программирования. Решив эту задачу любым известным методом, всегда можно восстановить оптимальное решение исходной задачи (1.109)-(1.111).

Пример 1.30

Решить задачу ДЛП:

$$\frac{2x_1 + 3x_2}{x_1 + x_2} \to \max,$$

$$x_1 - x_3 = 4,$$
$$x_2 + x_4 = 8,$$
$$x_j \geq 0, j = 1 \div 4.$$

- Вводим переменную $y_0 = \dfrac{1}{x_1 + x_2}$.
- Производим замену переменных: $y_j = y_0 x_j$.
- Приходим к следующей эквивалентной задаче ЛП:

$$2y_1 + 3y_2 \to max$$
$$y_1 - y_3 - 4y_0 = 0$$
$$y_2 + y_4 - 8y_0 = 0$$
$$y_1 + y_2 = 1$$
$$y_j \geq 0, j = 0 \div 4.$$

Переходим к M-задаче (одновременно вектор A_3 делаем единичным базисным вектором, умножая на "-1" первое уравнение-ограничение):

$$2y_1 + 3y_2 - My_5 \to max$$
$$-y_1 + y_3 + 4y_0 = 0$$
$$y_2 + y_4 - 8y_0 = 0$$
$$y_1 + y_2 + y_5 = 1$$
$$y_j \geq 0, j = 0 \div 5.$$

Решим эту задачу, для разнообразия придав M конкретное большое значение 100.

			2	3	0	0	-100	0
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
A_3	0	0	-1	0	1	0	0	4
A_4	0	0	0	**1**	0	1	0	-8
A_5	-100	1	1	1	0	0	1	0
Табл. 1		-100	-102	-103	0	0	0	0
A_3	0	0	-1	0	1	0	0	**4**
A_2	3	0	0	1	0	1	0	-8
A_5	-100	1	1	0	0	-1	1	8
Табл. 2		-100	-102	0	0	103	0	-824
A_6	0	0	-1/4	0	1/4	0	0	1
A_2	3	0	-2	1	2	1	0	0
A_5	-100	1	**3**	0	-2	-1	1	0
Табл.3		-100	-308	0	206	103	0	0
A_6	0	1/12	0	0	1/12	-1/12	1/12	1
A_2	3	2/3	0	1	2/3	1/3	2/3	0
A_1	2	1/3	1	0	-2/3	-1/3	1/3	0
Табл.4		8/3	0	0	2/3	1/3	308/3	0

Решение эквивалентной задачи:

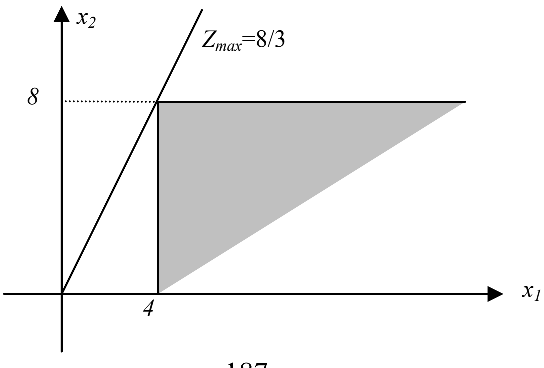
$$y_1=1/3,\ y_2=2/3,\ y_0=y_6=1/12;\ Z_{опт}=8/3.$$

Решение исходной задачи:

$$x_1=y_1/y_0=1/3:1/12=4;$$
$$x_2=y_2/y_0=2/3:1/12=8;$$
$$Z_{опт}=8/3.$$

Геометрическая интерпретация

$$x_1 - x_3 = 4 \quad : \quad x_3 = x_1 - 4 \geq 0$$
$$x_2 + x_4 = 8 \quad : \quad x_4 = 8 - x_2 \geq 0$$
$$\Downarrow$$
$$\boldsymbol{x_1 \geq 4;}$$
$$\boldsymbol{x_2 \leq 8.}$$



1.5.3. Задача ДЛП со свободными членами в числителе и знаменателе

Задача со свободными членами имеет вид

$$Z = \frac{c_0 + \sum_{j=1}^{n} c_j x_j}{d_0 + \sum_{j=1}^{n} d_j x_j} \to \max \qquad (1.114)$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_i \quad (i = \overline{1, m}) \qquad (1.115)$$

$$x_j \geq 0 \ (j = 1 \div n). \qquad (1.116)$$

Кроме того, предполагается, что в области неотрицательных решений системы уравнений (1.115) имеет место $d_0 + \sum_{j=1}^{n} d_j x_j > 0$.

Примем обозначение:

$$y_0 = \frac{1}{d_0 + \sum_{j=1}^{n} d_j x_j}. \qquad (1.117)$$

Введем новые переменные:

$$y_j = y_0 x_j \ (j = 1 \div n) \qquad (1.118)$$

или $x_j = \frac{y_j}{y_0}$.

Из (1.117) имеем: $d_0 y_0 + y_0 \sum_{j=1}^{n} d_j x_j = 1$. Подставим в это выражение $x_j = \frac{y_j}{y_0}$. Получим $d_0 y_0 + \sum_{j=1}^{n} d_j y_j = 1$. Теперь исходная задача приобретает следующий вид:

$$F^* = c_0 y_0 + \sum_{j=1}^{n} c_j y_j \to \max$$

$$-a_i y_0 + \sum_{j=1}^{n} a_{ij} y_j = 0, \quad (i = \overline{1,m})$$

$$d_0 y_0 + \sum_{j=1}^{n} d_j y_j = 1$$

$$y_j \geq 0 \ \ (j = \overline{1,n}); \quad y_0 \geq 0.$$

Рассмотрим геометрическую интерпретацию этой задачи, для чего возьмем случай двух переменных:

$$Z = \frac{c_0 + c_1 x_1 + c_2 x_2}{d_0 + d_1 x_1 + d_2 x_2} \to \max$$

$$a_{i1} x_1 + a_{i2} x_2 \leq a_i \ \ (i=1,2,...,m)$$

$$x_1, x_2 \geq 0.$$

Будем считать, что в области допустимых решений (D) имеет место $d_0 + d_1 x_1 + d_2 x_2 \neq 0$.

Положим значение ЦФ равным некоторому числу h. То есть ЦФ будет принимать одно и то же значение во всех точках прямой

$\dfrac{c_0 + c_1 x_1 + c_2 x_2}{d_0 + d_1 x_1 + d_2 x_2} = h$ или

$$(c_0 - hd_0) + (c_1 - d_1 h) x_1 + (c_2 - d_2 h) x_2 = 0. \qquad (1.119)$$

Для того чтобы найти допустимые решения, на которых ЦФ принимает значение h, прямая должна иметь общие точки с многоугольником допустимых решений.

Начнем увеличивать параметр h. Увеличение этого параметра приведет к вращению прямой (1.119) вокруг некоторой точки $\overline{x} = (\overline{x}_1, \overline{x}_2)$ либо по, либо против часовой стрелки, в зависимости от сочетания параметров c_j, d_j ($j=0,1,2$).

Координаты точки $\overline{x} = (\overline{x}_1, \overline{x}_2)$ определяются следующим образом[15] (рис.1.29):

$$\overline{x}_1 = \frac{-c_0 d_2 + c_2 d_0}{c_1 d_2 - c_2 d_1},$$

$$\overline{x}_2 = \frac{-c_1 d_0 + c_0 d_1}{c_1 d_2 - c_2 d_1}.$$

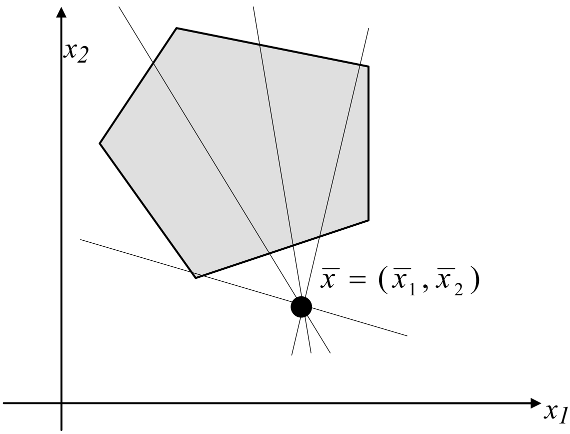


Рис.1.29. Геометрическая интерпретация задачи ДЛП со свободными членами

Из геометрических соображений ясно, что, если допустимое множество ограничено, при некотором значении $h=h^*$ прямая (1.119) станет опорной к допустимому множеству. При этом в точке (точках) касания будет достигнуто искомое оптимальное решение.

Контрольные вопросы и задачи к разделу 1.5

1. Как при решении задачи дробно-линейного программирования интерпретировать случай, когда перменная $y_0=0$ в оптимальном решении эквивалентной задачи линейного программирования?

---

[15] Действительно, если h=0, то $c_0 + c_1 x_1 + c_2 x_2 = 0$. Если h=∞, то $d_0 + d_1 x_1 + d_2 x_2 = 0$. Решаем систему из двух уравнений с двумя неизвестными. Получаем решение.

2. Рассчитайте координаты точки, вокруг которой вращается линия уровня в задаче дробно-линейного программирования со свободными членами.

3. Решите задачу дробно-линейного программирования

$$\frac{4x_1 + 6x_2}{-x_1 + 2x_2} \to \max$$

$$3x_1+2x_2 \le 37$$
$$4x_1+3x_2 \le 51$$
$$x_1 \ge 5$$
$$x_2 \ge 5$$
$$x_{1,2} \ge 0.$$

4. Решите задачу дробно-линейного программирования со свободными членами

$$\frac{3x_1 + 2x_2 + 1}{x_1 - x_2 + 2} \to \max$$

$$4x_1+3x_2 \le 8$$
$$-2x_1+5x_2 \le 9$$
$$x_{1,2} \ge 0.$$

1.6. Квадратичное программирование

1.6.1. Метод множителей Лагранжа

Метод Лагранжа является наиболее распространенным методом решения нелинейных оптимизационных задач с ограничениями в виде уравнений:

$$\begin{aligned} &f(x_1,...,x_n) \to \max \\ &g_i(x_1,...,x_n) = 0 \ \ (i = 1..m) \end{aligned} . \qquad (1.120)$$

При этом делается допущение о дифференцируемости всех функций $f(x_1,...,x_n)$, $g_i(x_1,...,x_n)$.

Метод основан на построении функции Лагранжа вида

$$L(x_1,..,x_n,\lambda_1,...,\lambda_m) = f(x_1,..,x_n) + \sum_{i=1}^{m} \lambda_i g_i(x_1,..,x_n). \qquad (1.121)$$

Пусть $x^* = (x_1^*,...,x_n^*)$ – оптимальное решение задачи (1.120). Тогда показано, что существует такой вектор $\lambda^* = (\lambda_1^*,...,\lambda_m^*)$, что

$$\nabla L(x_1^1,..,x_1^*,\lambda_1^*,...,\lambda_m^*)=0\,. \qquad (1.122)$$

Рассмотрим двумерный случай с одним нелинейным уравнением. Пусть задача имеет вид

$$\begin{aligned} &f(x_1,x_2)\to \max \\ &g(x_1,x_2)=0 \end{aligned}\,. \qquad (1.123)$$

Пусть уравнение $g(x_1,x_2)=0$ задает в пространстве гладкую кривую (рис.1.30).

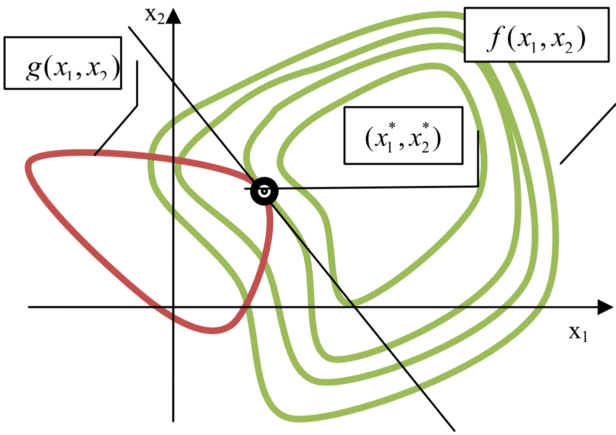


Рис.1.30. Геометрическая интерпретация метода множителей Лагранжа

Из геометрических соображений очевидно, что оптимальное решение задачи (x_1^*,x_2^*) может быть расположено только в точке касания кривой $g(x_1,x_2)=0$ и какой-либо линии уровня целевой функции $f(x_1,x_2)=h$, поскольку в бесконечно малой окрестности любой точки пересечения (но не касания) этих двух кривых можно найти точки как превышающие значение h, так и меньшие этого значения. Поскольку для каждой точки в пространстве градиент произвольной функции $\psi(x_1,x_2)$ перпендикулярен касательной к линии уровня этой функции, требование касания кривых $g(x_1,x_2)=0$ и $f(x_1,x_2)=h$ эквивалентно требованию параллельности векторов градиентов соответствующих функций:

$$\nabla f(x_1,x_2)=\lambda\nabla g(x_1,x_2)\,. \qquad (1.124)$$

Таким образом, равенство (1.124) должно выполняться в оптимальном решении (x_1^*, x_2^*) задачи (1.123). Однако мы решаем обратную задачу – поиска (x_1^*, x_2^*). Очевидно, что искомая точка должна удовлетворять ограничениям исходной задачи (1.123):

$g(x_1^*, x_2^*) = 0$.

Построим функцию Лагранжа для задачи (1.123):

$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2)$.

Вычислим элементы её градиента $\nabla L(x_1, x_2, \lambda) = (\frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \frac{\partial L}{\partial \lambda})$ и приравняем их нулю:

$$\frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1} = 0 \tag{1.125}$$

$$\frac{\partial L}{\partial x_2} = \frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2} = 0 \tag{1.126}$$

$$\frac{\partial L}{\partial \lambda} = g(x_1, x_2) = 0 . \tag{1.127}$$

Очевидно, что уравнения (1.125) и (1.126) эквивалентны необходимому условию касания кривых $g(x_1, x_2) = 0$ и $f(x_1, x_2) = h$ (1.123), а уравнение (1.127) требует, чтобы решение уравнений (1.125) и (1.126) находилось в допустимой области. Таким образом мы показали справедливость утверждения (1.122) в двумерном случае с одним ограничением. В действительности это утверждение справедливо для задач любой размерности.

В общем случае, условие (1.122) является только необходимым условием для оптимального решения задачи (1.120). Для получения критериев достаточности необходимо анализировать вторые производные функции Лагранжа, что значительно усложняет процедуру поиска глобального экстремума. На практике метод множителей Лагранжа, как правило, применяется для задач, в которых условие (1.122) является одновременно необходимым и достаточным для существования точки глобального экстремума. Важным множеством таких задач являются задачи квадратичного программирования, рассмотренные в разделе 1.6.4. В этом случае алгоритм решения задачи (1.120) с помощью метода множителей Лагранжа выглядит следующим образом.

Шаг 1. Построить функцию Лагранжа (1.121):

$$L(x_1,..,x_n,\lambda_1,...,\lambda_m) = f(x_1,..,x_n) + \sum_{i=1}^{m} \lambda_i g_i(x_1,..,x_n).$$

Шаг 2. Вычислить градиент функции (первые частные производные функции по переменным x_j и λ_i):

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j}$$

$$\frac{\partial L}{\partial \lambda_i} = g_i(x_1,..,x_n).$$

Шаг 3. Приравнять нулю все элементы градиента функции Лагранжа, решить систему из $m+n$ уравнений с $m+n$ неизвестными из :

$$\frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0 \ \ (j = 1..n)$$

$$g_i(x_1,...,x_n) = 0 \ \ (i = 1..m). \qquad (1.128)$$

Шаг 4. Исследовать стационарные точки – решения системы уравнений (1.128). Поскольку мы предположили, что имеем дело с задачей, для которой условие (1.122) является достаточным условием оптимальности решения, полученные решения системы (1.128) являются оптимальными решениями исходной задачи (1.120).

1.6.2. Элементы теории выпуклого программирования

Для обоснования аппарата выпуклого программирования нам понадобятся некоторые элементы линейной алгебры, которые для справки приведены в этом разделе.

Функция $f(X)$ называется *строго выпуклой*, если для любых двух различных точек X_1 и X_2 выполняется неравенство:

$$f(\lambda X_1 + (1-\lambda)X_2) < \lambda f(X_1) + (1-\lambda) f(X_2), \text{ где } 0 < \lambda < 1.$$

Функция $f(X)$ называется <u>*строго вогнутой*</u>, если функция $-f(X)$ – строго выпукла.

Специальный случай, когда $f(X) = CX + X^T DX$, где $C = (c_1, c_2, \ldots, c_n)$, а D – симметрическая матрица. Доказано, что $f(X)$ будет строго выпуклой, если матрица D положительно определенная, и строго вогнутой, если матрица D отрицательно определенная.

Пусть $X = (x_1, x_2, \ldots, x_n)^T$ и $D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$.

Тогда функция $Q(X) = X^T DX = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_i x_j$ называется <u>*квадратичной формой*</u>.

Будем считать, что D – симметрическая матрица. Действительно, значение квадратичной формы не изменится, если каждый коэффициент из пары d_{ij} и d_{ji} $(i \neq j)$ заменить на $(d_{ij} + d_{ji})/2$.

Различают следующие типы квадратичных форм.

Квадратичная форма называется <u>*положительно определенной*</u>, если $Q(X) > 0$ для всех $X \neq 0$.

Квадратичная форма называется <u>*положительно полуопределенной*</u>, если $Q(X) \geq 0$ для всех X и существует вектор $X \neq 0$ такой, что $Q(X) = 0$.

Квадратичная форма называется <u>*отрицательно определенной*</u>, если квадратичная форма $-Q(X) > 0$ является положительно определенной.

Квадратичная форма называется <u>*отрицательно полуопределенной*</u>, если квадратичная форма $-Q(X) > 0$ является положительно полуопределенной.

Во всех остальных случаях квадратичная форма называется <u>*неопределенной*</u>.

Для того чтобы определить тип квадратичной формы, используются необходимые и достаточные условия, которые определяются следующими утверждениями.

Квадратичная форма *Q(X)* будет положительно определенной (положительно полуопределенной), если значения всех угловых миноров определителя $|D|$ положительны (неотрицательны). В этом случае матрица D называется положительно определенной (полуопределенной).

Квадратичная форма *Q(X)* будет отрицательно определенной, если значения k-х угловых миноров определителя $|D|$ отличны от нуля и имеют знак $(-1)^k$, $k=1, 2, \ldots, n$. В этом случае матрица D называется отрицательно определенной.

Квадратичная форма *Q(X)* будет отрицательно полуопределенной, если значения k-х угловых миноров определителя $|D|$ равны нулю либо имеют знак $(-1)^k$, $k=1, 2, \ldots, n$. В этом случае матрица D называется отрицательно полуопределенной.

k-м угловым минором определителя $|D|$ называется определитель вида:

$$\Delta_k = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1k} \\ d_{21} & d_{22} & \cdots & d_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ d_{k1} & d_{k2} & \cdots & d_{kk} \end{bmatrix}, \quad k=1, 2, \ldots, n.$$

Пример 1.31

$$Q(X) = X^T DX = -2x_1^2 - 2x_2^2 + 2x_1 x_2 = (x_1, x_2)\begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Здесь D преобразована к симметрическому виду.

Найдем все угловые миноры:

$$\Delta_1 = -2 < 0 \ (\text{знак } (-1)^1), \quad \Delta_2 = \begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix} = 3 > 0 \ (\text{знак } (-1)^2).$$

Значения k-х угловых миноров определителя $|D|$ отличны от нуля и имеют знак $(-1)^k$, $k=1, 2, \ldots, n$.

То есть матрица D отрицательно определенная и, соответственно, функция *Q(X)* является строго вогнутой.

Пример 1.32

$$Q(X) = X^TDX = -2x_1^2 - 2x_2^2 + 4x_1x_2 = (x_1, x_2)\begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Найдем все угловые миноры:

$$\Delta_1 = -2 < 0 \ \text{(знак } (-1)^1\text{)}, \quad \Delta_2 = \begin{vmatrix} -2 & 2 \\ 2 & -2 \end{vmatrix} = 0.$$

То есть матрица D отрицательно полуопределенная и соответственно, функция $Q(X)$ является нестрого вогнутой.

1.6.3. Выпуклый анализ, условия Куна-Такера

Дана задача нелинейного программирования

$$\begin{cases} z = f(x_1, x_2, ..., x_n) \to \max(\min) \\ g_i(x_1, x_2, ..., x_n) \le 0, (i = 1, 2, ..m) \end{cases}. \tag{1.129}$$

Задача (1.129) относится к классу задач *выпуклого программирования*, если z является вогнутой (выпуклой) функцией, а область допустимых решений, которая определяется ограничениями $g_i(x_1, x_2, \ldots, x_n) \le 0$ $(i = 1, 2, \ldots, m)$, представляет собой выпуклое множество.

Важнейшее свойство задач выпуклого программирования заключается в том, что любая экстремальная точка является точкой глобального экстремума.

Рассмотрим геометрическую интерпретацию задачи выпуклого программирования. Пусть в задаче (1) n=2 и m=3.

Обозначим $X' = (x'_1, x'_2)$ – точку максимума целевой функции в задаче без ограничений, $X'' = (x''_1, x''_2)$ – точку условного максимума целевой функции (оптимальное решение задачи (1.129)). На рис. 1.31 представлены возможные случаи решения задачи. Пунктирные линии – линии уровня, на которых ЦФ принимает одно и то же значение.

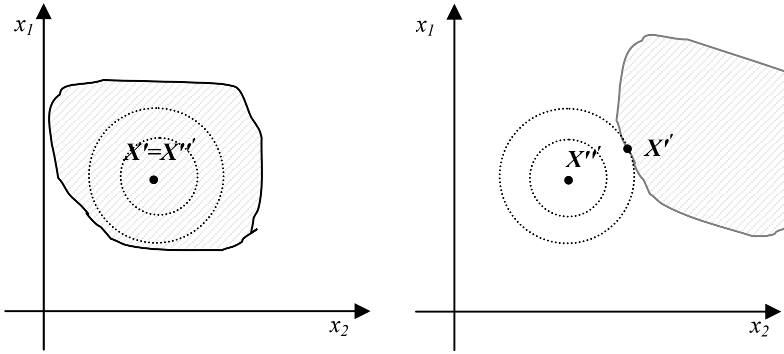


Рис.1.31. Случаи решения задачи выпуклого программирования

Перепишем теперь задачу в матрично-векторной форме:

$$z = f(X^T) \rightarrow max$$
$$G(X) \le 0,$$

где $X = (x_1, x_2, \ldots, x_n)^T$, $G(X) = (g_1(X), g_2(X), \ldots, g_m(X))^T$.

От ограничений-неравенств перейдем к ограничениям-равенствам, введя дополнительные переменные $s_i^2 \ge 0$ $(i = 1, 2, \ldots, m)$:

$z = f(x_1, x_2, \ldots, x_n) \rightarrow \max,$

$g_i(x_1, x_2, \ldots, x_n) + s_i^2 = 0, \quad (i = 1, 2, \ldots, m).$

В векторной форме задача имеет вид:

$$z = f(X^T) \rightarrow max$$
$$G(X) + S^2 = 0,$$

где $S^2 = (s_1^2, s_2^2, \ldots, s_m^2)^{\mathrm{T}}$.

Построим функцию Лагранжа:

$$L(X, S, \Lambda) = f(X^T) - \Lambda^T[G(X) + S^2].$$

Здесь $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^{\mathrm{T}}$ – вектор-столбец множителей Лагранжа.

В задаче максимизации (минимизации) необходимым условием оптимальности является неотрицательность (неположительность) множителей Λ. Остальные условия формулируются следующим образом.

$$\frac{\partial L}{\partial X} = \nabla f(X) - L\nabla G(X) = 0 \qquad (1.130)$$

$$\frac{\partial L}{\partial S} = -2\Lambda^T S = 0 \tag{1.131}$$

$$\frac{\partial L}{\partial \Lambda} = -(G(X) + S^2) = 0\,. \tag{1.132}$$

Рассмотрим условия (1.131):

$$\Lambda^T S = (\lambda_1,\ \lambda_2,\ \dots,\ \lambda_m)\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix} = (\ \lambda_1 s_1,\ \lambda_2 s_2,\ \dots,\ \lambda_m s_m) = (0,\ 0,\ \dots,\ 0)$$

или, что эквивалентно:

$$\lambda_i s_i = 0 \ \text{ для всех } \ i = 1,\ 2,\ \dots,\ m. \tag{1.133}$$

Условия (1.133) можно интерпретировать следующим образом.

Если $\lambda_i \neq 0$, то $s_i = 0$. То есть, ограничение $g_i(X) \leq 0$ должно обращаться в строгое равенство ($g_i(X) = 0$).

Если $s_i \neq 0$, то $\lambda_i = 0$. При этом ограничение $g_i(X) \leq 0$ должно обращаться в строгое неравенство ($g_i(X) < 0$).

Таким образом, условия фактически эквивалентны следующим условиям:

$$\lambda_i\, g_i(X) = 0 \text{ для всех } \ i = 1,\ 2,\ \dots,\ m.$$

В векторной форме эти условия записываются следующим образом:

$$\Lambda^T G = 0. \tag{1.134}$$

Учитывая условия (1.134) и приведенную выше интерпретацию условий (1.133), условия (1.132) можно заменить условиями $G(X) \leq 0$.

Теперь для задачи максимизации можно сформулировать условия Куна-Таккера, необходимые для того, чтобы векторы X и Λ определяли стационарную точку:

$$\begin{cases} \Lambda \geq 0 \\ \nabla f(X) - \Lambda \nabla G(X) = 0 \\ \Lambda^T G = 0 \\ G(X) \leq 0 \end{cases} \tag{1.135}$$

Для задачи выпуклого программирования необходимые условия Куна-Таккера являются также и достаточными условиями. Этот факт используется при решении задач квадратичного программирования.

1.6.4. Сведение задачи квадратичного программирования к задаче линейного программирования

Задачи квадратичного программирования часто встречаются в экспериментальной физике, в том числе в той её области, которая связана с ядерными исследованиями. Это обусловлено формулой оценки ошибки в экспериментальных данных как суммы квадратов отклонений измеряемой величины от среднего. Как правило, условия проведения эксперимента и последующие вычисления строятся таким образом, чтобы минимизировать ошибку, что позволяет явным образом сформулировать оптимизационную задачу, в которой целевая функция имеет вид квадратичной формы. Классическим примером широкого применения задачи квадратичного программирования (без ограничений) в эксперименте является известный метод наименьших квадратов [3]. Далее в пособии будет рассматриваться методика решения подобных задач в условиях существования линейных ограничений на допустимую область задачи.

Задача квадратичного программирования имеет следующий вид:

$$\begin{cases} z = CX + X^T DX \to \max \\ AX \leq b \\ X \geq 0 \end{cases}, \tag{1.136}$$

где

$$X = (x_1, x_2, \ldots, x_n)^{\mathrm{T}},\ C = (c_1, c_2, \ldots, c_n),\ b = (b_1, b_2, \ldots, b_m)^{\mathrm{T}},$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}.$$

Функция $X^T DX$ является квадратичной формой. В дальнейшем будем считать, что D – симметрическая матрица. Действительно, значение квадратичной формы не изменится, если каждый коэффициент из пары d_{ij} и d_{ji} $(i \neq j)$ заменить на $(d_{ij} + d_{ji})/2$.

Кроме того, примем (пока) следующее допущение. В задаче максимизации матрица D – отрицательно определенная. В задаче минимизации матрица D – положительно определенная. Это означает, что функция z является строго выпуклой по переменным X в случае задачи минимизации и строго вогнутой – в задаче максимизации. Линейность ограничений гарантирует выпуклость допустимого множества.

Для приведенных предположений необходимые условия Куна-Таккера являются также достаточными для установления глобального оптимума.

Рассмотрим случай максимизации. При этом общая постановка задачи имеет следующий вид:

$$\begin{cases} z = CX + X^T DX \to \max \\ G(X) = \begin{bmatrix} A \\ -E \end{bmatrix} X - \begin{bmatrix} b \\ 0 \end{bmatrix} \le 0 \end{cases}. \qquad (1.137)$$

Здесь в ограничения $G(X)$ явным образом включено требование неотрицательности переменных ($X \geq 0$).

Обозначим $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^{\mathrm{T}}$ и $M = (\mu_1, \mu_2, \ldots, \mu_n)^{\mathrm{T}}$ – множители Лагранжа, соответствующие ограничениям $AX \leq b$ и $X \geq 0$.

Условия Куна-Таккера (1.135) имеют вид:

Условия Куна-Таккера для задачи кв. программирования	Условия для общей задачи	
$$\begin{cases} z = CX + X^T DX \to \max \\ G(X) = \begin{bmatrix} A \\ -E \end{bmatrix} X - \begin{bmatrix} b \\ 0 \end{bmatrix} \le 0 \end{cases}$$	$z = f(X^T) \to max,$ $G(X) \le 0$	
$\Lambda \ge 0,\ M \ge 0$ $C+2X^TD - \Lambda^TA + M = 0.$	$\Lambda \ge 0$	(1.138)
Здесь $\nabla z = C+2X^TD$ и $\nabla G=(A - E)^T$	$\nabla f(X) - \Lambda \nabla G(X) = 0$	(1.139)
$\lambda_i \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right) = 0,\ (i=1 \div m),$	$\Lambda^T G = 0$	(1.140)
$\mu_j x_j = 0,\ (i=1, 2, \dots, n),$		(1.141)
$AX - b \le 0,\ -X \le -0$	$G(X) \le 0$	(1.142)

Обозначим $S = b - AX$ – вектор дополнительных переменных $S = (s_1, s_2, \dots, s_m)^T$. Тогда приведенные выше условия приобретают вид:

$$-2X^TD+\Lambda^TA - M = C \text{ из } (1.139), \qquad (1.143)$$

$$AX + S = b \text{ из } (1.142), \qquad (1.144)$$

$$\mu_j x_j = 0 = \lambda_i s_i \text{ для всех } i \text{ и } j \text{ из } (1.140) \text{ и } (1.141), \qquad (1.145)$$

$$X, \Lambda, M, S \ge 0 \text{ из } (1.135) \text{ и } (1.139). \qquad (1.146)$$

Ввиду того, что $D^T = D$, в результате транспонирования системы уравнений (1.143) будет иметь место $-2DX+A^T \Lambda - M = C$.

Теперь можно записать систему уравнений, решение которой позволяет найти векторы X и Λ, определяющие стационарную точку функции Лагранжа, и следовательно, координаты оптимального решения исходной задачи (X).

$$\begin{bmatrix} -2D & A^T & -E & 0 \\ A & 0 & 0 & E \end{bmatrix} \begin{bmatrix} X \\ \Lambda \\ M \\ S \end{bmatrix} = \begin{bmatrix} C^T \\ b \end{bmatrix} \qquad (1.147)$$

$\mu_j x_j = 0 = \lambda_i s_i$ для всех i и j , $X, \Lambda, M, \ S \geq 0$.

Все уравнения, за исключением $\mu_j x_j = 0 = \lambda_i s_i$, являются линейными относительно переменных $X, \Lambda, M, \ S$. Следовательно, исходная задача сводится к нахождению решения системы линейных уравнений, удовлетворяющих дополнительным ограничениям $\mu_j x_j = 0 = \lambda_i s_i$.

По предположению функция z строго вогнутая, а область допустимых решений – выпуклое множество. То есть, решение, удовлетворяющее всем перечисленным условиям, должно быть единственным и оптимальным.

Решение рассматриваемой системы уравнений можно найти путем реализации первого этапа двухэтапного метода (метода вспомогательной задачи). Однако необходимость удовлетворения ограничениям $\mu_j x_j = 0 = \lambda_i s_i$ требует некоторой модификации алгоритма симплекс-метода:

- если переменная μ_j входит в состав базисных переменных очередного опорного решения, то переменную x_j нельзя вводить в базис;
- если переменная x_j входит в состав базисных переменных очередного опорного решения, то переменную μ_j нельзя вводить в базис;
- если переменная λ_i входит в состав базисных переменных очередного опорного решения, то переменную s_i нельзя вводить в базис;
- если переменная s_i входит в состав базисных переменных очередного опорного решения, то переменную λ_i нельзя вводить в базис.

Для того чтобы использовать первый этап двухэтапного метода для решения системы уравнений (1.137), необходимо провести предварительный анализ системы уравнений (1.137) на предмет выявления отрицательных свободных членов.

Если таковые имеются, левую и правую части соответствующих уравнений нужно умножить на "-1". Только после этого можно вводить искусственные переменные.

В противном случае вспомогательная задача может оказаться недопустимой или решена неправильно.

Итак, поиск оптимального решения исходной задачи

$$\begin{cases} z = CX + X^T DX \to \max \\ G(X) = \begin{bmatrix} A \\ -E \end{bmatrix} X - \begin{bmatrix} b \\ 0 \end{bmatrix} \le 0 \end{cases}$$

сводится к решению следующей задачи линейного программирования:

$$z'' = -R \to \max$$

$$\begin{bmatrix} -2D & A^T & -E & E & 0 \\ A & 0 & 0 & 0 & E \end{bmatrix} \begin{bmatrix} X \\ \Lambda \\ M \\ R \\ S \end{bmatrix} = \begin{bmatrix} C^T \\ b \end{bmatrix} \qquad (1.148)$$

$$X, \Lambda, M, R, S \ge 0$$

при дополнительных ограничениях $\mu_j x_j = 0 = \lambda_i s_i$ для всех i и j ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$).

Здесь $R = (r_1, r_2, \ldots, r_n)$ – вектор искусственных переменных, введенных в каждое из n первых уравнений.

Таким образом, система ограничений вспомогательной задачи имеет полный единичный базис.

Следует еще раз отметить особенность этой задачи. Она всегда имеет решение (вспомним двухэтапный метод: целевая функция вспомогательной задачи z'' ограничена сверху). Кроме того, в случае вогнутости целевой функции исходной задачи $z = CX + X^T DX$, на оптимальном решении задачи (1.148) должно иметь место $z'' = 0$.

Пример 1.33

Решить задачу квадратичного программирования:

$$z = 6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2 \to \max$$

$$x_1 + x_2 \le 1$$

$$2x_1 + 3x_2 \le 4$$

$$x_1, x_2 \ge 0.$$

Здесь

$C = (6, 3)$, , $A = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix}$, $D^* = \begin{bmatrix} -2 & -4 \\ 0 & -3 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Придадим матрице D^* симметрическую форму: $D = \begin{bmatrix} -2 & -2 \\ -2 & -3 \end{bmatrix}$

$\Lambda = (\lambda_1, \lambda_2)^{\mathrm{T}}$ – множители Лагранжа, соответствующие ограничениям $AX \le b$;

$M = (\mu_1, \mu_2)^{\mathrm{T}}$ – множители Лагранжа, соответствующие ограничениям $X \ge 0$.

Построим задачу линейного программирования (4.44) в терминах решаемой задачи.

Наша задача имеет вид:

$z'' = -r_1 - r_2 \to max$,

$$\begin{bmatrix} 4 & 4 & 1 & 2 & -1 & 0 & 1 & 0 & 0 & 0 \\ 4 & 6 & 1 & 3 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \\ \mu_1 \\ \mu_2 \\ r_1 \\ r_2 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 1 \\ 4 \end{bmatrix}$$

$x_1, x_2, \lambda_1, \lambda_2, \mu_1, \mu_2, r_1, r_2, s_1, s_2 \ge 0.$

При дополнительных условиях: $\mu_j x_j = 0 = \lambda_i s_i$:

$$\Leftrightarrow \mu_1 x_1 = \mu_2 x_2 = \lambda_1 s_1 = \lambda_2 s_2 = 0.$$

Эта (вспомогательная) задача имеет решение:

$z''_{opt} = 0$; $x_1=1$, $x_2=0$, $l_1=2$, $l_2=0$, $\mu_1=0$, $\mu_2=3$, $r_1=0$, $r_2=0$, $s_1=0$, $s_2=2$.

Как видно, дополнительные условия $\mu_1 x_1=\mu_2 x_2=\lambda_1 s_1=\lambda_2 s_2= 0$ выполнены. По этому решению находим оптимальное решение исходной задачи:

$z_{opt} = 4$; $x_1=1$, $x_2=0$.

Рассмотрим особенность решения задачи линейного программирования (1.148).

Одновременное включение всех ограничений вида (1.145) неэффективно с вычислительной точки зрения, поскольку каждое такое ограничение требует дополнительной проверки состава базиса на каждой итерации симплекс-метода, при принятии решения о вводимой в базис переменной. С другой стороны, часть (как правило – большая) этих ограничений выполняется автоматически. В этой связи весьма эффективным представляется следующий итерационный алгоритм решения задачи (1.148).

Обозначим через Ω множество пар переменных, для которых должны выполняться дополнительные ограничения. В предельном случае это множество будет включать все пары:

$$\Omega = \{(\mu_j x_j), (\lambda_i s_i) \mid \ i =1, 2, ..., m;\ j =1, 2, ..., n \}.$$

<u>Шаг 1 .</u> Положить $\Omega = \varnothing$.

<u>Шаг 2 .</u> Решить задачу с дополнительными ограничениями Ω.

<u>Шаг 3 .</u> Если $\mu_j x_j = 0 = \lambda_i s_i$ для всех i и j ($i =1, 2, \ldots, m$; $j =1, 2, \ldots, n$), то конец – найдена точка, удовлетворяющая условиям Куна-Таккера. В противном случае выполнить следующий шаг.

<u>Шаг 4 .</u> $\Omega = \Omega \cup \{(\mu_j x_j), (\lambda_i s_i) \mid \ \mu_j x_j \neq 0,\ \lambda_i s_i \neq 0;\ \ i = \overline{1,m};\ \ j = \overline{1,n} \}$.

Выполнить шаг 2.

В рассмотренном примере задача была решена при $\Omega = \varnothing$: шаг 4 не потребовался. В общем же случае для удовлетворения дополнительных условий может потребоваться несколько итераций.

Основная проблема, связанная с применимостью приведенного подхода к решению задачи квадратичного программирования, заключается в том, целевая функция должна быть вогнутой (в задаче на максимум).

В общем случае анализ на вогнутость является весьма трудоемкой задачей.

Пример 1.34

Рассмотрим следующие модели, которые отличаются друг от друга только коэффициентами ЦФ при произведении $x_1 x_2$:

$$z = -2x_1^2 - 2x_2^2 + 2x_1 x_2 \to \max$$

$$x_1 \ge 1$$

$$x_2 \ge 1$$

$$x_1, x_2 \ge 0.$$

Квадратичная форма $Q(X) = X^T D X = \left(x_1, x_2\right)\begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Здесь D преобразована к симметрическому виду.

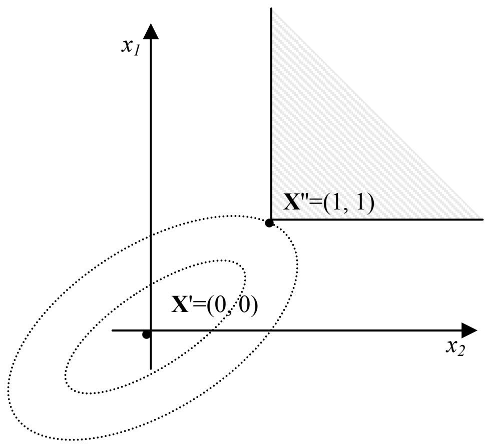
Вспомним, что квадратичная форма $Q(X)$ будет отрицательно определенной, если значения k-х угловых миноров Δ_k определителя $|D|$ отличны от нуля и имеют знак $(-1)^k$, k=1, 2, … , n. В этом случае матрица D будет отрицательно определенной.

Найдем все угловые миноры:

$\Delta_1 = -2 < 0$ (знак $(-1)^1$), $\quad \Delta_2 = \begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix} = 3 > 0$ (знак $(-1)^2$).

То есть матрица D отрицательно определенная, и соответственно, функция $Q(X)$ является строго вогнутой. Задача имеет единственное решение и это решение – оптимальное: $Z^{opt} = -2,\ x_1 = 1,\ x_2 = 1$.

Ниже приведена геометрическая интерпретация решения задачи.



Пример 1.35

$$z = -2x_1^2 - 2x_2^2 + 4x_1x_2 \to \max$$

$$x_1 \geq 1$$

$$x_2 \geq 1$$

$$x_1, x_2 \geq 0.$$

Найдем все угловые миноры:

$$\Delta_1 = -2 > 0 \ \ (\text{знак } (-1)^1), \qquad \Delta_2 = \begin{vmatrix} -2 & 2 \\ 2 & -2 \end{vmatrix} = 0\ .$$

То есть матрица D отрицательно полуопределенная, и соответственно, функция $Q(X)$ является не строго вогнутой. Одно из оптимальных решений задачи: $Z^{opt} = 0,\ x_1 = 1,\ x_2 = 1$. Заметим, что любое из решений, для которых $x_1 = x_2 \geq 0$, является оптимальным: на всех таких решениях ЦФ имеет нулевое значение.

Пример 1.36

$$z = -2x_1^2 - 2x_2^2 + 8x_1x_2 \to \max$$

$$x_1 \geq 1$$

$$x_2 \geq 1$$

$$x_1, x_2 \geq 0.$$

Найдем все угловые миноры:

$$\Delta_1 = -2 < 0 \ \ (\text{знак } (-1)^1), \qquad \Delta_2 = \begin{vmatrix} -2 & 4 \\ 4 & -2 \end{vmatrix} = -12 < 0.$$

То есть матрица D неопределенная, и соответственно, функция $Q(X)$ не является ни вогнутой, ни выпуклой. Задача не относится к классу задач квадратичного программирования и, более того, не является задачей выпуклого программирования.

Приведенные примеры объясняют необходимость предварительного анализа целевой функции на вогнутость (в задаче максимизации). Если в результате такого анализа будет установлено, что матрица D положительно определена или положительно полуопределена, можно просто изменить знак целевой функции (изменить направление экстремизации). При этом матрица D станет отрицательно определенной или отрицательно полуопределенной.

Контрольные вопросы и задачи к разделу 1.6

1. Почему в квадратичном программировании необходимо вычислять определенность квадратичной формы?

2. Можно ли решать задачу квадратичного программирования методом сведения к задаче линейного программирования на максимум (минимум) в случае отрицательной (положительной) полуопределенности квадратичной формы? Чем будет отличаться результат от случая полной определенности квадратичной формы?

3. Получите с помощью метода множителей Лагранжа необходимые условия экстремума для следующей задачи нелинейного программирования:

$$x_1^3 + 2x_2^2 + x_1x_2 - e^{x_1} + x_3 \to \max$$

$$x_1^2 + x_2^2 + x_2^2x_3^2 = 25$$

$$x_1^2x_2^2 + e^{x_2} + 2x_3 = 5$$

4. Сведите задачу квадратичного программирования к задаче ЛП и исследуйте её квадратичную форму на определенность

$$6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2 \to \max$$

$$x_1 + x_2 \le 1$$

$$2x_1 + 3x_2 \le 4$$

5. Сведите задачу квадратичного программирования к задаче ЛП и исследуйте её квадратичную форму на определенность

$$x_1^2 + 3x_2^2 - 2x_1x_2 \to \max$$

$$x_1 + 2x_2 \le 10$$

$$x_1 - 3x_2 \le 5$$

2. ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

Дискретное программирование (ДП) – один из наиболее важных и актуальных разделов математического программирования. Значительный круг задач планирования, принятия решений и технических задач сводится к выбору лучших параметров из некоторой дискретной совокупности заданных величин. Поскольку задачи ДП являются подмножеством задач математического программирования (МП), их постановка в общем виде выглядит также:

$$\varphi_0(x_1, x_2, ..., x_n) \to extr$$

$$\varphi_i(x_1, x_2, ..., x_n) \le 0, \; i = \overline{1, m}$$

$$x \in D.$$

Главным свойством, позволяющим отнести задачу к классу задач дискретного программирования, является дискретность области определения всех или нескольких параметров.

Несмотря на кажущуюся простоту постановки задач ДП, математические трудности, возникающие при их анализе, могут быть весьма значительными.

Достаточно вспомнить Великую Теорему Пьера Ферма. Суть этой теоремы сводится к следующему [2].

Диофантово уравнение $x^n + y^n = z^n$, где n – целое число, большее двух, не имеет решения в целых положительных числах.

Эта теорема может быть сформулирована, как следующая задача целочисленного программирования:

$$(x_1^{x_4} + x_2^{x_4} - x_3^{x_4})^2 \to \min$$

$$x_1 \ge 1; \; x_2 \ge 1; \; x_3 \ge 1; \; x_4 \ge 3$$

$$x_j - целые \; (j = \overline{1, 4}).$$

Любой метод, позволяющий решить эту задачу в обозримое время, будет конструктивным методом анализа теоремы Ферма:

если *min* ЦФ>0, то теорема доказана;

если *min* ЦФ=0, то утверждение теоремы Ферма опровергнуто.

2.1. Некоторые виды задач дискретного программирования

2.1.1. Линейное целочисленное программирование

Главным образом, в линейном целочисленном программировании исследуется модель следующего вида:

$$\sum_{j=1}^{n} c_j x_j \to \max \ (\min) \qquad (2.1)$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i \quad (i=\overline{1,m}) \qquad (2.2)$$

$$x_j \ge 0 \quad (j=\overline{1,n}) \qquad (2.3)$$

$$x_j - \textit{целые} \ \ (j=\overline{1,n_1}). \qquad (2.4)$$

При $n=n_1$ имеет место *задача линейного целочисленного программирования*; при $n>n_1$ *задача линейного частично-целочисленного программирования.*

Условие целочисленности (2.4) может быть заменено требованием *дискретности*:

$$x_j \in \{d_1^j, d_2^j, \ldots, d_{k_j}^j\} \quad (j=\overline{1,n_1}).$$

В этом случае имеет место задача *линейного программирования с дискретными переменными.*

Частным случаем задачи линейного целочисленного программирования (ЛЦП) является задача линейного целочисленного программирования *с булевыми переменными* ЛЦП(б):

$$\sum_{j=1}^{n} c_j x_j \to \max \ (\min)$$

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

$$x_j \in \{0,1\} \quad (j=\overline{1,n}).$$

Это – очень важный класс задач. Достаточно сказать, что если заранее установлены пределы возможных значений переменных, любая задача дискретного или целочисленного программирования может быть сведена к задаче ЛЦП(б).

2.1.2. Сведение к задачам булева программирования задач линейного программирования с дискретными переменными

Пусть имеется задача линейного программирования с дискретными переменными:

$$\sum_{j=1}^{n} c_j x_j \to \max \ (\min)$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i \quad (i = \overline{1,m})$$

$$x_j \in \{d_1^j, d_2^j, ..., d_{k_j}^j\}$$

$$x_j \ge 0 \quad (j = \overline{1,n}).$$

Введем булевы переменные (по одной переменной на каждое дискретное значение):

$y_1^j, y_2^j, ..., y_{k_j}^j$, $y_l^j \in \{0,1\}$, $l = \overline{1,k_j}$, $j = \overline{1,n}$.

Потребуем выполнения следующих условий:

$$\sum_{l=1}^{k_j} y_l^j = 1 \quad (j = \overline{1,n}). \tag{2.5}$$

Произведем замену переменных:

$$x_j = \sum_{l=1}^{k_j} d_l^j y_l^j.$$

Учитывая (2.5), всегда будет иметь место:

$\sum_{l=1}^{k_j} d_l^j y_l^j \in \{d_1^j, d_2^j, ..., d_{k_j}^j\}$ – то, что и требуется. Окончательно модель приобретает вид:

$$\sum_{j=1}^{n} \left[c_j \sum_{l=1}^{k_j} \left(d_l^j y_l^j \right) \right] \to \max \ (\min)$$

$$\sum_{j=1}^{n} a_{ij} \sum_{l=1}^{k_j} d_l^j y_l^j \le b_i \quad (i = \overline{1,m})$$

$$\sum_{l=1}^{k_j} y_l^j = 1, \ j = \overline{1,n},$$

$$y_l^j \in \{0,1\}, \ l = \overline{1,k_j}, \ j = \overline{1,n}.$$

2.1.3. Сведение к задачам булева программирования некоторых нелинейных задач с дискретными переменными

Отдельные нелинейные задачи с дискретной областью определения переменных могут быть сведены к виду ЛЦП(б)-задачи.

Пусть имеется задача, в которой $x_j \in \{d_1^j, d_2^j, ..., d_{k_j}^j\}$, $x_j \geq 0 \;\; (j = \overline{1,n})$.

Пусть далее в целевой функции или в ограничениях фигурирует нелинейная функция $\phi(x_j)$ (рис. 2.1). Эта функция, в частности, может быть задана таблично.

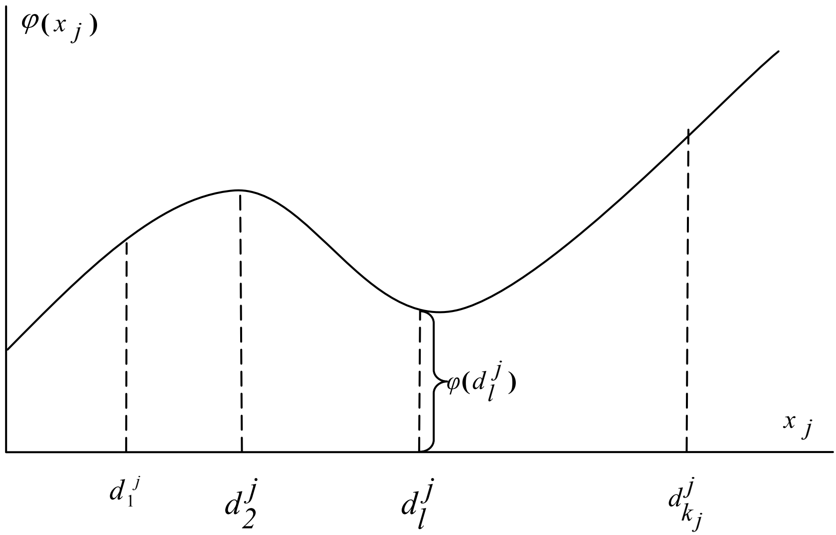


Рис.2.1. Некоторая нелинейная функция

То есть

$$\phi(x_j) \in \left\{\phi(d_1^j), \phi(d_2^j), ..., \phi(d_{k_j}^j)\right\}. \tag{2.6}$$

Введем булевы переменные:

$y_1^j, y_2^j, ..., y_{k_j}^j$, $\;y_l^j \in \{0,1\}$, $\;l = \overline{1,k_j}$, $\;j = \overline{1,n}$.

Представим $\varphi(x_j)$ в виде:

$$\varphi(x_j)=\sum_{l=1}^{k_j}\varphi(d_l^j)y_l^j \ .$$

Теперь, для того чтобы $\varphi(x_j)$ принимало одно и только одно значение из множества (2.6), введем ограничение:

$$\sum_{l=1}^{k_j} y_l^j = 1 .$$

Произведем замену переменной x_j :

$$x_j = \sum_{l=1}^{k_j} d_l^j y_l^j \ .$$

Пример 2.1

Дана нелинейная задача:

$$\varphi(x_1) + \sum_{j=2}^{n} c_j x_j \to \max$$

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

$$x_j \in \{1,\ 2,\ 3,\ 4\} \ (j = \overline{1,n}).$$

Свести задачу к задаче линейного программирования с булевыми переменными.

- Вводим булевы переменные y_1, y_2, y_3, y_4 ; $y_l \in \{0,1\}$, $l=\overline{1,4}$.
- Представляем x_1 в виде: $x_1 = \sum_{l=1}^{4} l y_l$.
- Потребуем, чтобы только одна булева переменная принимала единичное значение: $\sum_{l=1}^{4} y_l = 1$.
- Представляем $\varphi(x_j)$ в виде: $\varphi(x_j) = \sum_{l=1}^{4} \varphi(l) y_l$.

Окончательно:

$$\sum_{l=1}^{4}\varphi(l)y_l + \sum_{j=2}^{n}c_j x_j \to \max$$

$$a_{i1}\sum_{l=1}^{4} l y_l + \sum_{j=2}^{n}a_{ij}x_j \le b_i \quad (i=\overline{1,m})$$

$$\sum_{l=1}^{4} y_l = 1$$

$$x_j \in \{1,\ 2,\ 3,\ 4\} \ (j=\overline{2,n})$$

$$y_l \in \{0,1\} \ l=\overline{1,4}.$$

Если оптимальное решение задачи представлено вектором:

$$x^* = \left(x_2^*, x_3^*, \ldots x_n^*, y_1^*, y_2^*, y_3^*, y_4^*\right).$$

По этому решению определяется оптимальное значение переменной x_1 исходной задачи:

$$x_1 = \sum_{l=1}^{4} l y_l^* .$$

2.1.4. Задачи комбинаторного типа

В задачах комбинаторного типа поиск экстремума ЦФ осуществляется на некотором множестве комбинаций элементов заданного множества. В качестве таких комбинаций могут выступать, например, перестановки, сочетания, последовательности элементов.

Как правило, множество таких комбинаций, в соответствии с содержательной постановкой задач, конечно.

Приведем самую общую постановку экстремальной задачи комбинаторного типа.

Пусть задано конечное множество G некоторых комбинаций x_i $(i=1,2,\ldots,N)$.

На множестве G определена некоторая функция $f(x_i)$. Эта функция может быть задана аналитически, алгоритмически и т.д. Главное – это есть способ вычисления $f(x_i)$ для любой комбинации $x_i \in G$. Необходимо найти комбинацию, $x_k \in G$, на которой ЦФ достигает минимума (или максимума):

$$f(x_k) = \max\{f(x_i) \mid x_i \in G\}.$$

К экстремальным задачам комбинаторного типа относятся задачи ЛЦП с ограниченным допустимым множеством.

Что характерно для этих задач? Характерно то, что, несмотря на конечность множества допустимых комбинаций G, мощность этого множества для реальных задач столь высокая, что прямой перебор для поиска оптимальной комбинации становится нереализуемым.

Например, в задаче о назначениях $\|G\|=n!$; в задаче о коммивояжере $\|G\|=(n-1)!$.

Ряд задач комбинаторного типа в принципе можно свести к ЛЦП(б)-задачам. Однако задачи комбинаторного типа, представленные моделями ЛЦП или ЛЦП(б)-задач, имеют очень большую размерность, что затрудняет из решение.

2.1.5. Примеры прикладных задач дискретного программирования

Задача о ранце

В этой задаче речь идет о собравшемся в поход путешественнике, который должен упаковать в ранец (рюкзак) различные полезные предметы n наименований, причем могут потребоваться несколько одинаковых предметов одного и того же наименования.

Имеется m ограничений такого типа, как вес, объем, линейные размеры и т.д.

Пусть a_{ij} – i-я характеристика предмета j-го наименования ($i=\overline{1,m}$; $j=\overline{1,n}$); b_i – ограничение по i-й характеристике предмета (по весу, объему, и т.д.), $i=\overline{1,m}$.

Обозначим через x_j количество предметов предмета j-го наименования ($j=\overline{1,n}$), запланированное к загрузке в ранец.

Предполагается, что известна «полезность» c_j одного предмета j-го наименования ($j=\overline{1,n}$).

Тогда модель соответствующей задачи приобретает следующий вид:

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

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

$$x_j \ge 0 \quad , \ x_j - \text{целые} \ \ (j = \overline{1,n}).$$

Это уже не задача ЛП, так как требование целочисленности не выражается линейными ограничениями[16].

Задача о назначениях

Имеется n различных самолетов, которые требуется распределить между n авиалиниями.

Известно, что на j-той авиалинии i-й самолет будет приносить доход c_{ij} единиц.

Требуется так распределить самолеты, чтобы максимизировать суммарный доход. При этом, каждый самолет должен быть закреплен за соответствующей авиалинией, и на каждую авиалинию должен быть назначен самолет.

Введем булевы переменные x_{ij} $(i, j = \overline{1,n})$ такие, что:

$x_{ij}=1$, если i-й самолет направляется на на j-ю авиалинию;

$x_{ij}=0$, если i-й самолет не направляется на на j-ю авиалинию.

Тогда модель этой задачи приобретает следующий вид:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \to \max ,$$

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = \overline{1,n}$$

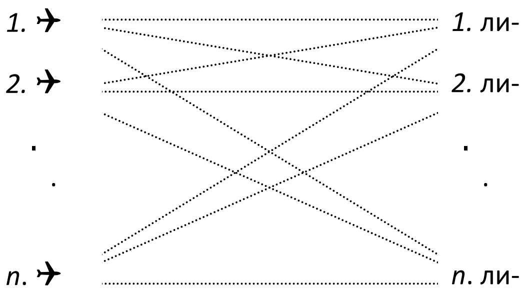
(каждый самолет назначается только на одну линию),

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = \overline{1,n}$$

(на каждую линию назначается только один самолет)

$$x_{ij} \in \{0,1\} ; \ i, j = \overline{1,n} .$$

[16] Первоначально эта задача была поставлена, как задача об оптимальной загрузке бомбардировщика бомбовым грузом.



Задача о коммивояжере

Имеется $n+1$ городов, пронумерованных числами $0,1,2,\dots n$.

Выехав из города «*0*», коммивояжер должен объехать все остальные города, побывав в каждом из них по одному разу, и вернуться в город «*0*».

Известны расстояния c_{ij} между городами i и j $(i, j = \overline{0,n})$.

В задаче требуется найти самый короткий маршрут.

Введем булевы переменные x_{ij} $(i, j = \overline{0,n};\ i \neq j)$:

$$x_{ij} = \begin{cases} 1, \text{ если в маршрут включен переезд из } i \text{ в } j; \\ 0 - \text{в противном случае.} \end{cases}$$

Приходим к модели:

$$\sum_{\substack{i,\, j=\overline{0,n} \\ i \neq j}} c_{ij} x_{ij} \to \min \tag{2.7}$$

$$\sum_{j=0}^{n} x_{ij} = 1,\ i = \overline{0,n},\ i \neq j \tag{2.8}$$

$$\sum_{i=0}^{n} x_{ij} = 1,\ j = \overline{0,n},\ i \neq j \tag{2.9}$$

$$x_{ij} \in \{0,1\};\ i, j = \overline{0,n},\ i \neq j. \tag{2.10}$$

Из условий (2.8), (2.9) и (2.10) следует, что из каждого города коммивояжер выезжает точно один раз (2.8) и в каждый город въезжает тоже точно один раз (2.9). Однако этого не достаточно. Эти условия не обеспечивают связности маршрута. Например, маршрут, представленный на рис.2.2, удовлетворяет всем ограничениям, но лишен смысла.

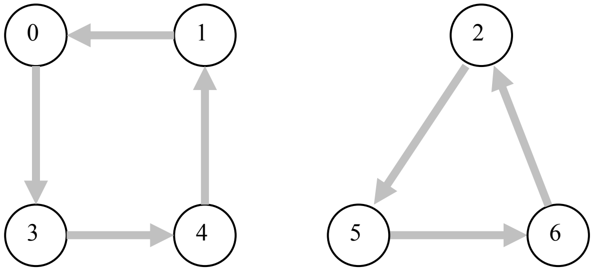


Рис.2.2. Пример некорректного маршрута

По условиям задачи маршрут должен представлять собой единственный цикл, обязательно проходящий через город «0» – исходный город.

Более точно, условия задачи должны быть дополнены такими условиями, которые запрещают любой цикл, не проходящий через город «*0*».

Дополним ограничения (2.8) и (2.9) ограничениями, которые на первый взгляд могут показаться несколько «искусственными»:

$$u_i - u_j + nx_{ij} \le n - 1; \quad i, j = \overline{1,n}, \ i \ne j. \qquad (2.11)$$

Здесь u_i – переменные, которые могут принимать произвольные значения ($i = \overline{1,n}$).

Покажем, что эти дополнительные ограничения запрещают любой цикл, не проходящий через город «*0*».

Действительно, рассмотрим некоторое решение, удовлетворяющее ограничениям (2.8)-(2.11):

$$(\{x_{ij}\}, \{u_i\}).$$

Поставим в соответствие этому решению маршрут такой, что дуга (i,j) принадлежит этому маршруту тогда и только тогда, когда $x_{ij} = 1$.

Вполне очевидно, что маршрут содержит цикл, так как в каждый город входит одна дуга и одна дуга выходит.

Предположим (от противного), что маршрут включает цикл, не содержащий города «*0*», состоящий из k дуг.

Очевидно, что каждой дуге этого цикла соответствует определенное неравенство системы (2.11), так как $i \neq 0$ и $j \neq 0$ (город "*0*" не входит в маршрут).

Просуммируем неравенства, соответствующие дугам частного цикла. При этом все переменные u_i сократятся, так как каждая из них войдет в сумму дважды с противоположными знаками.

Будет получено противоречивое неравенство:

$$nk \le (n-1)k\ ,$$

что доказывает неправомерность предположения о существовании такого цикла.

Таким образом, любое решение задачи (2.7)-(2.11) соответствует циклу, проходящему через город «0».

Покажем теперь, что любому циклу, проходящему через город «**0**», можно поставить в соответствие решение задачи (2.7)-(2.11).

Возьмем произвольный цикл:

$$l_0 \to l_1 \to l_2 \to \cdots \to l_n \to l_0\,,$$

где l_0 =*0*, а все l_r различны ($l_r \in \{1,2,...,n\}$, $r \in \{1,2,...,n\}$), т.е. r –это порядковый номер города l_r в маршруте, представленном циклом.

Рассмотрим следующий план задачи: $\boldsymbol{X}$=({ x_{ij} },{ u_i }). В этом плане числа x_{ij} возьмем в соответствии с их интерпретацией в модели. А именно, будем считать, что x_{ij} =*1*, если существует $r \in \{1,2,...,n\}$ такое, что $l_{r-1} = i$ и $l_r = j$ (т.е. в маршруте имеет место переезд из города *i* в город *j*) . В противном случае x_{ij} =*0*.

В качестве u_i примем порядковый номер города l_{u_i} в маршруте ($i \neq 0, u_i \in \{1,2,...,n\}$).

Определенный таким образом план удовлетворяет всем ограничениям задачи (2.7)-(2.11).
Действительно:
1. Возьмем x_{ij} =*0*. Тогда соответствующее ограничение системы (2.11) будет иметь вид:

$$u_i - u_j \le n-1,\ i \ne j\ .$$

Это ограничение всегда будет выполняться, так как $1 \le u_i \le n$ для всех $i \in \{1, 2, ..., n\}$.

2. Возьмем $x_{ij} = 1$. Это значит, что существует такое число $r \in \{1, 2, ..., n-1\}$, что $i = l_r$ и $j = l_{r+1}$. Тогда $u_i = r$, а $u_j = r+1$, следовательно

$$u_i - u_j + nx_{ij} = r - (r+1) + n \le n-1\,,$$

т.е. ограничение выполняется.

Таким образом, доказано, что любому маршруту соответствует допустимое решение задачи.

2.1.6. Методы решения дискретных задач

Обсудим изложенные задачи ЛЦП. Формальная запись каждой из них отличается от обычных задач ЛП *только* дополнительным ограничением на значения переменных. Однако именно это требование (требование целочисленности переменных) является существенным препятствием для исследования и решения подобных задач.

Самым естественным путем была попытка использования обычных, но несколько измененных методов линейного программирования.

Так, может показаться, что решение ЛЦП-задачи можно получить следующим образом:

- решить эту задачу любым из методов решения ЛП-задач (без требования целочисленности переменных);
- координаты полученного решения округлить до целых чисел.

Однако даже простые примеры показывают несостоятельность такого наивного "инженерного" подхода.

Пример 2.2

$$x_1 + 5x_2 \to \max,$$
$$10x_1 - 10x_2 \ge 1,$$
$$2x_1 + 2x_2 \le 9,$$
$$x_j \ge 0, \text{ целые, } j = 1, 2.$$

Решаем задачу без требования целочисленности: ЦФ=**13.3**

	$\mathbf{x_1}$=2.3	$\mathbf{x_2}$=2.2
I	**2**	**2**
II	**2**	**3**
III	**3**	**2**
IV	**3**	**3**

Ни один из этих вариантов не является планом задачи и весьма далек до оптимального ее решения (рис.2.3):

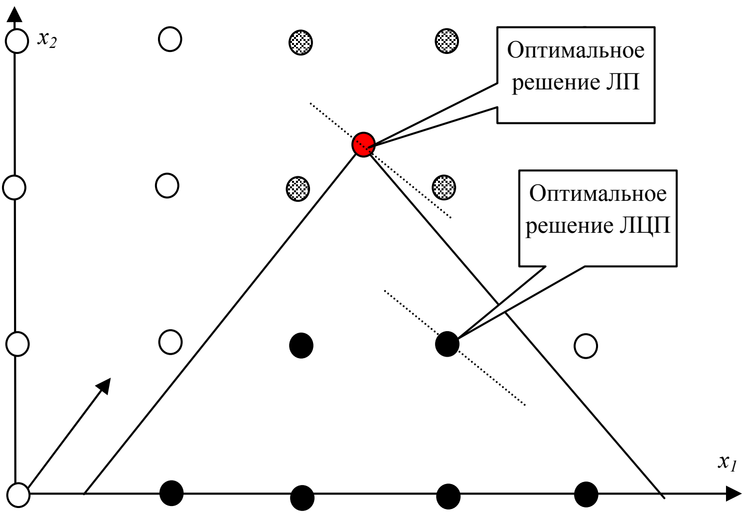


Рис.2.3. Графическая иллюстрация примера

Тем не менее, некоторые из целочисленных задач могут быть непосредственно сведены к задачам ЛП. Если это принципиально возможно, так и следует поступать. Так, в разделе 2.2 будет рассмотрена транспортная задача, основным свойством которой является целочисленность получаемого решения в случае целочисленности всех числовых параметров задачи.

Вместе с тем, задачу ЛЦП более общего вида, например, задачу о ранце, уже нельзя свести к обычной ЛП-задаче.

Именно поэтому теория решения ЛЦП-задач развивалась по двум направлениям.

Первое направление, ориентированное на решение общей задачи ЛЦП, носит название "Метод отсечения". Основная идея метода отсечения – сопоставление задаче ЛЦП обычной задачи ЛП, решение которой позволяет найти решение ЛЦП-задачи или убедиться в неразрешимости последней.

Второе направление состоит в разработке для специфических задач ЛЦП принципиально новых комбинаторных приемов. Наиболее популярные методы этого направления относятся к группе методов, известной под объединяющим названием "Метод ветвей и границ". Здесь также в некоторых случаях используются свойства задач ЛП.

Контрольные вопросы и задачи к разделу 2.1

1. Напишите основную форму задачи целочисленного программирования.

2. Чем отличается задача целочисленного программирования от задачи частично-целочисленного программирования?

3. Какой вид должна иметь нелинейная функция в задаче нелинейного целочисленного программирования, чтобы эту задачу можно было свести к задаче линейного целочисленного программирования?

4. Почему задачу целочисленного программирования некорректно решать с помощью симплекс-метода с простым округлением результата?

5. Постройте задачу частично-целочисленного программирования

Два завода З1 и З2, принадлежащие одной и той же компании, производят три сорта алюминия – А1, А2 и А3. При производстве алюминия используется боксит и электроэнергия. Запасы боксита холдинга составляют 10 тыс тонн. Боксит можно приобрести на рынке в виде 500-тонных поставок по цене 6 млн руб. за одну партию, но не более 10 партий. Стоимость 1 киловатт-часа электроэнергии составляет 2 рубля. Затраты сырьевых и энергетических ресурсов на производство, а также стоимость одной тонны алюминия каждого сорта сведены в таблице. Определить план выпуска сортов алюминия на каждом заводе, приносящий максимальную прибыль.

Сорт алюминия	Цена, руб./т	Затраты ресурсов на 1 тонну продукции			
		З1		З2	
		Боксит, т	Электроэнергия, кВт*ч	Боксит,т	Электроэнергия, кВт*ч
А1	60000	2	1000	2.1	950
А2	65000	2.3	800	2.2	600
А3	80000	2.7	700	2.5	400

6. Сведите к задаче линейного целочисленного программирования с булевыми переменными следующую задачу дискретного программирования:

$$x_1 + 3x_2 - x_3 \to \max$$
$$x_1 + x_2 + x_3 \le 10$$
$$x_2 \le 5$$
$$x_1 - x_3 \ge 2$$
$$x_2 \ge 0$$
$$x_1 \in \{2,5,7\}\, x_3 \in \{0,3,4\}.$$

7. Сведите к задаче линейного целочисленного программирования с булевыми переменными следующую задачу нелинейного программирования:

$$\sin(x_1) + x_2^3 \to \max$$
$$x_1 - x_2 \le 3$$
$$x_1 \le 8$$
$$x_1 \in \{1,3,5\}\, x_2 \in \{1,2,3\}.$$

8. Постройте линейную целочисленную модель задачи коммивояжера для следующей матрицы расстояний:

Город	**1**	**2**	**3**
1	-	2	5
2	3	-	2
3	8	1	-

9. Решите следующую задачу линейного целочисленного программирования, заменив целочисленные переменные на вещественные, применив симплекс-метод, и округлив результат. Постройте допустимое множество на графике и проверьте, попадает ли полученное округленное решение в допустимое множество:

$$x_1 + x_2 \to \max$$
$$9x_1 + x_2 \le 54$$
$$18x_1 - 10x_2 \ge 54$$
$$x_1, x_2 \ge 0. \text{ *целые.*}$$

2.2. Транспортная задача

2.2.1. Постановка транспортной задачи

Многие классы широко распространенных задач ЛП обладают особенностями, которые выражаются в специфическом строении матрицы коэффициентов системы ограничений и ЦФ. Эти особенности позволяют существенно упростить общий метод решения именно этих задач. При этом ускоряется процесс получения оптимального решения.

Ярким примером задачи со специальной структурой является *транспортная задача*.

Содержательно транспортная задача имеет следующую постановку.

Имеется m пунктов производства некоторого однородного продукта (например, угля, цемента и т.д.).

Обозначим эти пункты $A_1, A_2, \ldots, A_m$..

Имеется также n пунктов потребления этого продукта $B_1, B_2, \ldots, B_n$..

В пункте A_i $(i = \overline{1,m})$ сосредоточено a_i единиц продукции.

Потребность пункта B_j $(i = \overline{1,m})$ составляет b_j единиц.

Известна стоимость перевозки единицы продукции из пункта A_i $(i = \overline{1,m})$ в пункт B_j $(i = \overline{1,m}) - c_{ij}$.

Встречные перевозки запрещены, поэтому все $x_{ij} \ge 0$.

В этой задаче требуется составить такой план перевозки продукции из пунктов производства в пункты потребления, который удовлетворяет следующим требованиям:

1. Суммарная стоимость всех перевозок была минимальной;
2. Вся продукция из всех пунктов производства должна быть вывезена;

3. Потребности всех пунктов потребления должны быть удовлетворены, а именно, в каждый пункт потребления должно быть завезено столько продукции, сколько ему требуется – не больше и не меньше.

Начнем строить модель.

Обозначим x_{ij} – объем перевозки продукции из пункта A_i ($i = \overline{1,m}$) в пункт B_j ($i = \overline{1,m}$).

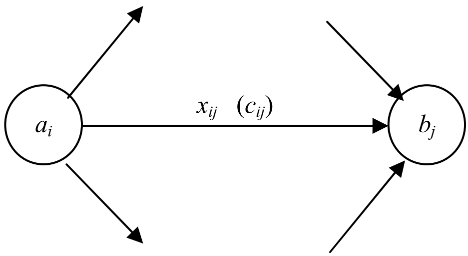


Рис.2.4 Связь между пунктом производства и пунктом потребления

Все данные занесем в табл.2.1.

Таблица 2.1. Транспортная таблица

j / i	1	2		n	Запасы
1	c_{11} x_{11}	c_{12} x_{12}		c_{1n} x_{1n}	a_1
2	c_{21} x_{21}	c_{22} x_{22}		c_{1n} x_{2n}	a_2
m	c_{m1} x_{m1}	c_{m2} x_{m2}		c_{mn} x_{mn}	a_m
Потребности	b_1	b_2		b_n	d

А теперь построим модель:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min \tag{2.12}$$

Вывоз продукции $\sum_{j=1}^{n} x_{ij} = a_i, \ i = \overline{1,m}$ (2.13)

Ввоз продукции $\sum_{i=1}^{m} x_{ij} = b_j, \ j = \overline{1,n}$ (2.14)

$$x_{ij} \geq 0, \ i = \overline{1,m}, \ j = \overline{1,n}. \tag{2.15}$$

Естественно, что в такой трактовке задачи должно выполняться еще одно условие:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = d - \tag{2.16}$$

иначе задача не будет иметь решения.

Задачу (2.12)-(2.15) принято называть *закрытой транспортной задачей*. При этом, если условие (2.16) выполняется, имеет место *сбалансированная транспортная модель*.

Если условие (2.16) не выполняется, то, очевидно, в некоторых пунктах производства остается не вывезенная продукция или потребность в продукции некоторых пунктов потребления не удовлетворяется. Имеет место *открытая транспортная задача*, для которой модель (2.12)-(2.15) не подходит.

Это – случай *несбалансированной транспортной модели*: возможности не соответствуют потребностям.

Как правило, в случае дисбаланса ($\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{n} b_j$) стремятся сбалансировать модель. Стремление выйти на закрытую транспортную задачу связано с тем, что эффективные методы решения предполагают ее сбалансированность.

2.2.2. Построение сбалансированной транспортной модели

Рассмотрим два случая дисбаланса.

Случай 1. $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j = d$ – спрос превышает предложение.

Модель должна быть построена таким образом, чтобы недостаток продукции оптимально распределялся между потребителями.

Введем фиктивный пункт производства A_{m+1} с объемом производства $a^*_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$.

Продукция, которая будет поставляться из пункта A_{m+1} в пункт B_j , на самом деле будет интерпретироваться как нехватка продукции в пункте B_j.

Определим стоимости перевозки продукции из фиктивного пункта производства A_{m+1} потребителям.

Здесь возможны два подхода. Согласно первому подходу, стоимость перевозки из пункта A_{m+1} в пункт B_j принимается нулевой ($c_{m+1,j}=0$). Действительно, ведь соответствующие перевозки не выполняются.

Однако эту ситуацию можно рассмотреть и с другой стороны.

Поставка из фиктивного пункта – это, фактически, недопоставка. За каждую единицу недопоставки можно определить штраф, установив его равным стоимости $c_{m+1,j}$.

Приведенный ниже рисунок иллюстрирует этот случай.

i \ j	1	2	Запас
1	c_{11} x_{11}	c_{12} x_{12}	a_1
2	c_{21} x_{21}	c_{22} x_{22}	a_2
3	c_{31} x_{31}	c_{32} x_{32}	a_3
	b_1	b_2	

$\Rightarrow$

i \ j	1	2	Запас
1	c_{11} x_{11}	c_{12} x_{12}	a_1
2	c_{21} x_{21}	c_{22} x_{22}	a_2
3	c_{31} x_{31}	c_{32} x_{32}	a_3
$\mathbf{4^*}$	c_{41} x_{41}	c_{42} x_{42}	a^*_{m+1}
	b_1	b_2	

$$a^*_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$$

Случай 2. $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ – предложение превышает спрос.

Вводим фиктивного потребителя B_{n+1}, потребность которого определяется величиной дисбаланса:

$$b_{n+1}^{*} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \ .$$

Здесь, как и в первом случае, поставка фиктивному потребителю из пункта A_i означает, что соответствующий объем продукции просто не вывозится из этого пункта. Т.е. стоимость перевозки единицы продукции из A_i в B_{n+1} можно считать нулевой. Однако можно назначить штраф за хранение единицы не вывезенной продукции – этот штраф и будет равен стоимости $c_{i,n+1}$.

В транспортную таблицу добавляется столбец.

Таким образом, любую модель можно сбалансировать. Поэтому в дальнейшем будем считать, что имеет место сбалансированная модель.

Следующее ограничение – это ограничение на однопродуктовый характер модели. Это ограничение также легко снимается, если несколько видоизменить исходную модель.

2.2.3. Сведение многопродуктовой транспортной задачи к однопродуктовой

Пример 2.3

Имеется 3 атомных электростанции в городах A_1, A_2, A_3. В процессе работы АЭС получается четыре вида радиоактивных отходов – M1, M2, M3, M4. Также существует 2 пункта переработки этих отходов: B1 и B2.

Ниже приведена таблица, в которой указаны объемы выработки отходов различных видов на разных АЭС и мощности пунктов переработки по каждому из отходов (в тех же единицах).

АЭС	Вид отхода				Всего
	M1	M2	M3	M4	
A1	-	-	700	300	1000
A2	500	600	-	400	1500
A3	800	400	-	-	1200
B1	700	500	500	600	2300
B2	600	500	200	100	1400

Для того чтобы учесть многопродуктовый характер модели, изменим задачу следующим образом. Вместо того, чтобы рассматривать каждую АЭС как отдельный пункт, разобьем её на несколько пунктов в соответствии с количеством видов отходов, образующихся на этой АЭС. Аналогично поступим и с пунктами переработки: будем считать, что каждый из этих пунктов состоит из четырех подпунктов – по одному на каждый вид отходов. Таким образом, в терминах транспортной задачи имеем 7 пунктов производства и 8 пунктов потребления.

Теперь нужно учесть то простое обстоятельство, что нельзя осуществить доставку из пункта «производства» отходов одного вида в пункт переработки отходов другого вида. Соответствующие перевозки нужно запретить. Для этого достаточно назначить очень высокие стоимости в соответствующих клетках таблицы.

			B1				B2				
			M1	M2	M3	M4	M1	M2	M3	M4	
		j / i	1	2	3	4	5	6	7	8	Выработка
A1	M3	1	M	M	C_{13}	M	M	M	C_{17}	M	700
	M4	2	M	M	M	C_{24}	M	M	M	C_{28}	300
A2	M1	3	C_{31}	M	M	M	C_{35}	M	M	M	500
	M2	4	M	C_{42}	M	M	M	C_{46}	M	M	600
	M4	5	M	M	M	C_{54}	M	M	M	C_{58}	400
A3	M1	6	C_{61}	M	M	M	C_{65}	M	M	M	800
	M2	7	M	C_{72}	M	M	M	C_{76}	M	M	400
Переработка			700	500	500	600	600	500	200	100	

Итак, уже одни эти примеры говорят о том, что класс задач, описываемых однопродуктовой закрытой транспортной моделью, достаточно широк.

2.2.4. Свойства закрытой транспортной задачи

Закрытая транспортная задача обладает рядом свойств, которые позволяют решать эту задачу гораздо эффективнее, чем обычным симплекс-методом.

1. Задача в любом случае допустима и имеет решение.

Допустимость, например, вытекает из того, что решение $x_{ij} = {}^{a_i b_j}\!/_{d}$ – удовлетворяет всем ограничениям задачи:

$$\sum_{j=1}^{n} \frac{a_i b_j}{d} = a_i \frac{1}{d} \sum_{j=1}^{n} b_j = a_i \frac{d}{d} = a_i \,, \quad \sum_{i=1}^{m} \frac{a_i b_j}{d} = b_j \frac{1}{d} \sum_{i=1}^{m} a_i = b_j \frac{d}{d} = b_j \,.$$

Тот же факт, что задача имеет оптимальное решение, вытекает из ограниченности допустимого множества. Действительно:

$$x_{ij} \le \min\{a_i, b_j\}.$$

2. Ранг матрицы системы ограничений-уравнений r=m+n-1.

Ниже приведена матрица, составленная из коэффициентов левой части системы ограничений.

$$\begin{array}{l}
x_{11}x_{12}\ldots x_{1n}x_{21}x_{22}\ldots\ldots x_{2n}\ldots\ldots\ldots\ldots x_{m1}x_{m2}\ldots x_{mn} \\
1\;\;1\ldots\ldots1\;\;0\;\;0\ldots\ldots0\;\ldots\ldots\ldots\ldots0\;\;0\ldots\;\;0\;\text{-}1 \\
0\;\;0\ldots\ldots0\;\;1\;\;1\ldots\ldots1\;\ldots\ldots\ldots\ldots0\;\;0\ldots\;\;0\;\text{-}2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
0\;\;0\ldots\ldots0\;\;0\;\;0\ldots\ldots0\;\ldots\ldots\ldots\ldots1\;\;1\ldots\;\;1\;\text{-}m \\
\hline
1\;\;0\ldots\ldots0\;\;1\;\;0\ldots\ldots0\;\ldots\ldots\ldots\ldots1\;\;0\ldots\;\;0\;\text{-}1 \\
0\;\;1\ldots\ldots0\;\;0\;\;1\ldots\ldots0\;\ldots\ldots\ldots\ldots0\;\;1\ldots\;\;0\;\text{-}2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
0\;\;0\ldots\ldots1\;\;0\;\;0\ldots\ldots1\;\ldots\ldots\ldots\ldots0\;\;0\ldots\;\;1\;\text{-}n
\end{array}$$

Любая строка этой матрицы линейно выражается через остальные строки, значит $r < m+n$.

Любые $m+n+1$ строки линейно-независимы. Можно показать, что при использовании метода Жордана-Гаусса левая и правая части одного уравнения обатятся в нуль.

3. Каждый вектор-столбец этой матрицы имеет точно две единичные координаты. Остальные координаты – нулевые.

4. Если все запасы и все потребности транспортной задачи выражены целыми числами, то задача имеет оптимальное целочисленное решение.

Транспортная задача – это задача ЛП. Как и в случае симплекс-метода, ее решение всегда начинается с некоторого опорного решения (опорного плана).

Поэтому здесь также будем говорить о базисных и о свободных переменных, об опорных планах (ненулевым координатам которых соответствует система линейно-независимых векторов). Наконец, о вырожденных решениях (некоторые базисные переменные принимают нулевое значение).

Запишем в таблицу условия задачи (рис.2.5).

	1	j	n	Запасы
1	c_{11}	c_{1j}	c_{1n}	a_1
i	c_{i1}	c_{ij}	c_{in}	a_i
m	c_{m1}	c_{mj}	c_{mn}	a_m
Потребности	b_1	b_j	b_n	$d=\Sigma$

Рис.2.5. Транспортная таблица

Сразу следует обратить внимание на компактность этой таблицы по сравнению с обычной симплекс-таблицей. (Здесь $m\times n$ элементов, там – $(m+n)\times m\times n$.)

Клетки этой таблицы будем последовательно заполнять координатами строящегося опорного решения: $\overline{x}_{11}, \overline{x}_{12}, \ldots, \overline{x}_{mn}$.

При этом, координату $\overline{x}_{ij}$ будем записывать в соответствующую клетку только в том случае, если $\overline{x}_{ij}$ – базисная переменная. Такую клетку будем называть <u>*занятой*</u>. Клетки, соответствующие свободным переменным, будем называть <u>*незанятыми*</u> – их заполнять не будем.

Если базисная переменная в решении принимает нулевое значение (вырожденное решение), соответствующую клетку все равно будем считать занятой и записывать в нее «0».

Теперь рассмотрим 4 метода построения допустимого решения транспортной задачи. Эти методы объединяет общая идея.

<u>**Шаг 1**</u>. Поиск пары i и j.
<u>**Шаг 2**</u>. Заполнение клетки $(i\,j)$ значением $min\{a_i,b_j\}$ – максимально возможная нагрузка соответствующего маршрута.
<u>**Шаг 3**</u>. Вычеркивание пункта производства, исчерпавшего свои возможности или пункта потребления, потребности которого удовлетворены полностью.

Шаг 4. Повторение шага 1 (до тех пор, пока все возможности будут исчерпаны, а все потребности – удовлетворены).

2.2.5. Метод северо-западного угла

Шаг 1. Принимаем $i=j=1$.
Шаг 2. Если $b_j > a_i$, то $x_{ij} = a_i$. Корректируем потребность j-го пункта потребления:

$$b_j = b_j - a_i,$$

вычеркиваем i-ю строку (возможность i-го пункта производства исчерпана). Формально $i = i+1$. Переходим к шагу 5.
Шаг 3. Если $b_j < a_i$, то $x_{ij} = b_j$. Корректируем возможность (запас) i-го пункта производства:

$$a_i = a_i - b_j ,$$

вычеркиваем j-й столбец (потребность j-го пункта потребления удовлетворена полностью). Формально $j = j+1$. Переходим к шагу 5.
Шаг 4. Если $b_j = a_i$ (баланс), то принимаем $x_{ij} = b_j = a_i$. Затем вычеркиваем либо строку ($i = i+1$), либо столбец ($j = j+1$). При этом оставляем либо пункт потребления j с нулевой потребностью ($b_j = 0$) , либо пункт производства i с нулевым запасом ($a_i = 0$). Переходим к шагу 5.
Шаг 5. Если $i = m$ (осталась не вычеркнутой одна последняя строка), то:

$$x_{mj}= b_j , x_{mj+1}= b_{j+1},\ldots, x_{mn}= b_n . \quad \text{Конец.}$$

Шаг 6. Если $j = n$ (остался не вычеркнутым один последний столбец), то:

$$x_{in}= a_i , x_{i+1,n}= a_{i+1},\ldots, x_{mn}= a_m . \quad \text{Конец.}$$

В противном случае ($i \neq m$ и $j \neq n$) переходим к шагу 2[17].

[17] В соответствии с приведенным алгоритмом, каждый раз после заполнения очередной клетки из рассмотрения исключается либо один пункт производства, либо один пункт потребления, но не оба сразу. Условие сбалансированности гарантирует сходимость алгоритма.

<u>Пример 2.4</u>

Табл.1

	1	2	3	4		
1	**30**				~~50~~	20
2					40	
3					60	
	30	20	50	50		

Табл.2

	1	2	3	4		
1	30	**20**			**~~20~~**	0
2					40	
3					60	
	30	20	50	50		

Табл.3

	1	2	3	4	
1	30	20	**0**		**0**
2					40
3					60
	30	20	50	50	

Табл.4

	1	2	3	4	
1	30	20	0		0
2			**40**		40
3					60
	30	20	~~50~~	50	
			10		

Табл.5

	1	2	3	4		
1	30	20	0		0	
2			40		40	
3			**10**	**50**	60	⇐Осталась одна строка
	30	20	10	50		

Алгоритм гарантирует заполнение точно $m+n-1$ клеток таблицы. Таким образом формируются значения ($m+n-1$)-ой базисных переменных задачи. Остальные переменные имеют нулевые значения.

2.2.6. Метод наименьшей стоимости

Метод северо-западного угла игнорирует стоимости перевозок. Закрепление потребителя за поставщиком осуществляется по правилу: первый же поставщик пытается удовлетворить потребности первого же потребителя. Между тем, интуитивно ясно, что необходимо максимально загружать маршруты с минимальной стоимостью перевозок.

При таком подходе сгенерированное решение должно быть ближе к оптимальному. То есть для поиска оптимального решения должно потребоваться меньшее количество итераций.

В соответствии с методом наименьшей стоимости на каждом шаге рассматривается та пара «поставщик-потребитель», которая соответствует клетке с минимальной стоимостью перевозки единицы продукции.

Каждый раз после заполнения очередной клетки вычеркивается или один столбец, или одна строка (но ни то и другое одновременно) и соответствующим образом корректируются запасы или потребности.

То есть, по существу – это тот же метод северо-запарного угла, но только с другим правилом закрепления потребителя за поставщиком.

На последнем шаге и того и другого метода остается не вычеркнутой одна строка или один столбец. Это – критерий останова.

Шаг 1. В таблице ищется клетка с минимальной стоимостью. Если таких клеток несколько, выбирается любая из них. Пусть такая клетка находится на пересечении строки i и столбца j.

Шаг 2. Если $b_j > a_i$, то $x_{ij} = a_i$. Корректируем потребность j-го пункта потребления:

$$b_j = b_j - a_i,$$

вычеркиваем i-ю строку. Переходим к шагу 5.

Шаг 3. Если $b_j < a_i$, то $x_{ij} = b_j$. Корректируем возможность (запас) i-го пункта производства:

$$a_i = a_i - b_j ,$$

вычеркиваем j-й столбец. Переходим к шагу 5.

Шаг 4. Если $b_j = a_i$ (баланс), то принимаем $x_{ij} = b_j = a_i$. Затем вычеркиваем либо строку, либо столбец. При этом оставляем либо пункт потребления j с нулевой потребностью ($b_j = 0$), либо пункт производства i с нулевым запасом ($a_i = 0$). Переходим к шагу 5.

Шаг 5. Если осталась не вычеркнутой одна строка или один столбец, то заполняем все клетки этой строки (столбца). Конец. В противном случае переходим к шагу 1.

Пример 2.5

Договоримся, что при прочих равных будем вычеркивать столбец.

Табл.1

	1	2	3	4		
1	10	0 **15**	20	11	~~15~~	0
2	12	7	9	20	25	
3	0	14	16	18	5	
	5	15	15	10		

Табл.2

	1	2	3	4		
1	10	0 15	20	11	**0**	
2	12	7	9	20	25	
3	0 **5**	14	16	18	~~5~~	0
	5	15	15	10		

Табл.3

	1	2	3	4		
1	10	0 15	20	11	0	
2	12	7	9 **15**	20	~~25~~	10
3	0 5	14	16	18	**0**	
	5	15	15	10		

Табл.4

	1	2	3	4	
1	10	0 15	20	11 **0**	0
2	12	7	9 15	20 **10**	**10**
3	0 5	14	16	18 **0**	0
	5	15	15	10	

Значение ЦФ: Z=335.

2.2.7. Метод Фогеля

Как правило, этот метод дает лучший результат, чем МНС. Метод дает оптимальное, либо близкое к оптимальному решение.

Шаг 1. Вычислить штраф для каждого столбца и для каждой строки. Для этого в каждой строке (столбце) отыскивается элемент с минимальной стоимостью и ближайший к нему по стоимости элемент. Разность этих стоимостей и есть штраф.

	1	2	3	4		⇓ Штраф за невывоз
1	10	0	20	11	15	10
2	12	7	9	20	25	2
3	0	14	16	18	5	**14**
	5	15	15	10		
	10	7	7	7		⇐Штраф за недопоставку

<u>Шаг 2</u>. Отметить строку или столбец с самым большим штрафом. Если таких несколько, выбрать среди них любую строку или любой столбец.

<u>Шаг 3</u>. В отмеченной строке (или столбце) выбрать переменную с самой низкой стоимостью и придать ей максимально возможное значение. Скорректировать объем производства и спрос и вычеркнуть строку или столбец, соответствующий выполненному ограничению.

Если ограничение выполняется одновременно по строке и столбцу, то вычеркнуть либо строку, либо столбец. Оставшемуся столбцу придать нулевой спрос (нулевой объем производства). Строка (или столбец) с нулевым объемом производства (или спроса) хотя и не вычеркивается, но в дальнейших вычислениях не участвует.

<u>Шаг 4</u>. Если остается не вычеркнутой только одна строка (столбец) с положительным объемом производства (или один столбец с положительным объемом спроса), базисные переменные вычисляются методом наименьшей стоимости. При этом в задачу включаются не вычеркнутые строки (с нулевыми объемами производства) и не вычеркнутые столбцы (с нулевыми объемами спроса). Конец.

В противном случае выполняется шаг 1.

Пример 2.6

Табл. 1

	1	2	3	4		
1	10	0	20	11	15	10
2	12	7	9	20	25	2
3	0 **5**	14	16	18	~~5~~ 0	**14**
	5	15	15	10		
	10	7	7	7		

Табл. 2

	1	2	3	4		
1	10	0 **15**	20	11	~~15~~ 0	**11**
2	12	7	9	20	25	2
3	0 **5**	14	16	18	0	Не участв**.**
	5	15	15	10		
		7	11	9		

Осталась не вычеркнутой одна строка. Применяем метод наименьшей стоимости.

Табл. 3

	1	2	3	4		
1	10	0 **15**	20	11	0	Не участв**.**
2	12	7	9 **15**	20	~~25~~ 10	
3	0 **5**	14	16	18	0	Не участв**.**
	5	15	15	10		

Табл. 4

	1	2	3	4	
1	10	0 **15**	20	11 **0**	0
2	12	7	9 **15**	20 **10**	10
3	0 **5**	14	16	18 **0**	0
	5	15	15	10	

Значение Z=335.

2.2.8. Метод плавающих зон

Метод используется для приближенного решения транспортной задачи повышенной размерности. Полученный план близок к оптимальному.

МНС достаточно прямолинейно учитывает стоимости перевозок: на каждом шаге выбирается самый дешевый маршрут, причем, последствия принятого решения игнорируются.

Метод Фогеля учитывает последствия выбранной ячейки на шаг вперед. Использование механизма штрафов – это, по существу, попытка оценить важность маршрута с позиции того, каковы будут затраты, связанные с последующей загрузкой соседнего, ближайшего по стоимости маршрута.

Метод плавающих зон при принятии решений о загрузке маршрутов учитывает не только стоимости перевозок, но и объемы продукции как выпускаемые в пунктах производства, так и потребляемые в пунктах потребления. Идея метода сводится к следующему.

Независимо от содержательно постановки задачи, стоимость перевозки единицы продукции из пункта A_i в пункт B_j рассматривается, как расстояние между этими пунктами.

В этой связи, всю территорию, которая обслуживается пунктами производства, можно условно разбить на зоны, в которые входят пункты потребления по принципу минимального расстояния до соответствующих пунктов производства (рис. 2.6). Т.е. количество зон – это количество пунктов производства.

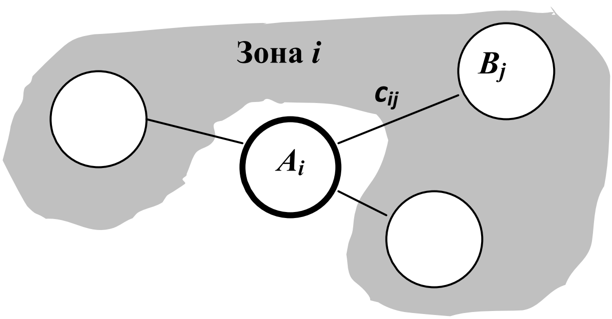


Рис.2.6 Закрепление пунктов потребления за зонами

Пункт потребления B_j ($j = \overline{1,n}$) попадает в зону A_i или, просто, в i-ю зону ($i = \overline{1,m}$), если $c_{ij} = \min_r \{c_{rj}\}$.

Обозначим через Q_i подмножество номеров пунктов потребления, относящихся к в i-й зоне ($i = \overline{1,m}$). При этом будем считать, что:

$$Q_i \cap Q_k = \varnothing \ \ при \ \ i \neq k \ \ и \ \ \bigcup_i Q_i = \{1, 2, ..., n\}.$$

Суммарный объем потребления пунктов, находящихся в i-й зоне, составляет:

$$b^{(i)} = \sum_{j \in Q_i} b_j.$$

Очевидно, что спрос объектов i-й зоны ($i = \overline{1,m}$) может быть удовлетворен i-м пунктом производства только в том случае, если $b^{(i)} \leq a_i$.

В противном случае i-я зона считается *<u>перегруженной</u>*: $b^{(i)} > a_i$.

Что делать, если i-я зона перегружена? Прежде всего, необходимо решить, какие объекты следует закрепить за i-м пунктом производства.

В методе плавающих зон этот вопрос решается с использованием специального механизма *<u>приоритетов</u>*, которые вычисляются точно так же, как штрафы столбцов в методе Фогеля:

$$P_j = c_{i',j} - c_{i,j}, \ (j \in Q_i).$$

Здесь $c_{i',j} = \min_{r \neq i} \{c_{r,j}\}$ – транспортные издержки, связанные с перевозкой единицы продукции в j-й пункт потребления из ближайшего пункта производства (из другой, ближайшей зоны).

Закрепление объектов за i-й зоной осуществляется в порядке не возрастания приоритетов – до тех пор, пока не будет исчерпан весь объем производства этой зоны.

Оставшиеся неудовлетворенными объекты i-й зоны вновь подключаются к массиву нерасмотренных объектов.

Если спрос некоторого объекта удовлетворен частично, этот объект также подключается к массиву нерассмотренных объектов с объемом потребления, равным нехватке.

После того, как объем производства перегруженной зоны полностью исчерпывается, соответствующий пункт производства исключается из рассмотрения – просто вычеркивается соответствующая строка таблицы.

Процедура закрепления объектов выполняется только для перегруженных зон. Если таких зон нет, алгоритм не работает – нужно переключиться, например, на МНС.

Итак, алгоритм сводится к поиску перегруженных зон и закреплению за ними объектов. Затем такие зоны исключаются, и решается, по существу, новая задача – для оставшихся пунктов производства и пунктов потребления, потребности которых не удовлетворены.

Шаг 1. Распределение объектов по зонам.

Шаг 2. Если зона одна, то эта зона не может быть перегружена, так как имеет место закрытая транспортная задача. Объем производства этой зоны распределяется в соответствии с потребностями. Конец. Иначе - поиск перегруженной зоны. Если такой зоны нет, конец: план нужно искать другим методом, например, МНС.

Шаг 3. Вычисление приоритетов для объектов перегруженной зоны.

Шаг 4. Закрепление объектов за пунктом производства перегруженной зоны в порядке невозрастания приоритетов до исчерпания всего объема производства. В процессе закрепления объектов выполняются следующие действия:

1. Если потребность очередного объекта удовлетворена, а запас зоны не израсходован, объект вычеркивается. Переход к следующему (в порядке невозрастания приоритетов) объекту.
2. Если потребность очередного объекта удовлетворена частично, а запас исчерпан, зона (пункт производства) вычеркивается. Очередной объект остается с объемом потребления, равным нехватке. Выполняется **шаг 1**.
3. Если потребность очередного объекта полностью удовлетворена, а объем производства зоны исчерпан, зона вычеркивается, а объект остается с нулевой потребностью. Выполняется **шаг 1**.

Пример 2.7

Табл. 1

	1	2	3	4	
1	10	0 **15**	20	11	15
2	12	7	9	20	25
3	0	14	16	18	5
b_i	5	~~15~~ 0	15	10	
	3	1	2	1	⇐зоны
		7		**7**	⇐приоритеты

1. Разбили объекты по зонам.
2. Начинаем с первой зоны (в нее вошли объекты 2,4): $b_2+b_4=15+10=25 > a_1=15$. Зона перегружена.
3. Вычисляем приоритеты объектов 2 и 4: P_2=7-0=7; P_4=18-11= =7, т.е. в зону 1 можно включить как второй, так и четвертый объект. Включаем второй. Потребность=15, запас=15.
4. Принимаем $x_{12}=15$. Исключаем зону 1, а второй объект оставляем с нулевой потребностью.

Табл. 2

	1	2	3	4	
1	10	0 **15**	20	11	15
2	12	7	9	20	25
3	0 **5**	14	16	18	5
b_i	~~5~~ 0	0	15	10	
	3	2	2	3	⇐зоны
	12			2	⇐приоритеты

Разбили объекты по зонам.

Начинаем со второй зоны (в нее вошли объекты 2,3): $b_2+b_3=0+15=15 < a_2=25$. Зона не перегружена. Рассматриваем третью зону (объекты 1,4): $b_1+b_4=5+10=15 > a_3=5$. Зона перегружена.

Вычисляем приоритеты объектов 1 и 4: P_1=12-0=12; P_4=20-18= 2. В зону 3 включаем первый объект. Потребность=5, запас=5.

Принимаем $\boldsymbol{x_{31}=5}$. Исключаем зону 3, а первый объект оставляем с нулевой потребностью.

Табл. 3

	1	2	3	4	
1	10	0 **15**	20	11	15
2	12 **0**	7 **0**	9 **15**	20 **10**	25
3	0 **5**	14	16	18	5
b_i	0	0	15	10	

Осталась одна зона. Т.е. распределять объекты по зонам не нужно. Распределяем объем производства 25 ед. между 1, 2, 3, 4 объектами в соответствии с потребностью этих объектов.

Значение Z=335. Как и в методе Фогеля.

2.2.9. Использование второй теоремы двойственности для обоснования метода потенциалов

Метод потенциалов является точным методом решения транспортной задачи. Своим обоснованием этот алгоритм обязан теории двойственности в линейном программировании.

В разделе 1.3 были рассмотрены математические модели взаимно-двойственных задач. В частности, рассматривалась пара задач:

$$\begin{array}{c} < c, x > \to \min, \\ AX = A_0, \\ x \geq 0. \end{array} \quad \Rightarrow \quad \begin{array}{c} < A_0^T, y > \to \max, \\ A^T Y \leq C. \end{array}$$

Запишем условия второй теоремы двойственности (теорема 1.7) для этой пары.

Пусть $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ – некоторое допустимое решение прямой задачи, а $\overline{y} = (\overline{y}_1, \overline{y}_2, ..., \overline{y}_m)$ – допустимое решение двойственной задачи. Необходимым и достаточным условием того, что эти решения являются оптимальными, является выполнение всех приведенных ниже равенств:

$$(a_{1j}\overline{y}_1 + a_{2j}\overline{y}_2 + ... + a_{mj}\overline{y}_m - c_j)\overline{x}_j = 0 \text{ для всех } j = \overline{1, n}.$$

Нас будет интересовать только одна сторона этого утверждения, а именно.

Если нестрогие неравенства двойственной задачи, соответствующие базисным переменным опорного решения прямой задачи при подстановке координат решения двойственной задачи обратятся в строгие равенства, то эти решения – оптимальные.

На этом, собственно, и построен метод потенциалов.

Запишем транспортную задачу в развернутом виде.

$$c_{11}x_{11}+\ldots+c_{1n}x_{1n} \quad + \quad c_{21}x_{21}+\ldots+c_{2n}x_{2n} \quad +\ldots+ \quad c_{m1}x_{m1}+\ldots+c_{mn}x_{mn} \quad \to \min$$

$$x_{11}+\ldots\ldots+x_{1n} \qquad\qquad =a_1\sim u_1$$

$$x_{21}+\ldots\ldots+x_{1n} \qquad\qquad =a_2\sim u_2$$

……………………………………………………………………………

$$x_{m1}+\ldots\ldots+x_{mn} \qquad =a_m\sim u_m$$

$$x_{11} \qquad +x_{21} \qquad +x_{m1} \qquad =b_1\sim v_1$$

$$x_{12} \qquad +x_{22} \qquad +x_{m2} \qquad =b_2\sim v_2$$

……………………………………………………………………………

$$x_{1n} \qquad +x_{2n} \qquad +x_{mn} \qquad =b_n\sim v_n$$

$$x_{ij}\geq 0 \quad (i=\overline{1,m},\ j=\overline{1,n}).$$

Каждому из ограничений первой группы (i-му ограничению) поставим в соответствие переменную u_i ($i=\overline{1,m}$). Назовем ее *<u>потенциалом строки</u>*.

Каждому из ограничений второй группы (j-му ограничению) поставим в соответствие переменную v_j ($j=\overline{1,n}$). Назовем ее *<u>потенциалом столбца.</u>*

Эти же обозначения используем при построении двойственной задачи.

$$\sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \to \max$$
$$u_i + v_j \leq c_{ij}\,,\ (i=\overline{1,m},\ j=\overline{1,n}) \qquad (2.17)$$

Предположим, что $\overline{x}=(\overline{x}_{11},\overline{x}_{12},...,\overline{x}_{mn})$ – опорный план прямой задачи, а $\overline{y}=(\overline{u}_1,\overline{u}_2,...,\overline{u}_m,\overline{v}_1,\overline{v}_2,...,\overline{v}_n)$ – некоторый план двойственной задачи.

Из второй теоремы двойственности следует, что если для всех базисных переменных решения $\overline{x}=(\overline{x}_{11},\overline{x}_{12},...,\overline{x}_{mn})$ имеет место:

$$\overline{u}_i + \overline{v}_j = c_{ij}, \qquad (2.18)$$

а для свободных переменных этого решения выполняется:

$$\overline{u}_i + \overline{v}_j \le c_{ij}, \tag{2.19}$$

то $\overline{x} = (\overline{x}_{11}, \overline{x}_{12}, ..., \overline{x}_{mn})$ и $\overline{y} = (\overline{u}_1, \overline{u}_2, ..., \overline{u}_m, \overline{v}_1, \overline{v}_2, ..., \overline{v}_n)$ – оптимальные решения соответствующих задач.

Теперь представим себе, что мы имеем некоторый опорный план прямой задачи $\overline{x} = (\overline{x}_{11}, \overline{x}_{12}, ..., \overline{x}_{mn})$.

Основная идея метода потенциалов заключается в попытке искусственно сконструировать план двойственной задачи и проверить, выполняются ли условия второй теоремы двойственности.

Конструирование плана осуществляется следующим образом. Известно, какие переменные в решении $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ являются базисными (заполненные клетки таблицы). Для заполненных клеток должны выполняться соотношения (2.18): $\overline{u}_i + \overline{v}_j = c_{ij}$.

Допустим, что эти соотношения выполняются. Попытаемся найти компоненты решения $\overline{y} = (\overline{u}_1, \overline{u}_2, ..., \overline{u}_m, \overline{v}_1, \overline{v}_2, ..., \overline{v}_n)$. Что любопытно: всего уравнений *m+n-1* (количество заполненных клеток), а количество неизвестных – *m+n*. Следовательно, чтобы решить систему уравнений, нужно зафиксировать одну из неизвестных, придав ей конкретное значение (например, положив $\overline{u}_1 = 0$). Остальные определятся однозначно.

Теперь имеем компоненты решения (пока это – просто вектор) $\overline{y} = (\overline{u}_1, \overline{u}_2, ..., \overline{u}_m, \overline{v}_1, \overline{v}_2, ..., \overline{v}_n)$, причем условия (2.18) выполняются автоматически.

Проверим выполнение условий (2.19). Для каждой свободной переменной (не заполненной клетки) должно выполняться

$$\overline{u}_i + \overline{v}_j \le c_{ij}.$$

Эта проверка в методе потенциалов сводится к вычислению <u>*косвенных стоимостей*</u>:

$$E_{ij} = c_{ij} - (\overline{u}_i + \overline{v}_j),$$

и к анализу знаков полученных величин.

Теперь, если оказалось, что все $E_{ij} \ge 0$ (или, что эквивалентно, $\overline{u}_i + \overline{v}_j \le c_{ij}$), то полностью выполнены условия второй теоремы, и решение $\overline{x} = (\overline{x}_{11}, \overline{x}_{12}, ..., \overline{x}_{mn})$ – оптимальное.

В противном случае (существует $E_{ij} < 0$) решение $\overline{x}$ не является оптимальным, а построенное нами решение $\overline{y}$ – вообще не допустимое.

Итак, способ проверки любого опорного решения транспортной задачи на оптимальность есть.

Пример 2.8

Проверим на оптимальность решение, полученное методом Фогеля.

Табл. 4

	v_1	v_2	v_3	v_4
u_1	10	0 **15**	20	11 **0**
u_2	12	7	9 **15**	20 **10**
u_3	0 **5**	14	16	18 **0**

По заполненным клеткам составим и решим систему уравнений:

$u_1+v_2=0$ $u_1=0$ $v_2=0$
$u_1+v_4=11$ $u_1=0$ $v_4=11$
$u_2+v_3=9$ $u_2=9$ $v_3=0$
$u_2+v_4=20$ $v_4=11$ $u_2=9$
$u_3+v_1=0$ $u_3=7$ $v_1=-7$
$u_3+v_4=18$ $v_4=11$ $u_3=7$.

После этого по свободным клеткам рассчитаем косвенные стоимости.

	-7	*0*	*0*	*11*
0	10	0 **15**	20	11 **0**
9	12	7	9 **15**	20 **10**
7	0 **5**	14	16	18 **0**

E_{11}=10-(0-7)=17, E_{13}=20-(0+0)=20, E_{21}=12-(9-7)=10
E_{22}=7-(9+0)= -2!, E_{32}=14-(7+0)=7, E_{33}=16-(7+0)=9.

Решение не оптимальное, так как E_{22}= -2 < 0.

Рассмотрим теперь вопрос перехода к новому, лучшему опорному решению в методе потенциалов.

2.2.10. Переход к новому опорному решению в методе потенциалов

Пусть при проверке очередного опорного решения обнаружилось, что $\exists\, E_{ij} < 0$, т.е. рассматриваемое решение не оптимально.

Для косвенной стоимости справедливо следующее утверждение:

косвенная стоимость E_{ij} *– это коэффициент, с которым свободная переменная* x_{ij} *входит в ЦФ, если все базисные переменные выразить через свободные*[18].

В этой связи становится ясно, какую переменную нужно вводить в состав базисных, чтобы получить новое, лучшее опорное решение: переменную с отрицательной косвенной стоимостью (желательно, с максимальной по абсолютной величине отрицательной косвенной стоимостью).

Ответим теперь на вопрос, а какую переменную следует выводить из состава базисных?

Пусть $E_{\mu\nu} < 0$. Вводим переменную $x_{\mu\nu}$ в состав базисных переменных. Придадим этой переменной некоторое положительное приращение $\rho > 0$. То есть потребуем, чтобы из пункта μ в пункт ν было отправлено продукции в количестве ρ единиц. Назовем клетку с координатами $(\mu\ ,\ \nu)$ *положительной вершиной*. Теперь рассмотрим рис. 2.7.

[18] Вспомним определение оценки свободного вектора в симплекс-методе.

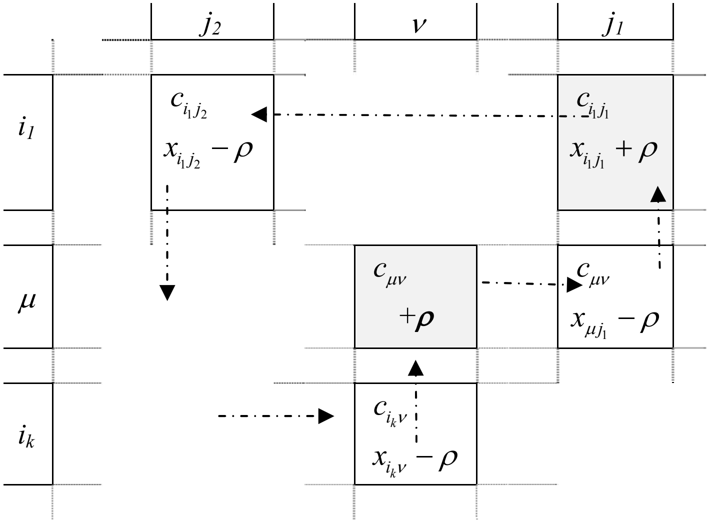


Рис.2.7. Приращение значения в свободной клетке

Отправить в ν-й пункт ρ единиц продукции можно только уменьшив поставку из пункта μ в в другие пункты на ρ единиц.

Найдем заполненную клетку (базисную переменную) в той же строке. Пусть это будет $x_{\mu j_1}$. Уменьшим поставку в пункт j_1 на ρ единиц. Теперь поставка в этот пункт составляет ($x_{\mu j_1} - \rho$) единиц. Соответствующую вершину назовем *отрицательной*.

Недопоставка в пункт j_1 на ρ единиц компенсируется за счет другого пункта – это занятая клетка того же столбца. Пусть это будет $x_{i_1 j_1}$. Теперь поставка в этот пункт составляет ($x_{i_1 j_1} + \rho$) единиц. Соответствующую вершину назовем *положительной*.

Но это можно сделать только за счет недопоставки ρ единиц в другой пункт – на рис.2.7 этим пунктом является пункт j_2. Уменьшим поставку в пункт j_2 на ρ единиц. Теперь поставка в этот пункт составляет ($x_{i_1 j_2} - \rho$) единиц. Соответствующую вершину назовем *отрицательной*. В итоге образуется последовательность вершин $x_{\mu\nu}, x_{\mu j_1}, x_{i_1 j_1}, \ldots$, продолжающаяся до тех пор, пока очередной положительной вершиной не станет исходная незанятая клетка, соответствующая переменной $x_{\mu\nu}$.

Образовался <u>*цикл пересчета*</u>, все вершины которого, кроме одной, соответствуют базисным переменным. В разделе 2.2.11 будет показано, что такой цикл – единственный для конкретной свободной клетка в данной транспортной таблице.

Переход к новому опорному решению в методе потенциалов производится следующим образом.

Начнем увеличивать ρ. Исходное значение переменной $x_{\mu\nu}$ – это нуль (свободная переменная). Соответственно, с увеличением ρ значение этой переменной будет увеличиваться. Будут увеличиваться также значения базисных переменных, соответствующих положительным вершинам построенного цикла пересчета. Значения базисных переменных, не входящих в цикл пересчета, будут оставаться неизменными.

Что же касается переменных, связанных с отрицательными вершинами, их значения начнут уменьшаться. Ввиду того, что новое решение должно быть опорным и, естественно, допустимым, ρ можно увеличивать только до тех пор, пока значение одной или нескольких переменных, связанных с отрицательными вершинами цикла пересчета, не обратится в нуль. Соответствующую переменную, очевидно, нужно выводить из состава базисных переменных. При этом, если в нуль обращаются несколько переменных, из базиса нужно выводить только одну из них. Остальные же такие переменные должны остаться в составе базисных с нулевыми значениями: новое опорное решение в этом случае является вырожденным опорным решением.

Итак, предельное значение ρ – это минимальный объем перевозок, найденный в отрицательных вершинах цикла.

Пусть $\overline{x}_{ks}$ – минимальный объем перевозок, найденный в одной из отрицательных вершин цикла пересчета. Эта величина и будет представлять собой искомое предельное значение ρ – значение переменной $x_{\mu\nu}$, которая в новом опорном решении вводится в состав базисных переменных. Значения базисных переменных, соответствующих вершинам цикла, корректируются – увеличиваются или уменьшаются на величину $\rho = \overline{x}_{ks}$. При этом переменная x_{ks}, принявшая нулевое значение, выводится из состава базисных переменных нового опорного решения – соответствующая клетка таблицы объявляется незанятой.

Если кроме x_{ks} после пересчета некоторые другие базисные переменные обращаются в нуль, в соответствующие клетки записывается нуль, и они объявляются «условно-занятыми».

Увеличение на величину $\rho = \overline{x}_{ks}$ переменных, соответствующих положительным вершинам цикла пересчета, и уменьшение на эту же величину переменных, соответствующих отрицательным вершинам, называется <u>*сдвигом по циклу пересчета*</u>.

<u>Пример 2.9</u>

Дана транспортная таблица, в которой записано некоторое опорное решение.

Табл. 1

		3	*2*	*2*	*1*
		v_1	v_2	v_3	v_4
0	u_1	3 **5**	2 **25**	4	1 **20**
-1	u_2	2 **25**	3	1 **15**	5
2	u_3	3	2	4 **20**	4

Здесь $6=m+n-1=3+4-1$ заполненных клеток, соответствующих базисным переменным.

По заполненным клеткам составим систему уравнений для нахождения потенциалов ($u_i + v_j = c_{ij}$):

$$u_1+ v_1=3 \quad u_1=0 \quad v_1=3$$
$$u_1+ v_2=2 \quad u_1=0 \quad v_2=2$$
$$u_1+ v_4=1 \quad u_1=0 \quad v_4=1$$
$$u_2+ v_1=2 \quad v_1=3 \quad u_2=-1$$
$$u_2+ v_3=1 \quad u_2=-1 \quad v_3=2$$
$$u_3+ v_3=4 \quad v_3=3 \quad u_3=2.$$

Для удобства последующих вычислений занесем полученные результаты в новую таблицу.

Табл. 2

	3	*2*	*2*	*1*
0	3 5+ρ	2 **25-ρ**	4	1 **20**
-1	2 **25-ρ**	3	1 **15+ρ**	5
2	3	2 +ρ	4 **20-ρ**	4

Теперь по каждой незаполненной клетке посчитаем соответствующую косвенную стоимость $E_{ij} = c_{ij} - (u_i + v_j)$:

E_{13}=4-(2+0)=2, E_{22}=3-(-1+2)=2, E_{24}=5-(-1+1)=5
E_{31}=3-(2+3)= -2, E_{32}=2-(2+2)=-2, E_{34}=4-(2+1)=1.

Наличие отрицательных косвенных стоимостей говорит о неоптимальности записанного в таблицу решения. Это решение можно улучшить, введя в состав базисных переменных переменную $\boldsymbol{x_{31}}$ или $\boldsymbol{x_{32}}$. Введем $\boldsymbol{x_{32}}$.

Для этого построим цикл пересчета.

Для определения величины ρ находим среди отрицательных вершин такую, которой соответствует минимальное значение переменной в опорном решении. Это $\boldsymbol{x_{33}=20}$. Следовательно, принимаем $\boldsymbol{\rho=20}$ и выполняем сдвиг по циклу пересчета. После этого сдвига переменная $\boldsymbol{x_{33}}$ принимает нулевое значение. Эту переменную выведем из состава базисных переменных (введем же переменную $\boldsymbol{x_{32}}$).

В табл. 3 записано новое опорное решение.

Табл. 3

	3	*2*	*2*	*1*
0	3 25	2 **5**	4	1 **20**
-1	2 **5**	3	1 **35**	5
0	3	2 **20**	4	4

По заполненным клеткам составим систему уравнений для нахождения потенциалов ($u_i + v_j = c_{ij}$):

$$
\begin{array}{lll}
u_1+v_1=3 & u_1=0 & v_1=3 \\
u_1+v_2=2 & u_1=0 & v_2=2 \\
u_1+v_4=1 & u_1=0 & v_4=1 \\
u_2+v_1=2 & v_1=3 & u_2=-1 \\
u_2+v_3=1 & u_2=-1 & v_3=2 \\
u_3+v_2=2 & v_2=2 & u_3=0.
\end{array}
$$

По каждой незаполненной клетке посчитаем стоимость:

E_{13}=4-(2+0)=2, E_{22}=3-(-1+2)=2, E_{24}=5-(-1+1)=5

E_{31}=3-(0+3)= 0, E_{33}=4-(0+2)= 2, E_{34}=4-(0+1)=3.

Все косвенные стоимости неотрицательные, следовательно, решение, записанное в этой таблице – оптимальное.

$Z_{\text{опт}}$=190. На исходном решении ЦФ имеет значение 230.

Пример 2.10

Мы решали методом Фогеля следующую задачу (Пример 2.6):

	1	2	3	4	
1	10	0	20	11	15
2	12	7	9	20	25
3	0	14	16	18	5
	5	15	15	10	

Получили ее решение:

Потенциалы строк u_i \ Потенциалы столбцов v_j	-9	0	0	11
0	10	0 **15**	20	11 **0**
9	12	$\boxed{7}$	9 **15**	20 **10**
9	0 **5**	14	16	18 **0**

И показали, что оно не оптимальное, так как E_{22}= -2 < 0.

Продолжим решение. Построим цикл пересчета.

Определяем ρ = 10 и производим сдвиг по циклу пересчета. Переменная x_{24} выводится из состава базисных переменных, а переменная x_{22} вводится в состав базисных переменных.

	-9	*0*	*2*	*11*
0	10	0 **15-ρ**	20	11 **0+ρ**
7	12	7 +ρ	9 **15**	20 **10-ρ**
9	0 **5**	14	16	18 **0**

	-9	*0*	*2*	*11*
0	10	0 **5**	20	11 **10**
7	12	7 **10**	9 **15**	20
9	0 **5**	14	16	18 **0**

По заполненным клеткам составим и решим систему уравнений:

$u_1+v_2=0$ $\qquad\qquad$ $u_1=0$ $\quad$ $v_2=0$

$u_1+v_4=11$ $\quad$ $u_1=0$ $\quad$ $v_4=11$

$u_2+v_2=7$ $\quad$ $v_2=0$ $\quad$ $u_2=7$

$u_2+v_3=9$ $\quad$ $u_2=7$ $\quad$ $v_3=2$

$u_3+v_1=0$ $\quad$ $u_3=9$ $\quad$ $v_1=-9$

$u_3+v_4=18$ $\quad$ $v_4=11$ $\quad$ $u_3=9.$

Теперь по свободным клеткам рассчитаем косвенные стоимости:

E_{11}=10-(0-9)=19, $\quad$ E_{13}=20-(0+2)=18, $\quad$ E_{21}=12-(7-9)=14

E_{24}=20-(7+11)= 2, $\quad$ E_{32}=14-(9+0)=5, $\quad$ E_{33}=16-(9+2)=5.

Нет ни одной отрицательной косвенной стоимости, значит записанное в таблице решение – оптимальное.

На этом решении ЦФ принимает значение: $Z_{опт}$=315 – меньшее (т.е. лучшее), чем решение, полученное ранее приближенными методами (там $Z_{опт}$=335).

2.2.11. Выводы по методу потенциалов

Метод потенциалов дает нам эффективное средство подсчета косвенных стоимостей незанятых клеток таблицы – свободных переменных. Как уже отмечалось, косвенная стоимость $E_{ij} = c_{ij} - (u_i + v_j)$ – это коэффициент, с которым свободная переменная x_{ij} входит в ЦФ, если все базисные переменные выразить через свободные. В ЛП – это оценка соответствующего свободного вектора (правда, взятая с обратным знаком).

Для вычисления оценок в симплекс-методе необходимо знать коэффициенты разложения соответствующих векторов или, как в МСМ, обратную базисную матрицу текущего опорного решения.

Таким образом, первой особенностью транспортной задачи является весьма экономная процедура вычисления оценок – ни о какой обратной матрице речи не идет.

После принятия решения о том, какую переменную нужно вводить в состав базисных переменных, вопрос о том, какую переменную выводить из состава базисных переменных, решается весьма своеобразно – с использованием цикла пересчета. Здесь остаются неясности, часть из которых необходимо снять. Прежде всего, нужно строго определить понятие цикла пересчета.

Циклом в матрице назовем ломаную, вершины которой расположены в клетках, а звенья лежат вдоль строк и столбцов матрицы (рис. 2.8). При этом ломаная должна удовлетворять следующим условиям:

1. Ломаная должна быть *связной* в том смысле, что из одной ее вершины можно попасть в любую другую вершину по звеньям ломаной;
2. В каждой вершине ломаной встречаются ровно два звена, причем одно из них располагается по строке, а другое – по столбцу[19].

В случае самопересечения точка пересечения не является вершиной.

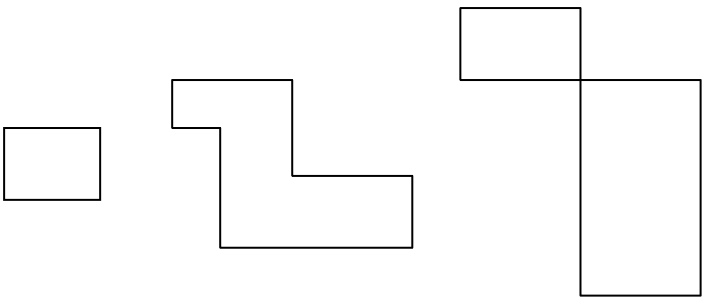


Рис.2.8. Примеры циклов в матрице

[19] Если назвать две вершины, являющиеся концами одного звена, «соседними», то из условия 2) вытекает, что каждая вершина цикла обладает ровно двумя соседними: одна из них располагается по строке, другая – по столбцу.

Циклом пересчета данной свободной клетки назовем цикл, одна из вершин которого находится в этой свободной клетке, а все остальные – в занятых.

ТЕОРЕМА 2.1

Пусть в матрице перевозок записано некоторое опорное решение. Каков бы ни был состав свободных и базисных переменных любого опорного решения транспортной задачи, в матрице перевозок не существует цикла, все вершины которого находятся в базисных клетках.

Доказательство

Запишем систему ограничений транспортной задачи в следующей форме:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}x_{ij} = A_0 \ , \tag{2.20}$$

где
$$A_{ij} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ i \\ \vdots \\ m \\ 1 \\ \vdots \\ n \\ \vdots \\ n \end{matrix} , \qquad A_0 = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_j \\ \vdots \\ b_n \end{pmatrix}.$$

Пусть $\overline{x} = (\overline{x}_{11},...,\overline{x}_{ij},...,\overline{x}_{mn})$ – опорное решение транспортной задачи.

Введем два подмножества σ и ω такие, что:

$(i, j)\in\sigma$, если $\overline{x}_{ij}$ – базисная переменная;

$(i, j)\in\omega$, если $\overline{x}_{ij}$ – свободная переменная.

Тогда, очевидно, будет иметь место:

$$\sum_{(i,j)\in\sigma} A_{ij}\overline{x}_{ij} = A_0 . \tag{2.21}$$

Пусть, от противного, существует цикл, все вершины которого находятся в заполненных клетках. При этом $\sigma' \subseteq \sigma$ – подмножество базисных переменных, входящих в цикл пересчета.

Придадим любой переменной $\overline{x}_{ij}$ $(i, j) \in \sigma'$ некоторое приращение $\rho > 0$ и произведем сдвиг по циклу пересчета. Получим новое решение системы (2.20) $\overline{\overline{x}} = (\overline{\overline{x}}_{11}, ..., \overline{\overline{x}}_{ij}, ..., \overline{\overline{x}}_{mn})$, причем:

$$\sum_{(i,j)\in\sigma} A_{ij}\overline{\overline{x}}_{ij} = A_0. \tag{2.22}$$

Ввиду того, что $\overline{x} \neq \overline{\overline{x}}$, получено новое разложение вектора A_0 по тому же базису. А этого не может быть, так как разложение любого вектора по базису единственно. Следовательно, предположение о том, что существует цикл, все вершины которого находятся в заполненных клетках, оказалось неверным. Теорема доказана.

Следствие. *Для каждой свободной клетки в транспортной таблице существует единственный цикл пересчета*

Предположим, что для некоторой свободной клетки существуют два цикла пересчета. На рис. 2.10 показано, что в этом случае при отбрасывании свободной клетки образуется цикл пересчета, все вершины которого находятся в базисных клетках, что противоречит теореме 2.1.

Рис.2.10. Иллюстрация единственности цикла пересчета

Казалось бы, приведенный пример (рис. 2.9) опровергает утверждение теоремы. Здесь цикл включает только заполненные клетки. Однако решение, записанное в таблице, хотя и допустимое, но не опорное.

	20	→	**20**	40
50	↑	**10**	↓	60
	20	←	**20**	40
50	40	10	40	140

Рис.2.9. Ложный пример, якобы опровергающий теорему 2.1

Цикл пересчета может не включать все базисные переменные (занятые клетки). Те из базисных переменных, которые входят в цикл в качестве положительных вершин, при переходе к новому опорному решению увеличивают свое значение на одну и ту же величину. Базисные переменные, соответствующие отрицательным вершинам, при переходе к новому опорному решению уменьшают свое значение на одну и ту же величину. Базисные переменные, не входящие в цикл пересчета, при переходе к новому опорному решению сохраняют свое прежнее значение.

2.2.12. Распределительный метод решения транспортной задачи

Пусть имеется некоторое опорное решение и $\overline{x}_{ij}$ – свободная переменная, а $\overline{x}_{kl}$ - базисная переменная.

Построим цикл пересчета для $\overline{x}_{ij}$.

Что касается $\overline{x}_{kl}$, возможны следующие ситуации:

1. $\overline{x}_{kl}$ соответствует положительной вершине;
2. $\overline{x}_{kl}$ соответствует отрицательной вершине;
3. $\overline{x}_{kl}$ в цикл не входит.

Рассмотрим эти ситуации подробнее.

1. Если $\overline{x}_{kl}$ соответствует положительной вершине, то при переходе к новому опорному решению $\overline{x}_{kl}$ увеличится на ту же величину, что и $\overline{x}_{ij}$. То есть, если выразить базисные переменные через свободные, переменная $\overline{x}_{ij}$ войдет в выражение для $\overline{x}_{kl}$ с коэффициентом «+1»;

2. Если $\overline{x}_{kl}$ соответствует отрицательной вершине, то при переходе к новому опорному решению $\overline{x}_{kl}$ уменьшится на ту же величину, на которую увеличится $\overline{x}_{ij}$. То есть, если выразить базисные переменные через свободные, переменная $\overline{x}_{ij}$ войдет в выражение для $\overline{x}_{kl}$ с коэффициентом «-1»;

3. Если $\overline{x}_{kl}$ не входит в цикл пересчета, то при переходе к новому опорному решению значение $\overline{x}_{kl}$ не изменится. То есть, если выразить базисные переменные через свободные, переменная $\overline{x}_{ij}$ войдет в выражение для $\overline{x}_{kl}$ с коэффициентом «0».

Схематично эти ситуации можно представить следующим образом:

1. $\overline{x}_{kl}=+1\times\overline{x}_{ij}+\ldots\ldots$
2. $\overline{x}_{kl}=-1\times\overline{x}_{ij}+\ldots\ldots$
3. $\overline{x}_{kl}=0\times\overline{x}_{ij}+\ldots\ldots$

Выразим теперь ЦФ через свободные переменные.

Как подсчитать, с каким коэффициентом переменная $\overline{x}_{ij}$ войдет в ЦФ?

Она войдет в соответствующее выражение непосредственно (с коэффициентом $\overline{x}_{ij}$) и через посредство тех и только тех базисных переменных, которые входят в цикл пересчета.

Но $\overline{x}_{ij}$ входит в выражение для $\overline{x}_{kl}$ только с одним из двух коэффициентов: +1 или -1, в зависимости от того, является вершина $\overline{x}_{kl}$ положительной или отрицательной. А $\overline{x}_{kl}$ входит в ЦФ с коэффициентом c_{kl}. То есть посредством $\bar{x}_{kl}$ (входящей в цикл) переменная $\bar{x}_{ij}$ войдет в ЦФ с коэффициентом $+c_{kl}$ или $-c_{kl}$, в зависимости от того является вершина $\bar{x}_{kl}$ положительной или отрицательной.

Схематично эту ситуацию можно представить следующим образом:

$$Z = c_{ij}\,\bar{x}_{ij} + c_{kl}\,\bar{x}_{kl} + \ldots$$

$$\bar{x}_{kl} = \left\{\begin{matrix} +1 \\ -1 \\ 0 \end{matrix}\right\} x_{ij} + \ldots$$

$$Z = c_{ij}\,\bar{x}_{ij} + c_{kl}\,(\left\{\begin{matrix} +1 \\ -1 \\ 0 \end{matrix}\right\} x_{ij} + \ldots) + \ldots$$

$$Z = (c_{ij} + \left\{\begin{matrix} +1 \\ -1 \\ 0 \end{matrix}\right\} c_{kl} + \ldots)\,\bar{x}_{ij} + \ldots$$

Как видно из этого выражения, в круглых скобках записана алгебраическая сумма всех стоимостей, которым соответствуют вершинам цикла пересчета.

Таким образом, коэффициент, с которым свободная переменная $\bar{x}_{ij}$ входит в ЦФ, равен алгебраической сумме всех стоимостей, соответствующих вершинам цикла, включая вершину, в которой находится свободная переменная.

То есть косвенную стоимость свободной переменной $\bar{x}_{ij}$ можно определить не по формуле $E_{ij} = c_{ij} - (u_i + v_j)$, а другим путем. Нужно построить для этой переменной цикл пересчета и подсчитать алгебраическую сумму всех стоимостей, которым соответствуют вершинам цикла пересчета.

Имеется некоторое опорное решение.

Шаг 1. Для данного опорного решения организуем последовательный просмотр списка свободных клеток (переменных).

Шаг 2. Для очередной свободной клетки (пусть это будет $\overline{x}_{ij}$) строим цикл пересчета и подсчитываем алгебраическую сумму стоимостей по всем вершинам цикла (E_{ij}).

Шаг 3. Если $E_{ij} < 0$, выполняем шаг 4. В противном случае проверяем, все ли свободные клетки просмотрены. Если да, то очередное решение оптимальное. Конец. В противном случае выполняем шаг 2.

Шаг 4. Переменную $\overline{x}_{ij}$ вводим в состав базисных переменных. Для этого среди отрицательных вершин находим вершину с минимальным значением соответствующей базисной переменной. Пусть это будет $\overline{x}_{\lambda\mu}$. Производим сдвиг по циклу пересчета. Переменная $x_{\lambda\mu}$ выводится из состава базисных переменных. Имеем новое опорное решение. Выполняем шаг 1.

Основной недостаток метода – большое количество циклов пересчета, которые приходится строить на каждой итерации. Достоинство – не нужно специально вычислять потенциалы строк и столбцов.

Если нет эффективной процедуры построения цикла пересчета, предпочтение отдается методу потенциалов.

Пример 2.11

Дана транспортная задача и известно ее опорное решение. Определить косвенные стоимости переменных x_{32}, x_{34}, x_{46}.

	15 **50**	20	25	15	10	10 **10**	20
	5 **20**	5	1 **30**	4	2 **50**	1	15
E_{32}**=0**	3	5	5	1	4	5 **40**	2
	20	7 **100**	5	1 20	3	5	4
	20	1 **40**	2 **30**	3	4	6	6 **100**

	15 **50**	20	25	15	10	10 **10**	20
	5 **20**	5	1 **30**	4	2 **50**	1	15
E_{34}**=2**	3	5	5	1	4	5 **40**	2
	20	7 **100**	5	1 20	3	5	4
	20	1 **40**	2 **30**	3	4	6	6 **100**

	15 **50**	20	25	15	10	10 **10**	20
	5 **20**	5	1 **30**	4	2 **50**	1	15
E_{46}**=-2**	3	5	5	1	4	5 **40**	2
	20	7 **100**	5	1 20	3	5	4
	20	1 **40**	2 **30**	3	4	6	6 **100**

2.2.13. Дополнительные ограничения в постановке транспортной задачи

При постановке и нахождении решения конкретных транспортных задач часто бывает необходимо учитывать дополнительные ограничения, которые еще не рассматривались.

Рассмотрим подробно некоторые из таких ограничений.

1. Запрещена перевозка из пункта A_i в пункт B_j. В этом случае вводится очень высокий тариф (M) перевозки из пункта A_i в пункт B_j единицы продукции:

$$c_{ij}=M\ .$$

2. Из пункта A_i в пункт B_j требуется обязательно перевести точно d_{ij} единиц продукции. В клетку на пересечении i-й строки и j-го столбца вносится величина d_{ij}. Корректируется запас a_i и потребность b_j : $a_i' = a_i - d_{ij}$, $b_j' = b_j - d_{ij}$. В дальнейшем эта клетка считается свободной, а для того чтобы соответствующая переменная не попала в состав базисных переменных оптимального решения, этой клетке приписывается очень большой тариф: $c_{ij}=M$.

3. Из пункта A_i в пункт B_j требуется перевести не менее α_{ij} единиц продукции. Считаем, что из пункта A_i в пункт B_j уже перевезено α_{ij} единиц продукции. Уменьшаем запас a_i и потребность b_j на величину α_{ij} : $a_i' = a_i - d_{ij}$, $b_j' = b_j - \alpha_{ij}$. Далее задача решается обычным методом, после чего корректируется полученное решение. Смысл этой корректировки заключается в следующем. Если в оптимальном решении переменная x_{ij} принимает значение $\overline{x}_{ij}$, то в окончательном решении ей приписывается значение $\overline{x}_{ij} = \overline{x}_{ij} + \alpha_{ij}$.

4. Из пункта A_i в пункт B_j требуется перевести не более α_{ij} единиц продукции. В исходную таблицу вводится дополнительный столбец B_j' (рис. 2.11):

	j	
	c_{1j}	
	⋮	
	c_{ij}	
	⋮	
	c_{mj}	
	b_j	

Старая таблица

$\Rightarrow$

	j	j'	
	c_{1j}	c_{1j}	
	⋮	⋮	
	c_{ij}	$\boldsymbol{M}$	
	⋮	⋮	
	c_{mj}	c_{mj}	
	α_{ij}	b_j-α_{ij}	

Новая таблица

Рис.2.11. Моделирование дополнительного ограничения

В окончательном решении принимается $\overline{x}_{ij} = \overline{x}_{ij} + \overline{x}_{ij'}$, $i = \overline{1,m}$.

Естественно, что открывается возможность комбинировать эти приемы и решать задачи с достаточно «серьезными» ограничениями[20].

Пример 2.12

Построить транспортную модель с дополнительными ограничениями.

	1	**2**	**3**	**4**	**5**	
1	5	3	2	4	8	~~160~~ 110
2	7	6	5	3	1	90
3	8	9	4	5	2	~~140~~ 80
	90	~~60~~ 10	80	70	~~90~~ 30	

Дополнительные ограничения:

Из A_1 в B_2 должно быть перевезено не менее 50 ед. ($\boldsymbol{x_{12} \geq 50}$);

Из A_3 в B_5 должно быть перевезено не менее 60 ед. ($\boldsymbol{x_{35} \geq 60}$);

Из A_2 в B_4 должно быть перевезено не более 40 ед. ($x_{24} \leq 40$).

Учтем теперь ограничение $x_{24} \leq 40$. Для этого нужно ввести но-

[20] В частности, достаточно просто учитывается ограничение $d_н \leq x_{ij} \leq d_в$.

вый столбец $4'$.

	1	**2**	**3**	**4**	**5**	**6**$=4'$	
1	5	3	2	4	8	4	110
2	7	6	5	3	1	**M**	90
3	8	9	4	5	2	5	80
	90	10	80	**40**	30	70-40 **30**	

Теперь обычным способом решаем эту задачу (находим исходное опорное решение, а далее, например, используем метод потенциалов). Однако помним, что полученные значения $\overline{x}_{12}$ и $\overline{x}_{35}$ нужно будет увеличить соответственно на 50 и 60 единиц.

2.2.14. Транспортная модель с промежуточными пунктами

Транспортная модель с промежуточными пунктами соответствует реальной ситуации, когда между исходными и конечными пунктами перевозок имеются промежуточные пункты для временного хранения грузов (*транзитные* пункты). Это – более общая, чем обычная транспортная модель, где перевозки осуществляются непосредственно между пунктами отправления и назначения.

Проведем сквозную нумерацию всех пунктов (исходных, конечных, транзитных): $H_1, H_2, \ldots, H_R$.

Транспортную модель представим орграфом, вершинам которого соответствуют пункты $H_1, H_2, \ldots, H_R$.

В том случае, если существует возможность перевозки продукции непосредственно из H_k в H_l, эти вершины свяжем взвешенной дугой (рис. 2.12), направленной из H_k в сторону H_l ($k,l \in \{1,2,\ldots,R\}, k \neq l$). В качестве веса этой дуги примем $c_{k,l}$ – стоимость перевозки единицы продукции из H_k в H_l.

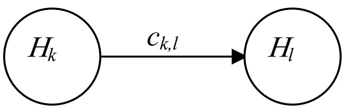


Рис.2.12. Дуга между промежуточными пунктами

Модель предполагает возможность сосредоточения запасов продукции в любом пункте. Кроме того, каждый пункт может иметь собственную потребность в определенном количестве продукции. Этот факт будем отображать на графе следующим образом (рис. 2.13):

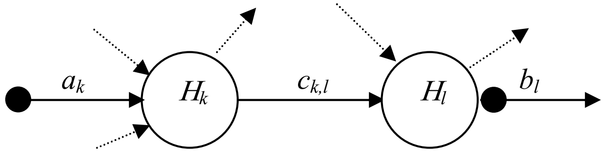


Рис.2.13. Запасы и потребности в модели

Здесь a_k – количество единиц продукции, сосредоточенной в пункте H_k ; b_l – количество единиц продукции, составляющее потребность пункта H_l .

В рассматриваемой модели перевозки транзитом могут осуществляться через любые пункты (в соответствии с направлением взвешенных дуг на орграфе), даже через некоторые пункты назначения.

Поэтому все множество пунктов $H_1, H_2, \ldots, H_R$ можно разбить на три класса.

1. Пункты, которым соответствуют как входящие, так и исходящие взвешенные дуги, назовем *транзитными пунктами* (ТП).
2. Пункты, которым соответствуют только исходящие взвешенные дуги, назовем *истинными пунктами отправления* (ИПО).
3. Пункты, которым соответствуют только входящие взвешенные дуги, назовем *истинными пунктами назначения* (ИПН).

Представленную таким орграфом транспортную модель с промежуточными пунктами преобразуем в обычную (закрытую) транспортную модель с помощью введения, т.н. *буфера*.

Объем буфера (B) должен быть таким, чтобы вместить объем всего предложения (или спроса):

B = Общий объем предложения (спроса).

Объемы спроса (предложения) перечисленных выше трех подмножеств пунктов определяются следующим образом:

- объем предложения ИПО = объем исходного предложения;
- объем спроса ИПН = объем исходного спроса;
- объем предложения ТП = объем исходного предложения + B;

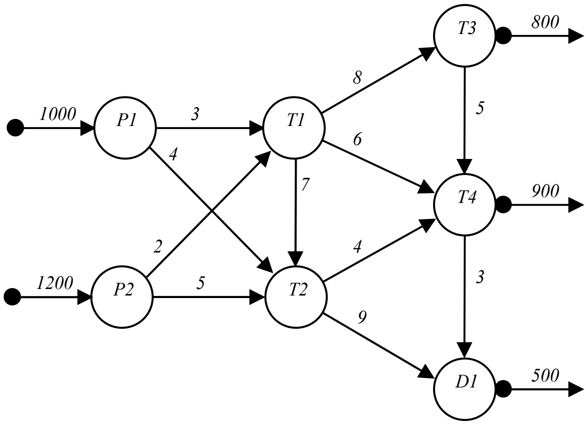
- объем спроса ТП = объем исходного спроса + *B*.

При построении закрытой транспортной модели в качестве поставщиков принимаются все ИПО и все ТП. В качестве потребителей – все ТП и все ИПН.

Для того чтобы запретить перевозки между пунктами, не связанными в исходном орграфе взвешенными дугами, соответствующим маршрутам (запрещенным), приписывается достаточно высокая стоимость (M). Маршрутам, соответствующим петлям (из одного ТП в тот же самый ТП) приписывается нулевая стоимость.

Ниже рассматривается пример решения транспортной задачи с промежуточными пунктами.

Пример 2.13



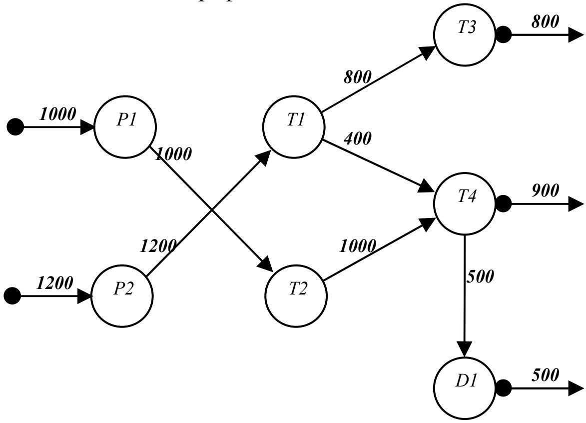
Здесь *P1* и *P2* – ИПО; T1, T2, T3, T4 – ТП; D1 – ИПН.

B = 1000+1200=800+900+500=2200.

Строим транспортную таблицу и решаем задачу:

	T1	T2	T3	T4	D1	
P1	3	4 1000	M	M	M	**1000**
P2	2 1200	5	M	M	M	**1200**
T1	0 1000	7	8 800	6 400	M	**2200**
T2	M	0 1200	M	4 1000	9	**2200**
T3	M	M	0 2200	5	M	**2200**
T4	M	M	M	0 1700	3 500	**2200**
	2200	**2200**	**3000**	**3100**	**500**	**11000**

Решение задачи на графе:



Контрольные вопросы и задачи к разделу 2.2

1. Сравните метод Фогеля и метод плавающих зон с точки зрения вычислительных затрат на решение задачи и степени приближенности получаемого приближенного решения к оптимальному.

2. Дана транспортная задача. Эта задача решается методом Фогеля. Имеем значение ЦФ $Z_{исх}$. Введем в эту задачу дополнительное ограничение (например, запретим какой-либо маршрут). Получим значение ЦФ $Z_{доп}$. Можно ли утверждать следующее и почему:

а) $Z_{исх} \leq Z_{доп}$

б) $Z_{исх} \geq Z_{доп}$?

3. Возможен ли в методе потенциалов случай зацикливания, аналогичный тому, который возможен в симплекс-методе в случае вырожденности задачи?

4. Докажите утверждение, что косвенная стоимость – это коэффициент, с которым свободная переменная входит в целевую функцию, если все базисные переменные выразить через свободные.

5. Каким образом можно привести открытую транспортную задачу с промежуточными пунктами к закрытой?

6. Решите задачи методами северо-западного угла, наименьшей стоимости, Фогеля и методом плавающих зон:

	1	2	3	4	
1	11	10	6	17	55
	17	11	16	7	40
3	9	14	15	9	80
	50	40	45	40	

	1	2	3	4	
1	16	6	15	17	45
2	10	7	6	6	50
3	10	7	9	10	115
	55	55	45	55	

	1	2	3	4	
1	11	7	12	9	60
2	12	15	16	17	45
3	15	12	11	14	115
	60	60	55	45	

7. В процессе решения задачи с помощью метода потенциалов была получена следующая транспортная таблица:

	1	2	3	4
1	16	15 20	12 25	11 10
2	8 40	15 20	16	16
3	17	6	12	7 60

Продолжите решение задачи.

8. Решите задачу распределительным методом, применив метод наименьшей стоимости для поиска исходного опороного решения:

	1	2	3	4	
1	16	10	14	9	60
2	7	16	6	15	45
3	15	8	12	17	115
	60	60	55	45	

9. Постройте закрытую транспортную модель на основании следующей таблицы:

с дополнительными ограничениями

	1	2	3	4	
1	11	11	16	7	55
2	9	10	10	15	60
3	16	15	15	10	105
	60	55	50	55	

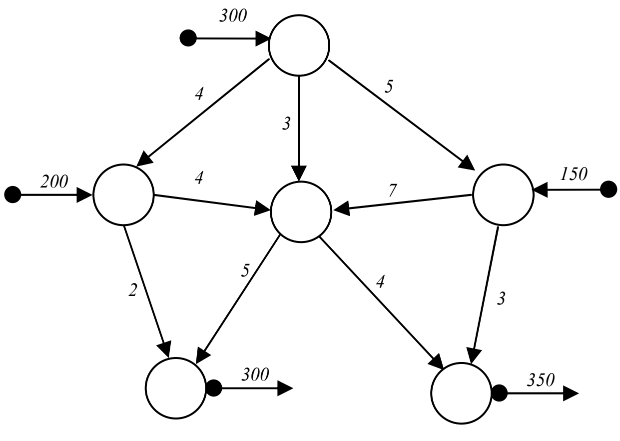
а) запрещена перевозка из пункта А1 в пункт В3;

б) из пункта А2 в пункт В4 необходимо перевезти ровно 30 единиц продукции;

в) из пункта А1 в пункт В2 необходимо перевезти не менее 45 единиц продукции;

г) из пункта А3 в пункт В1 можно перевезти не более 5 единиц продукции.

10. Решите транспортную задачу с промежуточными пунктами:



2.3. Метод отсечения

2.3.1. Основные понятия метода отсечения

В своей основе метод отсечения использует тот факт, что, если все вершины многогранного множества D имеют целочисленные координаты, то и оптимальное опорное решение соответствующей задачи ЛП будет удовлетворять требованию целочисленности.

Рассмотрим ЛЦП-задачу:

$$\sum_{j=1}^{n} c_j x_j \to \max \tag{2.23}$$

$$D_ц \begin{cases} D \begin{cases} \sum\limits_{j=1}^{n} a_{ij} x_j = a_i, \; i = \overline{1,m} & (2.24) \\ x_j \ge 0, \; j = \overline{1,n} & (2.25) \end{cases} \\ x_j - \text{целые}, \; j = \overline{1,n} & (2.26) \end{cases}$$

ЛП

ЛЦП

Допустимое множество этой задачи, определенное ограничениями (2.24)-(2.26), обозначим $D_ц$.

Очевидно, $D_ц$ – множество *целых* точек выпуклого многогранного множества D, определенного ограничениями (2.24)-(2.25).

С целью упрощения последующих выкладок будем считать, что D – замкнутое ограниченное множество. То есть $D_ц$ – конечное множество.

Обозначим $D_ц^*$ выпуклую линейную оболочку множества $D_ц$.

Множество $D_ц^*$ называется ***выпуклой линейной оболочкой*** множества $D_ц = \{x^1, x^2, ..., x^N\}$, если содержит все возможные выпуклые линейные комбинации вида:

$$\sum_{i=1}^{N} \alpha_i x^i, \quad \alpha_i \ge 0 \; (i = \overline{1,N}), \sum_{i=1}^{N} \alpha_i = 1.$$

Что собой представляет выпуклая линейная оболочка $D_ц^*$?

Ее свойства можно определить из геометрической интерпретации (рис. 2.14).

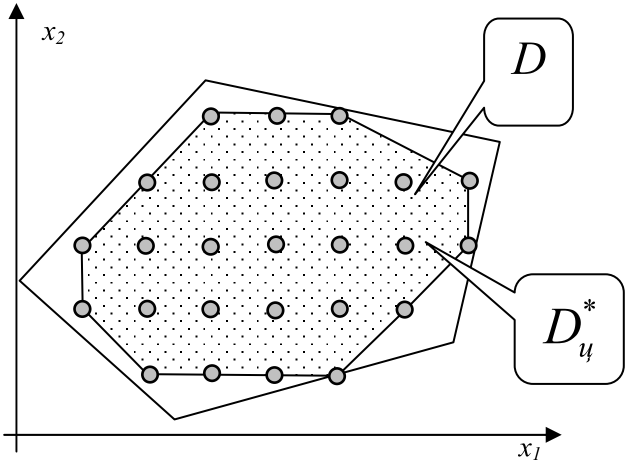


Рис.2.14. Геометрическая интерпретация выпуклой линейной оболочки

Рассмотрим эти свойства подробно.

1. Множество целых точек выпуклой линейной оболочки $D_ц^*$ совпадает с множеством $D_ц$.
2. $D_ц^*$ – выпуклый многогранник, имеющий целочисленные координаты всех вершин (целочисленный многогранник).
3. Любой оптимальный опорный план задачи:

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

$$x \in D_ц^*$$

$$x = (x_1, x_2, \dots x_n)$$

является оптимальным решением ЛЦП-задачи (2.23)-(2.26).

Таким образом, если удается построить выпуклую линейную оболочку множества $D_ц$, соответствующую задачу можно решить обычным симплекс-методом и автоматически получить решение исходной ЛЦП-задачи (2.23)-(2.26). Однако даже для задач небольшой размерности построение выпуклой линейной оболочки – чрезвычайно трудоемкий, сложный вычислительный процесс.

2.3.2. Идея методов отсечения

Предположим, что мы владеем эффективной процедурой построения последовательности задач линейного программирования: ЛП(0), ЛП(1),…, ЛП(**k**),… каждая из которых определяется своим множеством допустимых решений $D^{(0)}, D^{(1)},\ldots, D^{(k)},\ldots$ и *одной и той же* целевой функцией.

То есть

$$\text{ЛП}(k) \Rightarrow \begin{cases} \sum_{j=1}^{n} c_j x_j \to \max \\ x \in D^{(k)},\ x = (x_1, x_2, \ldots, x_n) \end{cases}$$

При этом последовательность задач ЛП(0), ЛП(1), …, ЛП(k),… обладает следующими свойствами:

1. $D^{(0)} \equiv D$, т.е. в качестве $D^{(0)}$ принимается многогранное множество, определенное ограничениями (2.24) и (2.25) исходной ЛЦП-задачи;
2. $D_ц^{(k)} = D_ц$, (k=0,1,2,…), где $D_ц^{(k)}$ – множество целых точек из $D^{(k)}$, $D_ц$ – допустимое множество исходной ЛЦП-задачи;
3. Если оптимальное решение $\overline{x}^k$ задачи ЛП(k), полученное на каком-то шаге k, имеет целочисленные координаты, то $\overline{x}^k$ – оптимальное решение исходной ЛЦП-задачи;
4. Если оптимальное решение $\overline{x}^k$ задачи ЛП(k), полученное на каком-то шаге k, имеет не целые координаты, то $\overline{x}^k$ – не является допустимым решением задачи ЛП($k+1$), т.е. $\overline{x}^k \notin D^{(k+1)}$.

Интуитивно ясно, что последовательное построение задач ЛП(0), ЛП(1), …, ЛП(k) – это, в некотором смысле, аппроксимация выпуклой линейной оболочки $D_ц^*$ множествами $D^{(k)}$ ($k=1,2,..$).

Способы построения последовательности задач {ЛП(k)}, обеспечивающие конечность процесса решения исходной ЛЦП-задачи, были предложены американским математиком Р. Гомори.

Р. Гомори разработал три алгоритма:

- для полностью целочисленных задач;
- для частично-целочисленных задач;
- для полностью целочисленных задач с использованием только операций сложения и умножения.

2.3.3. Правильное отсечение в алгоритме Гомори

Итак, в основу метода отсечения положено последовательное усечение исходного множества D с целью построения части выпуклой линейной оболочки в области максимума ЦФ.

В этой связи становится ясно, что любая задача ЛП(k+1) в последовательности задач {ЛП(k)} отличается от задачи ЛП(k) некоторым ограничением, которое должно обладать соответствующими свойствами.

Рассмотрим две эти задачи (рис. 2.15).

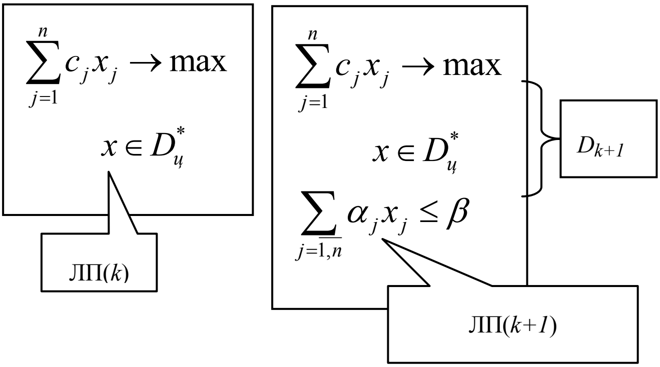


Рис. 2.15 Две задачи

Ограничение

$$\sum_{j=\overline{1,n}} \alpha_j x_j \le \beta \qquad (2.27)$$

называется <u>*правильным отсечением*</u>, если это ограничение удовлетворяет следующим требованиям:

1. Пусть $x^* = (x_1^*, x_2^*, ..., x_n^*)$ – оптимальное решение задачи ЛП(k), имеющее не целые координаты. Тогда $\sum_{j=\overline{1,n}} \alpha_j x_j^* > \beta$, т.е., ограничение (2.27) не выполняется (условие отсечения);

2. Пусть $x^0 = (x_1^0, x_2^0, ..., x_n^0)$ – любое целочисленное решение задачи ЛП(k). Тогда: $\sum_{j=\overline{1,n}} \alpha_j x_j^0 \le \beta$, т.е., ограничение (2.27) выполняется (условие правильности).

Линейный вид ограничения (2.27) позволяет использовать методы ЛП.

Как по решению задачи ЛП(k) построить ограничение – правильное отсечение для того, чтобы сформировать новую задачу последовательности ЛП(k+1)?

Предварительно нужно строго определить целую и дробную часть числа.

Целую часть произвольного вещественного числа α обозначим $[\alpha]$:

$[\alpha]$ – наибольшее целое число, не превосходящее α.

Дробной частью произвольного вещественного числа α называется число $\{\alpha\}$: $\{\alpha\}=\alpha - [\alpha]$.

Пример 2.14

| {7/3}=1/3, {-7/3}=2/3.

Решим задачу ЛП(0) – исходную задачу (2.23)-(2.25), игнорируя требование целочисленности (2.26).

Пусть $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ – оптимальное решение этой задачи. Пусть далее для определенности:

$\sigma = \{1, 2, ..., m\}$ – номера базисных переменных;

$\omega = \{m+1, m+2, ..., n\}$ – номера свободных переменных.

Если все координаты решения $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ – целые, то получено оптимальное решение исходной ЛЦП-задачи. В противном случае решение нужно продолжить.

Рассмотрим оптимальную симплекс-таблицу решенной задачи (рис. 2.16).

$$x_{10} = 1 * x_1 + \ldots + 0 * x_m + x_{1,m+1} * x_{m+1} + \ldots + x_{1,n} * x_n$$
$$x_{20} = 0 * x_1 + \ldots + 0 * x_m + x_{2,m+1} * x_{m+1} + \ldots + x_{2,n} * x_n$$
$$x_{m0} = 0 * x_1 + \ldots + 1 * x_m + x_{m,m+1} * x_{m+1} + \ldots + x_{m,n} * x_n$$
....

Разложение базисных переменных

Разложение свободных переменных

Рис.2.16. Оптимальная симплекс-таблица как задача ЛП

Пусть l – индекс некоторой нецелой координаты:
$$l \in \sigma = \{1, 2, \ldots, m\} .$$
Выпишем l-е уравнение (l-я строка симплекс-таблицы):
$$x_{l0} = x_l + \sum_{j \in \omega} x_{lj} x_j , \ (l \in \sigma) . \qquad (2.28)$$
А теперь рассмотрим следующее выражение:

$-\{x_{l0}\} = y_l + \sum_{j \in \omega} (-\{x_{lj}\} x_j)$, где y_l – некоторая переменная:
$$y_l = -\{x_{l0}\} + \sum_{j \in \omega} \{x_{lj}\} x_j . \qquad (2.29)$$
Относительно этого выражения справедлива следующая теорема.

ТЕОРЕМА 2.2 (о правильном отсечении)

Если $x = (x_1, x_2, \ldots, x_n)$ *– любое целочисленное решение (не обязательно опорное) задачи (2.23)-(2.26), т.е. допустимое решение ЛЦП-задачи, то:*
$$y_l \geq 0, \quad y_l \text{ – целое число.}$$
<u>Доказательство</u>. Перепишем (2.28):
$$x_l = x_{l0} - \sum_{j \in \omega} x_{lj} x_j .$$
$$x_l = [x_{l0}] + \{x_{l0}\} - \sum_{j \in \omega} [x_{lj}] x_j - \sum_{j \in \omega} \{x_{lj}\} x_j ,$$
$$x_l - [x_{l0}] + \sum_{j \in \omega} [x_{lj}] x_j = \{x_{l0}\} - \sum_{j \in \omega} \{x_{lj}\} x_j = -y_l .$$
Но x_l и x_j – целые, т.е. правая часть – целое число. Следовательно, и левая часть – также целое число.

Пусть теперь $y_l < 0$ (от противного). То есть

$$-\{x_{l0}\} + \sum_{j\in\omega}\{x_{lj}\}x_j < 0 .$$

Но $-\{x_{l0}\} > -1$, а $\{x_{lj}\} \geq 0$ и $x_j \geq 0$. Значит, имеет место $-1 < -\{x_{l0}\} + \sum_{j\in\omega}\{x_{lj}\}x_j < 0$, чего не может быть, так как $-\{x_{l0}\} + \sum_{j\in\omega}\{x_{lj}\}x_j$ – целое число. Полученное противоречие говорит о неправомерности предположения о том, что $y_l < 0$. Теорема доказана.

__Следствие__

Любое оптимальное опорное решение $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$ *ЛП-задачи (2.23)-(2.25), имеющее нецелую координату* $\overline{x}_l$*, не удовлетворяет условию:*

$$\left.\begin{aligned} y_l &= -\{x_{l0}\} + \sum_{j\in\omega}\{x_{lj}\}x_j \\ y_l &\geq 0 \end{aligned}\right\} \qquad (2.30)$$

__**Доказательство**__. Действительно, если $\overline{x}_l$ – не целое, то по определению $\{x_{l0}\} > 0$. Но все свободные переменные в опорном решении имеют нулевое значение. То есть $y_l = -\{x_{l0}\} < 0$. Следовательно, ограничение (2.30) не выполняется, что и требовалось доказать.

Итак, по решению некоторой задачи можно построить правильное отсечение.

2.3.4. Первый алгоритм Гомори

В соответствии с общей схемой метода отсечения будем строить последовательность ЛП-задач:

ЛП(0), ЛП(1),…, ЛП(k),…

Переменную y_l, которая определяется дополнительным ограничением (2.30) и строится по некоторой нецелой координате оптимального опорного решения задачи ЛП(k), обозначим x_{n+k+1} (считаем, что ЛП(0) – исходная задача без требования целочисленности переменных).

Через ω_k обозначим множество номеров свободных переменных в оптимальном опорном решении задачи ЛП(k).

Шаг 0. Положим $k=0$ и решим задачу ЛП(k).

Шаг 1. Если все базисные переменные оптимального решения – целые, то исходная ЛЦП-задача решена. Конец.

Если задача не имеет решения, то и исходная ЛЦП-задача также не имеет решения. Конец.

Шаг 2. Среди нецелых координат x_{i0} выбирается любая координата (обычно – с максимальной дробной частью) . Пусть это будет x_{l0}.

Шаг 3. По координате x_{l0} строится дополнительное ограничение:

$$x_{n+k+1} = -\{x_{l0}\} + \sum_{j\in\omega_k} \{x_{lj}\} x_j \ , \tag{2.31}$$

где x_{n+k+1} – дополнительная переменная, ω_k – множество номеров свободных переменных в оптимальном опорном решения задачи ЛП(k). Ограничение (2.31) вносится в состав ограничений задачи ЛП(k): таким образом, формируется задача ЛП($k+1$).

Формально, введение дополнительного ограничения – это дополнение симплекс-таблицы новой строкой

$$-\{x_{l0}\} = x_{n+k+1} + \sum_{j\in\omega_k} (-\{x_{lj}\}) x_j$$

и одним столбцом – единичным вектором A_{n+k+1} с единицей в позиции, соответствующей новой строке (рис. 2.17).

Баз	$C_{баз}$		c_1	c_2	……	c_l	……	*0*
		A_0	A_1	A_2		A_l		A_{n+k+1}
…………………………………………………………………..								
A_l	c_l	x_{l0}	x_{l1}	x_{l2}	……	*1*	……	*0*
…………………………………………………………………..								
$\boldsymbol{A_{n+k+1}}$	**0**	$\boldsymbol{-\{x_{l0}\}}$	$\boldsymbol{-\{x_{l1}\}}$	$\boldsymbol{-\{x_{l2}\}}$	…		…	**1**
		Δ	≥ 0	≥ 0		*0*		*0*

Рис.2.17. Новая симплекс-таблица

В новой задаче переменными являются (рис. 2.18):

$$\underbrace{\underbrace{\underbrace{x_1, x_2, \ldots, x_n}_{\text{ЛП(0)}}, x_{n+1}, x_2, \ldots, x_{n+k}}_{\text{ЛП(k)}}, x_{n+k+1}}_{\text{ЛП(k+1)}}$$

Рис.2.18. Переменные полученной задачи

Что теперь записано в симплекс-таблице?

- Оценки не изменились – все они неотрицательные (имеем оптимальное решение ЛП(k) и одну нулевую оценку введенного вектора $A_{\mathrm{n+k+1}}$).
- Но в разложении вектора A_0 появилась отрицательная координата.

Следовательно, в симплекс-таблице записан псевдоплан задачи ЛП(k+1)

<u>Шаг 4</u>. Применяем двойственный симплекс-метод для решения задачи ЛП(k+1). Очевидно, что на первой итерации вектор $A_{\mathrm{n+k+1}}$ выйдет из состава базисных векторов, так как ему соответствует единственная отрицательная координата ПДО-решения. Решаем задачу до конца. Принимаем k=k+1 и переходим к шагу 1.

Хотя это интуитивно ясно, существует доказательство конечности рассмотренного метода: за конечное число шагов будет найдено целочисленное решение задачи ЛЦП, либо будет установлена неразрешимость этой задачи.

Пример 2.15

Исходная задача

$x_1+2x_2 \to max$

$2x_1+\ x_2 \le 8$

$x_2 \le ,$

$x_{1,2} \ge 0$

$x_{1,2}$ – целые

Приведение к

Задача ЛП(0)

$x_1+2x_2 \to max$

$2x_1+\ x_2+x_3\ \ \ = 8$

$x_2\ \ \ +x_4 = 3$

$x_{1,2,3,4} \ge 0$

$x_{1,2,3,4}$ – целые

Баз	$C_{баз}$		*1*	*2*	*0*	*0*
		A_0	A_1	A_2	A_3	A_4
A_3	*0*	*8*	*2*	*1*	*1*	*0*
A_4	*0*	*3*	*0*	*1*	*0*	*1*
Табл. 1		0	-1	-2	0	0
A_3	*0*	*5*	*2*	*0*	*1*	*-1*
A2	*2*	*3*	*0*	*1*	*0*	*1*
Табл.2		6	-1	0	0	2
A1	*1*	*5/2*	*1*	*0*	*1/2*	*-1/2*
A2	*2*	*3*	*0*	*1*	*0*	*1*
Табл.3		17/2	0	0	1/2	3/2

По первой координате (не целой) строим дополнительное ограничение, вводим в состав базисных векторов вектор A_5 и заполняем симплекс-таблицу для задачи ЛП(1):

Баз	$C_{баз}$		1	2	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
A1	*1*	*5/2*	*1*	*0*	*1/2*	*-1/2*	*0*
A2	*2*	*3*	*0*	*1*	*0*	*1*	*0*
A_5	*0*	*-1/2*	*0*	*0*	*-1/2*	*-1/2*	*1*
Табл. 1		17/2	0	0	1/2	3/2	0

Применяем двойственный симплекс-метод:

Баз	$C_{баз}$		1	2	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
A1	*1*	*2*	*1*	*0*	*0*	*-1*	*1*
A2	*2*	*3*	*0*	*1*	*0*	*1*	*0*
A3	*0*	*1*	*0*	*0*	*1*	*1*	*-2*
Табл.2		8	0	0	0	1	1

Имеем целочисленное решение:

Геометрическая интерпретация

$$\begin{aligned}
&x_1+2x_2 \to max\\
&2x_1+\ x_2+x_3\quad = 8\\
&x_2\quad +x_4=3\\
&x_{1,2,3,4} \geq 0\\
&x_{1,2,3,4} - \textit{целые}.
\end{aligned}$$

Выразим переменные x_3 и x_4 через x_1 и x_1:

$$\left.\begin{aligned} x_3 &= 8 - 2x_1 - x_2\\ x_4 &= 3 - x_2 \end{aligned}\right\} \text{Т.к. } x_3,\, x_4 \geq 0 \quad \left\{\begin{aligned} 2x_1+\ x_2 &\leq 8\\ x_2 &\leq 3. \end{aligned}\right.$$

			1	2	0	0	0
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4	A_5
A1	*1*	*5/2*	*1*	*0*	*1/2*	*-1/2*	*0*
A2	*2*	*3*	*0*	*1*	*0*	*1*	*0*
A_5	*0*	*-1/2*	*0*	*0*	*-1/2*	*-1/2*	*1*
Табл. 1		17/2	0	0	1/2	3/2	0

При введении дополнительного ограничения мы пользовались выражением:

$$y_i = -\{x_{i0}\} - \sum_{j\in\omega}(-\{x_{ij}\})x_j$$

В нашем случае это

$$x_5 = -1/2 - (-1/2\ x_3) - (-1/2\ x_4) = -1/2 + 1/2\ x_3\ + 1/2\ x_4.$$

Ввиду того, что на переменную x_5 наложено требование неотрицательности ($x_5 \geq 0$), дополнительное ограничение можно переписать так:

$$1/2\ x_3\ + 1/2\ x_4 \geq 1/2 \text{ или } x_3 + x_4 \geq 1.$$

Но $x_3 = 8 - 2x_1 - x_2$ и $x_4 = 3 - x_2$. Следовательно

$$8 - 2x_1 - x_2 + 3 - x_2 \geq 1 \to 2x_1 + 2x_2 \leq 10 \to x_1 + x_2 \leq 5$$

Итак, задача ЛП(1) имеет вид (рис. 2.19):

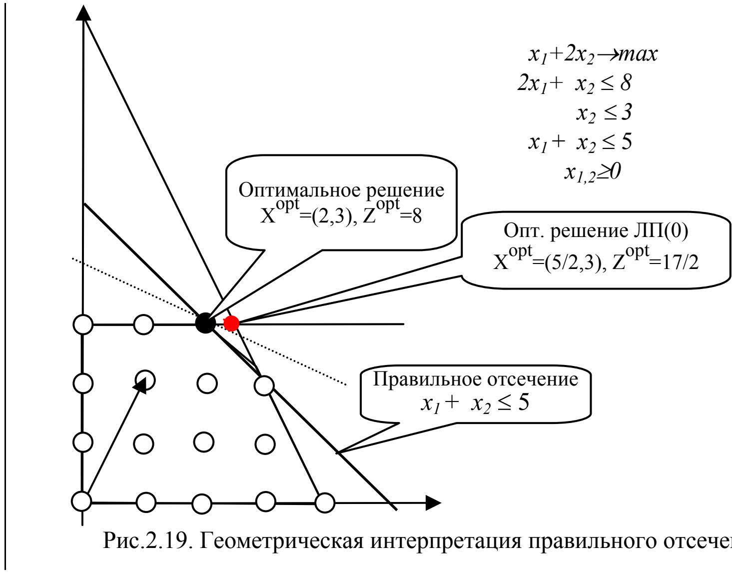


Рис.2.19. Геометрическая интерпретация правильного отсечения

Перед тем, как закончить изложение метода отсечения, остановимся на некоторых проблемах, связанных с использованием этого метода.

2.3.5. Проблемы первого алгоритма Гомори

В представленной редакции алгоритма, по мере увеличения количества задач, растет и размерность этих задач. Действительно, на каждой большой итерации в состав базисных переменных вводится новая дополнительная переменная и соответственно новое ограничение.

При этом количество свободных переменных не изменяется.

Так, размерность последней задачи $(m+k)\times(n+k)$, где

$(m+k)$ – количество строк,

$(n+k)$ – количество столбцов.

По сути дела, все зависит от количества итераций, а оно может быть очень большим.

Гомори предложил и обосновал следующий прием уменьшения размерности.

Если при решении задачи ЛП(r) в базис оптимального решения войдет некоторая переменная x_{n+k+1} $(k < r)$, то соответствующая строка и столбец просто вычеркиваются из симплекс-таблицы.

Таким образом, в самом худшем случае в базис войдут все векторы исходной задачи, а их n. Количество же свободных переменных – одно и то же (n-m). Следовательно, в худшем случае размерность задачи составит:

$$n\times(n+(n-m))=n\times(2n-m).$$

Есть в этом алгоритме и другие проблемы, на которых следует остановиться.

Ошибки округления. Эти ошибки, возникающие в процессе вычислений, могут приводить к получению неверного решения.

В процессе реализации алгоритма все промежуточные решения – оптимальные решения ЛП(k)-задач – не являются допустимыми решениями исходной ЛЦП-задачи. Это не позволяет получить промежуточное целочисленное решение, отличное от оптимального, что делает проблематичным использование этого алгоритма для приближенного решения задач ЛЦП.

Наконец, последнюю проблему рассмотрим на примере.

Пример 2.16

Исходная задача

$$\begin{aligned} &2x_1+x_2\to max\\ &x_1+x_2\le 8\\ &x_1 \le 5/2\\ &x_{1,2}\ge 0\\ &x_{1,2} - \text{целые} \end{aligned}$$

Казалось бы

Задача ЛП(0)

$$\begin{aligned} &2x_1+x_2\to max\\ &x_1+x_2+x_3 = 8\\ &x_1 + x_4 = 5/2\\ &x_{1,2,3,4}\ge 0\\ &x_{1,2,3,4} - \text{целые} \end{aligned}$$

Решаем задачу с помощью алгоритма Гомори:

Баз	$C_{баз}$	A_0	*2*	*1*	*0*	*0*
			A_1	A_2	A_3	A_4
A_3	*0*	*8*	*1*	*1*	*1*	*0*
A_4	*0*	*5/2*	*1*	*0*	*0*	*1*
Табл.1		0	-2	-1	0	0
A_3	*0*	*11/2*	*0*	*1*	*1*	*-1*
A_1	*2*	*5/2*	*1*	*0*	*0*	*1*
Табл.2		5	0	-1	0	2
A_2	*1*	*11/2*	*0*	*1*	*1*	*-1*
A_1	*2*	*5/2*	*1*	*0*	*0*	*1*
Табл.3		*21/2*	*0*	*0*	*1*	*1*

Баз	$C_{баз}$	A_0	2	1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_2	1	11/2	0	1	1	-1	0
A_1	2	5/2	1	0	0	1	0
A_5	0	-1/2	0	0	0	0	1
Табл.4		21/2	0	0	1	1	0

При работе по ДСМ срабатывает признак недопустимости.

Проблема заключается в том, что 1-й алгоритм Гомори предполагает целочисленность всех переменных, в том числе и дополнительных. Поэтому достаточным условием правильной работы этого алгоритма является целочисленность всех параметров задачи.

Правильное решение

Исходная задача

$$2x_1+x_2 \to max,$$
$$x_1+x_2 \le 8,$$
$$x_1 \le 5/2,$$
$$x_{1,2} \ge 0,$$
$$x_{1,2} – \text{целые}.$$

$x_1 \le 5/2$

Заменяем на

Задача ЛП(0)

$$2x_1+x_2 \to max,$$
$$x_1+x_2+x_3 = 8,$$
$$2x_1 + x_4 = 5,$$
$$x_{1,2,3,4} \ge 0,$$
$$x_{1,2,3,4} – \text{целые}.$$

Баз	$C_{баз}$	A_0	2	1	0	0
			A_1	A_2	A_3	A_4
A_3	0	8	1	1	1	0
A_4	0	5	2	0	0	1
Табл. 1		0	-2	-1	0	0
A_3	0	11/2	0	1	1	-1/2
A1	2	5/2	1	0	0	1/2
Табл.2		5	0	-1	0	1
A2	1	11/2	0	1	1	-1/2
A1	2	5/2	1	0	0	1/2
Табл.3		21/2	0	0	1	1/2

Баз	$C_{баз}$	A_0	2	1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_2	1	11/2	0	1	1	-1/2	0
A_1	2	5/2	1	0	0	1/2	0
A_5	0	-1/2	0	0	0	-1/2	1
Табл.4		21/2	0	0	1	1/2	0
A_2	1	6	0	1	1	0	-1
A_1	2	2	1	0	0	0	1
A4	0	1	0	0	0	1	-2
Табл.5		10	0	0	1	0	1

Получено оптимальное целочисленное решение. Геометрическая интерпретация приведена на рис. 2.20.

Задача ЛП(0)

$$2x_1+x_2\to max,$$

$$\begin{cases} x_1+x_2+x_3=8, \\ 2x_1+x_4=5, \end{cases} \Rightarrow \begin{cases} x_3=8-x_1-x_2\geq 0 \\ x_4=5-2x_1\geq 0 \end{cases} \Rightarrow \begin{cases} x_1+x_2\leq 8 \\ 2x_1\leq 5 \end{cases}$$

$$x_{1,2,3,4}\geq 0,$$

$x_{1,2,3,4}$ – *целые.*

Дополнительное ограничение:

$$-1/2=x_5-1/2x_4 \Rightarrow x_5=-1/2+1/2x_4\geq 0 \Rightarrow -1/2+1/2(5-2x_1)=2-x_1\geq 0$$

$$x_1\leq 2$$

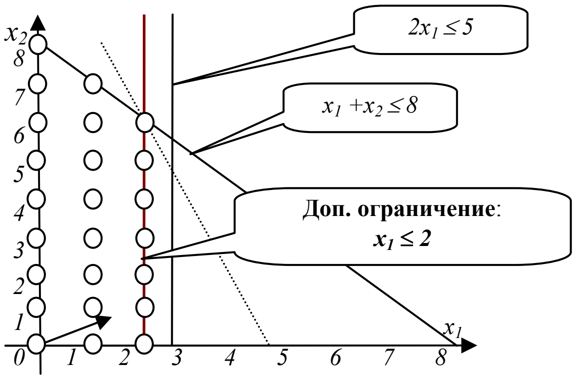


Рис. 2.20. Геометрическая интерпретация правильного отсечения

Контрольные вопросы и задачи к разделу 2.3

1. Дайте определение выпуклой линейной оболочки допустимого множества задачи ЛЦП.

2. Чем отличается выпуклая линейная оболочка множества от самого множества?

3. Назовите основные требования, которым должно отвечать правильное отсечение.

4. Выпишите уравнение правильного отсечения в первом алгоритме Гомори.

5. Какие особенности возникают при решении с помощью первого алгоритма Гомори задач с допустимым множеством, представленным в виде нестрогих неравенств?

6. Изобразите допустимое множество задачи линейного целочисленного программирования на графике. Нарисуйте границы выпуклой линейной оболочки допустимого множества. Получите аналитическое представление выпуклой линейной оболочки.

$$x_1 + 2x_2 \to \max$$
$$x_1 + 4x_2 \le 21.5$$
$$2x_1 \le 11$$
$$2x_2 \le 9$$
$$x_1, x_2 \ge 0, \textit{целые}$$

7. Дана задача линейного целочисленного программирования:

$$x_1 + 2x_2 \to \max$$
$$2x_1 + x_2 \le 13$$
$$-x_1 + 2x_2 \le 5$$
$$x_1 \le 5$$
$$x_2 \le 4$$
$$x_1, x_2 \ge 0, \textit{целые}$$

Исходя из определения, проверьте, являются ли следующие отсечения правильными:

а) $-x_1 + x_2 \le 2$

б) $x_1 + 2x_2 \le 11$

в) $x_1 + x_2 \le 8$

8. Решите задачу ЛЦП с помощью первого алгоритма Гомори

$$x_1 + 10x_2 \to \max$$
$$x_1 + 8x_2 \le 28$$
$$x_1 \le 4$$
$$x_1, x_2 \ge 0, \textit{целые}$$

9. Решите задачу ЛЦП с помощью первого алгоритма Гомори

$$x_1 + x_2 \to \max$$
$$-8x_1 + 15x_2 \le 15$$
$$x_1 \le 7.5$$
$$x_1, x_2 \ge 0, \textit{целые}$$

2.4. Метод ветвей и границ

Среди методов, ориентированных на точное и /или приближенное решение задач комбинаторного типа, особое место занимает группа методов, объединенных по общим названием "Метод ветвей и границ" (МВиГ).

Различные реализации МВиГ объединяет общая идея – идея замены полного перебора направленным, эвристическим поиском допустимых решений (комбинаций). Эффект достигается за счет массового отсеивания так называемых "бесперспективных" вариантов.

В основу различных модификаций МВиГ положено несколько принципов, которые рассматриваются в следующем разделе

2.4.1. Принципы метода ветвей и границ

- Вычисление верхней или нижней границы (оценки) ЦФ на допустимом множестве или некотором его подмножестве;
- Последовательное разбиение допустимого множества (ветвление), шаг за шагом сокращающее допустимое множество;
- Пересчет оценок (границ);
- Установление признака оптимальности;
- Оценка точности приближенного решения.

Вычисление границы

Для определенности рассмотрим в самой общей постановке задачу комбинаторного типа на минимум ЦФ:

$$F(x) \to min$$
$$x \in G.$$

Здесь G – множество допустимых решений (комбинаций); x – элемент множества G.

Иногда удается найти нижнюю границу (оценку) целевой функции F на G или на некотором его подмножестве $G' \subset G$.

Нижняя граница – это такое число $\eta(G)$ (или $\eta(G')$), что для $\forall x \in G$ имеет место $F(x) \geq \eta(G)$. Способ вычисления нижней границы существенно зависит от содержательной постановки задачи.

Например, в задаче поиска самого короткого маршрута из некоторой точки А в точку В с учетом существующей транспортной сети в качестве нижней границы можно взять евклидово расстояние между этими точками (короче пути нет и не может быть).

Ветвление

Реализация метода ветвей и границ связана с постепенным разбиением допустимого множества G на подмножества меньшей мощности – с представлением этого подмножества в виде дерева подмножеств. Принцип, по которому осуществляется разбиение, также зависит от конкретной задачи (рис. 2.21).

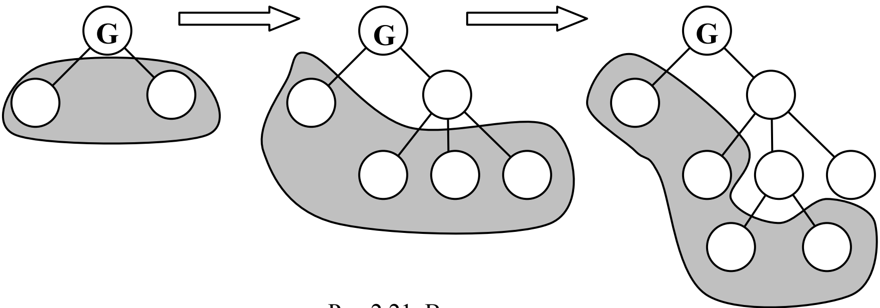


Рис.2.21. Ветвление задачи

Пересчет оценок

Пересчет оценок основан на очевидном факте. Если $G_1 \subset G_2$, то должно иметь место

$$\min_{\mathbf{X}\in\mathbf{G}_1} F(x) \geq \min_{\mathbf{X}\in\mathbf{G}_2} F(x).$$

Учитывая это обстоятельство, предполагается, что если разбить некоторое множество $G' \subset G$ на подмножества

$G'_1, G'_2, \ldots, G'_s$, причем $G' = \bigcup_{i=\overline{1,s}} G'_i$, то оценка любого подмножества должна быть не меньше оценки подмножества G':

$$\eta(G'_i) \geq \eta(G'),\ \ i = \overline{1,s}.$$

Таким образом, после разбиения любого подмножества, необходимо вычислить оценки всех вновь образованных подмножеств – уточнить оценки.

Признак оптимальности

Пусть $G = \bigcup_{i=\overline{1,s}} G_i$, и известно решение $\overline{x} \in G_\nu$ $(\nu \in \{1,2,\ldots,s\})$.

Пусть далее имеет место следующее соотношение:

$$F(\overline{x}) = \eta(G_v) \le \min_{i=\overline{1,s}}\{\eta(G_i)\}.$$

Тогда $\overline{x}$ – оптимальное решение задачи. Доказательство вытекает непосредственно из определения границы.

Оценка точности приближенного решения

Пусть $G=\bigcup_{i=\overline{1,s}} G_i$, и $\eta = \min_{i=\overline{1,s}}\{\eta(G_i)\}$.

Пусть далее известно некоторое допустимое решение $\overline{x} \in G_i$. В соответствии с определением границы должно иметь место следующее соотношение:

$$\eta \le \min_{\mathbf{x}\in\mathbf{G}} F(x) \le F(\overline{x}).$$

Если разность $\Delta = F(\overline{x}) - \eta$ невелика, т.е. не превышает некоторого порогового значения, выбранного для данной задачи, то $\overline{x}$ можно принять в качестве приближенного решения задачи, а Δ использовать для оценки точности решения.

Можно задаться и относительной погрешностью ε (%). В этом случае $\overline{x}$ принимается в качестве приближенного решения задачи, если имеет место:

$$\frac{F(\overline{x}) - \eta}{\eta} \times 100\% \le \boldsymbol{\varepsilon}.$$

2.4.2. Общая схема метода ветвей и границ

Дана задача

$$F(x) \to min,$$
$$x \in G.$$

Введем понятие рекорда, имеющее принципиально важное значение в методе ветвей и границ.

<u>Рекордом</u> $R= F(x')$ будем называть наименьшее из известных на конкретный момент времени значение, которое принимает ЦФ на некотором допустимом решении $x' \in G$.

<u>Шаг 1</u>. Заносим множество G в предварительно очищенный список задач S: $S=\{G\}$. Полагаем $R= M$, где M – достаточно большое положительное число.

<u>Шаг 2</u>. Для каждого подмножества списка вычисляем нижние границы (оценки). При этом оценки вычисляются только для тех подмножеств, для которых они не вычислялись ранее.

<u>Шаг 3</u>. Последовательно анализируются подмножества списка и вычисленные для этих подмножеств оценки в порядке неубывания оценок.

Пусть рассматривается очередное подмножество списка G^*. Если $G^*=\varnothing$, это подмножество удаляется из списка.

Если $\eta(G^*)\geq R$, то G^* также удаляется из списка (как "бесперспективное").

Если G^* – одноэлементное подмножество, или удается установить, что на некотором решении $x^*\in G^*$ имеет место $\eta(G^*)=F(x^*)<R$, фиксируется новый рекорд: $R= F(x^*)$.

В результате список "очищается" от всех бесперспективных подмножеств.

<u>Шаг 4</u>. Если список пустой[21] ($S=\varnothing$), то конец – задача не имеет решения.

<u>Шаг 5</u>. Из списка выбирается подмножество с минимальной оценкой. После выполнения процедуры на шаге 3 – это первое в списке подмножество. Обозначим его $G^{\min}$.

Если $G^{\min}$ – одноэлементное подмножество, или удается установить, что на некотором решении $X^{min}\in G^{\min}$ имеет место $\eta(G^{\min})=F(X^{\min})$, то получено оптимальное решение. Конец.

<u>Шаг 6</u>. По заранее установленному признаку подмножество G^{min} разбивается на ряд новых подмножеств. Эти подмножества заносятся в список S, после чего выполняется шаг 2.

Итак, в процессе работы по алгоритму метода ветвей и границ разбиению подлежат подмножества с минимальной оценкой.

Важно то обстоятельство, что оставшиеся в списке подмножества не отбрасываются (если, конечно, они не признаются бесперспективными). С чем это связано?

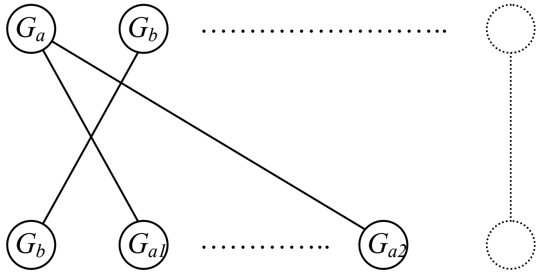
Дело в том, что нижняя граница может оказаться недостижимой: просто не существует такого решения, на котором значение ЦФ совпадает с нижней границей, а вычисление достижимой границы весьма трудно реализовать.

---

[21] Поскольку рекорд в данной редакции алгоритма хранится в списке задач, пустой список означает отсутствие решений. Существуют иные редакции алгоритма, в которых рекорд регистрируется отдельно от списка – в этом случае на этом шаге признак отсутствия допустимых решений срабатывать не будет.

Формально: $\eta(G) < \min\limits_{x \in G} F(x)$.

$\eta(G_a) < \quad \eta(G_b)$



$\eta(G_b) < \quad \eta(G_{a1}) < \ldots\ldots\ldots.. \eta(G_{a2})$

Рис.2.22. Разбиение множества

Рис. 2.22 иллюстрирует тот случай, когда после разбиения подмножества G_a, обладающего минимальной оценкой, дальнейшему разбиению должно подлежать подмножество G_b, так как после пересчета оценки вновь образованных подмножеств G_{a1} и G_{a2} оказались хуже, чем оценка G_b.

Итак, мы рассмотрели общие принципы и общую схему метода ветвей и границ. Как видно, строгого алгоритма нет. Алгоритм получится только после того, как удастся ответить на следующие вопросы:

1. Как вычислять границу $\eta(G) \le \min\limits_{x \in G} F(x)$ в задаче на минимум и $\eta(G) \ge \max\limits_{x \in G} F(x)$ в задаче на максимум ЦФ?
2. По какому принципу выполнять ветвление?

При решении этих двух вопросов обязательно нужно убедиться в том, что при $G_1 \subset G_2$ имеет место

$$\min_{x \in G_1} F(x) \ge \min_{x \in G_2} F(x).$$

3. По какому принципу устанавливать факт получения оптимального решения?

Таким образом, использование метода ветвей и границ предполагает, прежде всего, конкретизацию основных принципов этого метода.

2.4.3. Метод Лэнд и Дойг

Дана задача линейного частично-целочисленного программирования:

$$\sum_{j=1}^{n} c_j x_j \to \max \tag{2.32}$$

$$\sum_{j=1}^{n} a_{ij} x_j = a_i, \ i = \overline{1,m} \tag{2.33}$$

$$x_j \ge 0, \ j = \overline{1,n} \tag{2.34}$$

$$x_j - \textit{целые}, \ j = \overline{1,n_1}, \ n_1 \le n \ . \tag{2.35}$$

Тот факт, что (2.32)-(2.34) – задача линейного программирования, сразу наталкивает на мысль, что границу (верхнюю) удобнее всего вычислять, решая именно эту задачу. Действительно, ведь всегда имеет место

$$Z_{1,2,3}^{opt} \ge Z_{1,2,3,4}^{opt} .$$

Вопрос вычисления границы решен.

Как будем выполнять ветвление и какой признак оптимальности – станет ясно в процессе изложения алгоритма.

Рекордом будем называть допустимое решение задачи ЛЦП, доставляющее ЦФ значение, не меньшее, чем на любом известном на данный момент допустимом решении.

Шаг 1. Решим задачу ЛП (2.32)-(2.34). Если задача неразрешима, то и ЛЦП-задача (2.32)-(2.35) неразрешима. Полагаем N=1 и в предварительно очищенный список задач S заносим задачу ЛП (2.32)-(2.35), дав ей обозначение $З_1$. В этот же список заносим оптимальное решение задачи, которое принимается в качестве верхней границы η_1 .

Шаг 2. Имеем список задач S, в котором задачи упорядочены по невозрастанию оценок:

$$S = \{З_1(\eta_1), З_2(\eta_2), \dots, З_N(\eta_N)\} .$$

Если список пуст (S=∅), ЛЦП-задача не имеет решения. Конец.

В противном случае из списка выбирается задача $З_1$ – задача, имеющая максимальную оценку.

Шаг 3. Если оптимальный план выбранной задачи X_1 удовлетворяет требованию целочисленности (2.35), то X_1 – оптимальное решение исходной задачи, η_1 – оптимальное значение ЦФ. Конец.

Шаг 4. Выбираем нецелую координату решения X_1. Пусть это будет координата x_ν, имеющая в решении X_1 значение $\overline{x}_\nu$. Формируем и последовательно решаем две задачи – $З_1^1$ и $З_1^2$ (рис. 2.23):

$З_1^1$	$З_1$
	$x_\nu \geq [\,\overline{x}_\nu\,]+1$

$З_1^2$	$З_1$
	$x_\nu \leq [\,\overline{x}_\nu\,]$

Рис.2.23. Задачи, получаемые в результате ветвления

По результатам решения каждой из задач $З_1^1$ и $З_1^2$ выполняются следующие действия.

Если задача имеет решение, то она вместе со своим оптимальным решением заносится в список задач S – в позицию, соответствующую полученному значению ЦФ (оценки). Таким образом, все задачи в новом списке S' упорядочиваются по невозрастанию оценок.

Шаг 5. Проводится сквозная перенумерация задач в новом списке. Этот список приобретает вид:

$$S' = \{З_1(\eta_1), З_2(\eta_2), ..., З_l(\eta_l), ..., З_{N'}(\eta_{N'})\}$$

$$\eta_1 \geq \eta_2 \geq ... \geq \eta_l \geq ... \geq \eta_{N'} .$$

Из списка удаляются все бесперспективные задачи. Задача $З_r$ считается бесперспективной, если в списке есть задача $З_l$, имеющая целочисленное решение и $r > l$, т.е., $\eta_l \geq \eta_r$. Таким образом, срабатывает механизм рекордов. А именно, задача $З_l$, имеющая целочисленное решение (если таковая есть), в новом списке будет занимать последнюю позицию. Этой задаче будет соответствовать рекорд.

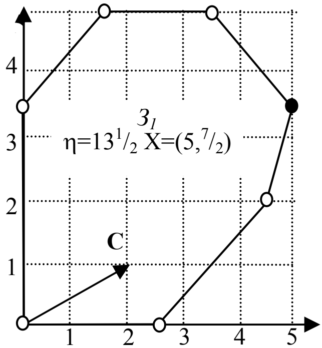
В новом, очищенном от бесперспективных задач списке будет N задач ($N \leq N'$). Принимаем $S = S'$ и переходим к шагу 2.

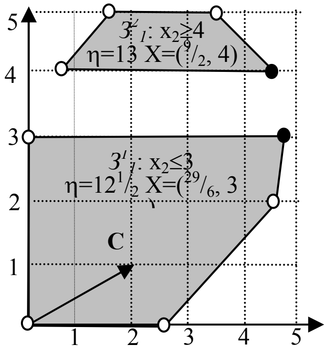
Пример 2.17

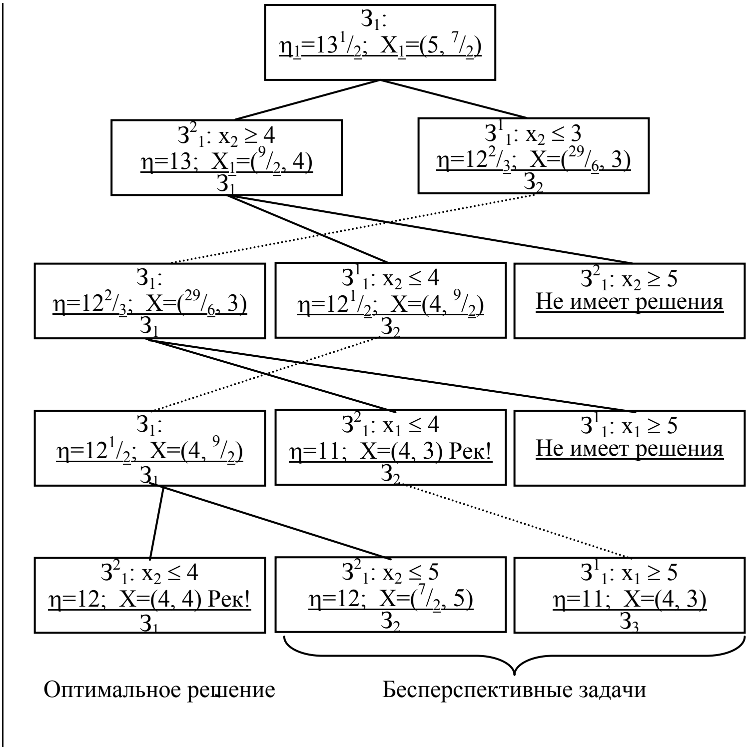
Дана задача ЛЦП с целевой функцией $Z=2x_1+x_2\to\max$. Область допустимых решений соответствующей ЛП-задачи задана графически.

Приведенная ниже диаграмма иллюстрирует начало формирования списка задач по методу Лэнд и Дойг: из исходной задачи формируются две задачи. Ветвление осуществляется по координате x_2, которая в оптимальном решении исходной задачи имеет нецелое значение ($^7/_2$) .

На следующей диаграмме представлена полная схема процесса решения задачи.







2.4.4. Использование штрафов в методе Лэнд и Дойг

Допустим, решена некоторая задача из списка задач, и получено нецелочисленное решение этой задачи.

Оптимальная симплекс-таблица имеет вид (рис. 2.24):

Баз	$C_{баз}$	A_0		c_p		c_j	
				A_p		A_j	
A_p	c_p	x_{p0}		*1*		x_{pj}	
		Z_0		*0*		$\Delta_j \ge 0$	

Рис.2.24. Оптимальная симплекс-таблица

Пусть x_{p0} – значение некоторой базисной переменной x_p в оптимальном решении, не удовлетворяющее требованию целочисленности. Для определенности примем, что $p \in \{1,2,...,m\}$; а x_j , где $j \in \{m+1,m+2,...,n\}$ – небазисные переменные.

В p-й строке последней симплекс-таблицы записано следующее уравнение:

$$x_{p0} = x_p + \sum_{j=m+1}^{n} x_{pj} x_j$$

или

$$x_p = x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j \ . \qquad (2.36)$$

Что означает ветвление по переменной x_p ?

Это означает, что необходимо сформировать две новые задачи, в систему ограничений одной из которых должно войти ограничение $x_p \geq [x_{p0}] + 1$.

В систему ограничений другой задачи должно войти ограничение $x_p \leq [x_{p0}]$ (рис. 2.25).

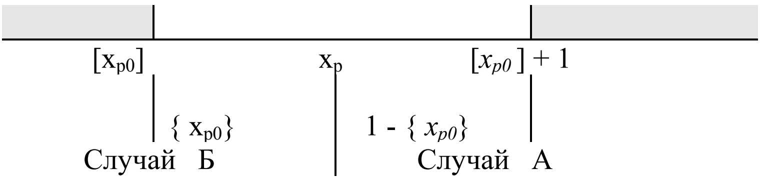


Рис.2.25. Задачи, получаемые при ветвлении

Случай А. Переменная x_p должна принять значение, не меньшее, чем $[x_{p0}] + 1$. Следовательно, эта переменная должна быть увеличена, по крайней мере, на величину:

$$1 - \{x_{p0}\}.$$

Как поведут себя свободные переменные?

$$x_p = x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j \ .$$

По крайней мере, одна из этих переменных (пусть это будет x_j, для которой $x_{pj} < 0$) должна увеличиться на некоторую величину δ, которую можно найти из уравнения (2.36):

$$x_p + (1 - \{x_{p0}\}) = x_{p0} - x_{pj}(0 + \delta).$$

Здесь "0" – старое значение x_j ; "δ" – новое значение.

Ввиду того, что $x_p = x_{p0}$, имеет место

$$1 - \{ x_{p0}\} = - x_{pj}\,\delta$$

Отсюда определяется δ:

$$\delta = \frac{1-\{x_{p0}\}}{-x_{pj}}$$

В разделе 1.2.15 было показано, что оценки свободных переменных – это взятые с обратным знаком коэффициенты, с которыми свободные переменные входят в целевую функцию, если базисные переменные выразить через свободные. То есть, справедливо следующее выражение:

$$Z = Z_0 - \sum_{j=m+1}^{n} \Delta_j x_j .$$

Пусть теперь переменная x_j увеличивается на величину $\delta = \frac{1-\{x_{p0}\}}{-x_{pj}}$. Пусть при этом остальные свободные переменные не изменяют своих значений – остаются нулевыми. Следовательно, целевая функция должна уменьшиться на величину $\Delta_j \frac{1-\{x_{p0}\}}{-x_{pj}}$ (помним, что $x_{pj} < 0$, а все $\Delta_j \geq 0$).

Теперь можно сделать вывод о том, что минимальная величина, на которую должно уменьшиться (ухудшиться) значение целевой функции в результате наложения ограничения $x_p \geq [x_{p0}] + 1$, определяется следующим образом:

$$Z_U^p = \min_{x_{pj}<0}\{\Delta_j \left(\frac{1-\{x_{p0}\}}{-x_{pj}}\right)\} .$$

Эту величину принято называть *штрафом сверху*.

Следует подчеркнуть важное обстоятельство: для определения этого штрафа задачу можно не решать.

Всегда ли можно вычислить штраф сверху? Нет, не всегда. Действительно, может оказаться, что p-я строка симплекс-таблицы не имеет ни одной отрицательной координаты $x_{pj} < 0$.

Вернемся к выражению $x_p = x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j$. Ограничение $x_p \geq [x_{p0}] + 1$ эквивалентно следующему ограничению:

$$x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j \geq [x_{p0}] + 1.$$

Отсюда следует $\sum_{j=m+1}^{n} x_{pj} x_j \leq -1 + x_{p0} - [x_{p0}]$. Но $x_{p0} - [x_{p0}] = \{x_{p0}\}$, т.е.

$$\sum_{j=m+1}^{n} x_{pj} x_j \leq -1 + \{x_{p0}\}.$$

Для того чтобы работать по симплекс-методу, это ограничение нужно привести к виду уравнения путем введения дополнительной переменной $y_p \geq 0$. Соответствующая строка симплекс-таблицы будет иметь вид:

$$-1 + \{x_{p0}\} = \sum_{j=m+1}^{n} x_{pj} x_j + y_p .$$

Левая часть этого уравнения – отрицательное число. Оценки остались неотрицательными. То есть теперь в симплекс-таблице записан псевдоплан: нужно работать по двойственному симплекс-методу. В столбце A_0 есть единственный отрицательный элемент $(-1 + \{x_{p0}\})$, но в соответствующей этому элементу строке нет ни одной отрицательной координаты.

Это – тот случай, когда в двойственном симплекс-методе срабатывает признак недопустимости задачи.

Таким образом, если штраф сверху не удается найти, соответствующую задачу можно не решать – она недопустима.

Случай Б. Переменная x_p должна принять значение, не большее, чем $[x_{p0}]$. Следовательно, эта переменная должна быть уменьшена, по крайней мере, на величину $\{x_{p0}\}$.

Как "поведут себя" свободные переменные?

$$x_p = x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j$$

Эти переменные могут только увеличиваться. По крайней мере, одна из этих переменных (пусть это будет x_j), для которой $x_{pj} > 0$, должна увеличиться на некоторую величину δ. Эту величину можно найти из уравнения (2.36):

$$x_p - \{x_{p0}\} = x_{p0} - x_{pj}(0 + \delta).$$

Здесь "0" – старое значение x_j ; "δ" – новое значение.

Ввиду того, что $x_p = x_{p0}$, имеет место:

$$-\{x_{p0}\} = -x_{pj}\,\delta.$$

Отсюда определяется δ:

$$\delta = \frac{\{x_{p0}\}}{x_{pj}}.$$

Как было показано ранее:

$$Z = Z_0 - \sum_{j=m+1}^{n} \Delta_j x_j .$$

Пусть переменная x_j увеличивается на величину $\delta = \frac{\{x_{p0}\}}{x_{pj}}$. Пусть при этом остальные свободные переменные не изменяют своих значений – остаются нулевыми. Следовательно, целевая функция должна уменьшиться на величину $\Delta_j \frac{\{x_{p0}\}}{x_{pj}}$ (помним, что $x_{pj} > 0$).

Теперь можно сделать вывод о том, что минимальная величина, на которую должно уменьшиться (ухудшиться) значение целевой функции в результате наложения ограничения $x_p \le [x_{p0}]$, определяется следующим образом:

$$Z_D^p = \min_{x_{pj}>0}\{\Delta_j \left(\frac{\{x_{p0}\}}{x_{pj}}\right)\}.$$

Эту величину принято называть *штрафом снизу*.

Так же, как и в предыдущем случае, для определения штрафа снизу задачу можно не решать.

Всегда ли можно вычислить штраф снизу? Нет, не всегда. Действительно, может оказаться, что *p*-я строка симплекс-таблицы не имеет ни одной положительной координаты $x_{pj} > 0$.

Вернемся к выражению $x_p = x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j$. Ограничение $x_p \le [x_{p0}]$ эквивалентно следующему ограничению:

$$x_{p0} - \sum_{j=m+1}^{n} x_{pj} x_j \le [x_{p0}].$$

Отсюда следует x_{p0} - $[x_{p0}] \le \sum_{j=m+1}^{n} x_{pj}x_j$. Но x_{p0} - $[x_{p0}] = \{ x_{p0}\}$, т.е.

$$\{x_{p0}\} \le \sum_{j=m+1}^{n} x_{pj}x_j .$$

Для того чтобы работать по симплекс-методу, это ограничение нужно привести к виду уравнения путем введения дополнительной переменной $y_p \ge 0$. Соответствующая строка симплекс-таблицы будет иметь вид:

$$- \{x_{p0}\} = - \sum_{j=m+1}^{n} x_{pj}x_j + y_p$$

Левая часть этого уравнения – отрицательное число. Оценки остались неотрицательными. То есть теперь в симплекс-таблице записан псевдоплан: нужно работать по двойственному симплекс-методу. В столбце A_0 есть единственный отрицательный элемент (- $\{ x_{p0}\}$), но в соответствующей этому элементу строке нет ни одной отрицательной координаты (так как все $x_{pj} \le 0$).

В двойственном симплекс-методе срабатывает признак недопустимости задачи.

Таким образом, если штраф снизу не удается найти, соответствующую задачу можно не решать – она недопустима.

Итак, мы только что сформулировали правила, по которым можно отбрасывать ветви, соответствующие задачам, не имеющим решения. Очень важно, что использование механизма штрафов позволяет вообще отказаться от решения соответствующих задач.

Следующий вопрос – это использование механизма штрафов для отсеивания бесперспективных ветвей.

Допустим, в результате решения серии задач ЛП в списке задач имеется некоторая задача, оптимальное решение которой удовлетворяет требованию целочисленности, и на этом решении целевая функция имеет значение R.

Пусть из списка выбирается некоторая задача, имеющая максимальную оценку Z^* (естественно, большую, чем R).

Пусть далее выбирается нецелая координата x_p , и по этой координате вычисляются штрафы. Если какой-либо штраф не удается вычислить, соответствующая ветвь отбрасывается, так как новая задача не имеет решения.

Пусть вычислены штрафы Z_U^p (сверху) и Z_D^p (снизу).

Тогда открывается возможность установить "бесперспективность" соответствующих новых задач:

1. $Z^* - Z_U^p \le R$ – решать задачу, которая порождается введением ограничения $x_p \ge [x_{p0}] + 1$ не имеет смысла: оценка будет заведомо плохой;
2. $Z^* - Z_D^p \le R$ – решать задачу, которая порождается введением ограничения $x_p \le [x_{p0}]$ не имеет смысла: оценка будет заведомо плохой.

Отсюда естественным образом вытекает следующее правило:

если $\max\{ Z^* - Z_U^p, Z^* - Z_D^p \} \le R$, следует вообще отказаться от ветвления по переменной x_p (соответствующая ветвь просто отбрасывается).

Ветвление имеет смысл осуществлять по той переменной, которая имеет максимальный штраф.

Штраф переменной x_p ($p=1,2,...,m$) определяется следующим образом:

$$Z^p = \max\{ Z_U^p, Z_D^p \}.$$

Таким образом, для ветвления выбирается переменная с максимальным штрафом – ускоряется процесс уточнения оценки.

После того, как выбрана переменная, формируются две задачи.

В первую очередь решается та задача, которая обладает минимальным штрафом (сверху или снизу).

Эта задача, если, конечно, она перспективна, решается и помещается в список.

Затем исследуется вторая задача (если она есть). Если она перспективна (первая решенная задача может сделать вторую бесперспективной), то решается и помещается в список задач.

В общем случае перспективность той или иной задачи проверяется два раза: до и после решения.

В заключение этого раздела о приближенном решении задач ЛЦП методом Лэнд и Дойг.

Предположим, что $V^к$ – это максимальная верхняя граница, известная к шагу k работы алгоритма.

Допустим, на этом шаге решается некоторая задача ЛП , и эта задача имеет оптимальное целочисленное решение X^k, доставляющее целевой функции значение Z^k . Тогда X^k считается достаточно близким к оптимальному решением, если имеет место:

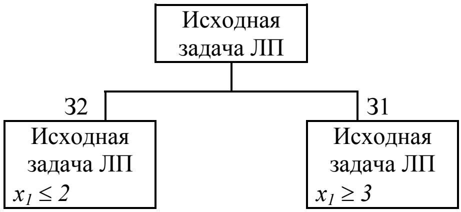
$$\frac{V^k - Z^k}{V^k} \le \varepsilon$$, где ε – заданная точность решения, например, ε = = 0.01

Пример 2.18

Дана оптимальная симплекс-таблица некоторой задачи, которая решается методом ветвей и границ:

Баз	$C_{баз}$	A_0	1	3	0	0
			A_1	A_2	A_3	A_4
A_1	1	5/2	1	0	1/2	-1/2
A_2	3	3	0	1	0	1
		23/2	0	0	1/2	5/2

Очевидно, что ветвление следует провести по переменной x_1.



Для того,чтобы определить, по какой ветви следует идти (какая из двух задач З1 и З2 должна решаться в первую очередь), найдем штрафы сверху и снизу.

З1: $Z_U^1 = \min\limits_{x_{1j}<0}\{\Delta_j \left(\dfrac{1-\{x_{10}\}}{-x_{1j}}\right)\} = \dfrac{5}{2}\left(\dfrac{1-\frac{1}{2}}{\frac{1}{2}}\right) = \dfrac{5}{2}$ – штраф сверху;

З2: $Z_D^1 = \min\limits_{x_{1j}>0}\{\Delta_j \left(\dfrac{\{x_{10}\}}{x_{1j}}\right)\} = \dfrac{1}{2}\left(\dfrac{\frac{1}{2}}{\frac{1}{2}}\right) = \dfrac{1}{2}$ – штраф снизу.

Ввиду того, что штраф снизу меньше, чем сверху, в первую очередь следует решать задачу З2, которая получается из исходной задачи добавлением ограничения :

$x_1 \le [x_{10}] = [5/2] = 2$.

Как технически осуществить решение задачи З2 ? Ведь исходной задачи нет. Есть только оптимальная симплекс-таблица, в которой записано решение исходной задачи.

Здесь используется следующий прием.

В первой строке симплекс-таблицы записано уравнение: $x_1 = x_{10} - \sum_{j=3}^{4} x_{1j} x_j$ или $x_1 = 5/2 - (1/2)x_3 + (1/2)x_4$. Учитывая, что дополнительное ограничение имеет вид $x_1 \le 2$, его можно записать следующим образом: $x_1 = 5/2 - (1/2)x_3 + (1/2)x_4 \le 2$.

Отсюда: $- (1/2)x_3 + (1/2)x_4 \le -1/2$.

Придадим этому ограничению ─ нестрогому неравенству форму ограничения-уравнения путем введения в правую часть дополнительной (неотрицательной) переменной x_5 :

$$(1/2)x_3 + (1/2)x_4 + x_5 = -1/2 .$$

Теперь можно воспользоваться оптимальной симплекс-таблицей исходной задачи. Добавив в эту таблицу новую строку, соответствующую этому ограничению, и один столбец, можно получить исходную симплекс-таблицу для решения задачи З2:

Баз	$C_{баз}$	A_0	1	3	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_1	1	5/2	1	0	1/2	-1/2	0
A_2	3	3	0	1	0	1	0
A_5	0	-1/2 1	0 0	0 0	-1/2 1	1/2 -1	1 -2
		23/2	0	0	1/2	5/2	0
A_1	1	2	Эту часть можно не считать				
A_2	3	3					
A_3	0	1					
		11					

Получено оптимальное решение задачи З2 . Заносим эту задачу в список.

Начинается решение задачи З1, которая получается из исходной задачи путем введения дополнительного ограничения $x_3 \ge 3$.

Штраф сверху известен:

$$Z_U^1 = \min_{x_{1j}<0} \{\Delta_j \left(\frac{1-\{x_{10}\}}{-x_{1j}} \right)\} = \frac{5}{2} \left(\frac{1-\frac{1}{2}}{\frac{1}{2}} \right) = \frac{5}{2}$$

То есть самое большое значение ЦФ, которое может быть получено, это 23/2 - 5/2=9.

Но в списке уже есть задача с целочисленным решением и лучшим (большим) оптимальным значением ЦФ (задача З2). Следовательно, задача З1 бесперспективна: ее не следует решать и заносить в список задач.

Теперь в списке есть единственная задача с целочисленным решением (задача З2). Следовательно, получено оптимальное решение исходной задачи:

$Z_{опт} = 11$, $X_{опт} = (2,3)$.

2.4.5. Метод ветвей и границ для задачи о коммивояжере

Пусть $A_1, A_2, ..., A_n$, – множество городов.

$C=(c_{ij})$ – квадратная матрица "расстояний" между городами прядка n.

Пусть коммивояжер выезжает из города A_1 и возвращается в этот город, побывав в каждом из остальных городов по одному разу.

Пусть далее Ω – множество всех возможных маршрутов. Обозначим $\Omega_{1,i_2,i_3,...,i_k}$ $(2 \le k \le n\text{-}1)$ – подмножество всех допустимых маршрутов, каждый из которых характеризуется тем, что выезжая из города A_1, коммивояжер последовательно посещает города $A_{i_2}, A_{i_3}, ..., A_{i_k}$, а затем – оставшиеся города и возвращается в город A_1.

Конкретизируем принцип ветвления. Пусть коммивояжер находится в пункте A_1 – в исходном пункте. Выбором первого города, в который нужно въехать из A_1, множество всех возможных маршрутов Ω разбивается на непересекающиеся подмножества $\Omega_{1,2}$, $\Omega_{1,3}$,... $\Omega_{1,n}$ (рис. 2.26):

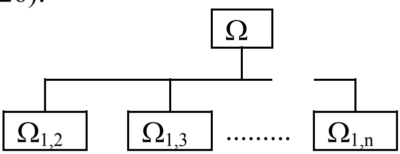


Рис.2.26. Ветвление в задаче коммивояжера

Пусть теперь после посещения городов $A_{i_2}, A_{i_3}, ..., A_{i_k}$ $(2 \le k \le n\text{-}$ $\text{-}1)$ принимается решение о выборе очередного города, в который нужно въехать, выехав из города A_{i_k}.

Фактически, этим решением множество допустимых маршрутов $\Omega_{1,i_2,i_3,...,i_k}$ разбивается на подмножества:

$\Omega_{1,i_2,i_3,\dots,i_k,j}$, где $j \notin \{1, i_2, i_3, \dots, i_k\}$.

Очевидно, что подмножество $\Omega_{1,i_2,i_3,\dots,i_{n-1}}$ состоит из одного-единственного маршрута:

$A_1, A_{i_2}, A_{i_3}, \dots, A_{i_{n-1}}, A_{i_n}, A_1$, где $i_n \notin \{1, i_2, i_3, \dots, i_{n-1}\}$.

Выбора нет.

Конкретизируем принцип вычисления нижней границы (оценки). Оценку подмножества $\Omega_{1,i_2,i_3,\dots,i_{k-1},i_k}$ $(2 \le k \le n-1)$ будем вычислять следующим образом:

$$\eta(\Omega_{1,i_2,i_3,\dots,i_{k-1},i_k}) = c_{1,i_2} + c_{i_2,i_3} + \dots + c_{i_{k-1},i_k} + \min_{j \notin \{1,i_2,\dots,i_k\}} \{c_{i_k,j}\} +$$

$$+(n-k) \min_{\substack{i \notin \{1,i_2,\dots,i_k\} \\ l \notin \{i_2,\dots,i_k\} \\ i \ne l}} \{c_{i,l}\}.$$

Здесь

$c_{1,i_2} + c_{i_2,i_3} + \dots + c_{i_{k-1},i_k}$ – фактическая стоимость всех уже состоявшихся переездов;

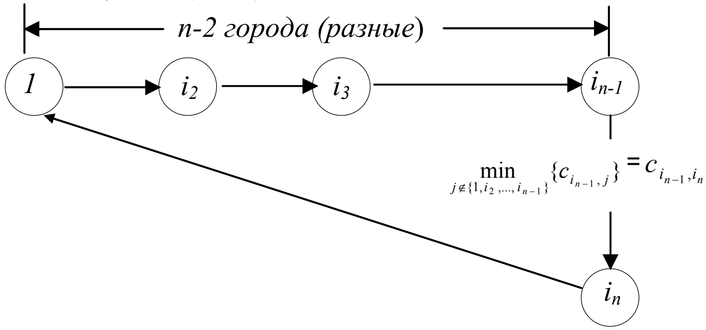
$\min\limits_{j \notin \{1,i_2,\dots,i_k\}} \{c_{i_k,j}\}$ – стоимость самого короткого переезда из пункта i_k в пункт j, в котором еще не был коммивояжер;

$(n-k)$ – количество оставшихся переездов;

$\min\limits_{\substack{i \notin \{1,i_2,\dots,i_k\} \\ l \notin \{i_2,\dots,i_k\}}} \{c_{i,l}\}$ – стоимость самого короткого из оставшихся переездов.

Очевидно, что при $k=n-1$ оценка $\eta(\Omega_{1,i_2,i_3,\dots,i_k})$ совпадает со стоимостью единственного маршрута, составляющего множество $\Omega_{1,i_2,i_3,\dots,i_k}$ – это стоимость маршрута (рис. 2.27):

$$A_1, A_{i_2}, A_{i_3}, \ldots, A_{i_{n-1}}, A_{i_n}, A_1, \quad \text{где} \quad i_n \notin \{1, i_2, i_3, \ldots, i_{n-1}\}.$$



Последний переезд:

$$[n\text{-}(n\text{-}1)] \times \min_{\substack{i \notin \{1, i_2, \ldots, i_{n-1}\} \text{ или } i = i_n \\ l \notin \{i_2, \ldots, i_{n-1}\} \text{ или } l = 1}} \{c_{i,l}\} = 1 \times$$

Рис.2.27. Маршрут

Как видно, предложенная оценка весьма оптимистическая. Однако, по мере продвижения (по мере разбиения множества допустимых маршрутов), эта оценка уточняется – убывать, в принципе, не может.

Таким образом, конкретизирован еще один принцип метода ветвей и границ – принцип уточнения оценок.

Теперь о последнем принципе – о принципе оптимальности.

Здесь вполне естественна следующая схема.

Имеется список подмножеств допустимых маршрутов.

Каждое из этих подмножеств имеет свою оценку.

Пусть на очередном шаге работы алгоритма из списка выбирается подмножество с минимальной оценкой.

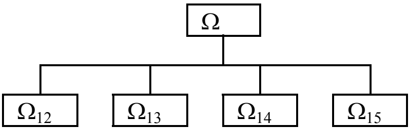
Если этому подмножеству соответствует полный маршрут (т.е. k=n-1) , то этот маршрут является оптимальным.

Пример 2.19

В задаче нужно объехать 5 городов, выехав из города 1. Стоимости переездов представлены в таблице.

	4	9	6	1
2		9	2	10
11	11		8	1
5	4	3		8
1	11	1	8	

Разбиваем полное множество всех маршрутов Ω на 4 подмножества:



Находим оценку (нижнюю границу) каждого из этих подмножеств.

Подмножество Ω_{12}:

	4	9	6	1
2		9	2	10
11	11		8	1
5	4	3		8
1	11	1	8	

$$\eta(\Omega_{12})=c_{12}+\min_{j\neq 1,2}\{c_{2,j}\}+(5\text{-}2)\min_{\substack{i\neq 1,2\\ l\neq 2}}\{c_{i,l}\}=4+2+3\times 1=9.$$

Подмножество Ω_{13}:

	4	9	6	1
2		9	2	10
11	11		8	1
5	4	3		8
1	11	1	8	

$$\eta(\Omega_{13})=c_{13}+\min_{j\neq 1,3}\{c_{3,j}\}+(5\text{-}2)\min_{\substack{i\neq 1,3\\ l\neq 3}}\{c_{i,l}\}=9+1+3\times 1=13.$$

Подмножество Ω_{14}:

	4	9	$\boxed{6}$	1
2		9	2	10
11	11		8	1
5	4	3		8
1	11	1	8	

$$\eta(\Omega_{14})=c_{14}+\min_{j\neq 1,4}\{c_{4,j}\}+(5\text{-}2)\min_{\substack{i\neq 1,4\\ l\neq 4}}\{c_{i,l}\}=6+3+3\times 1=12.$$

Подмножество Ω_{15}:

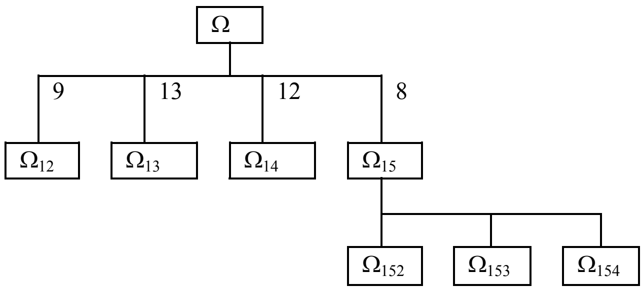
	4	9	6	$\boxed{1}$
2		9	2	10
11	11		8	1
5	4	3		8
1	11	1	8	

$$\eta(\Omega_{15})=c_{15}+\min_{j\neq 1,5}\{c_{5,j}\}+(5\text{-}2)\min_{\substack{i\neq 1,3\\ l\neq 3}}\{c_{i,l}\}=1+1+3\times 2=8.$$

Заносим подмножества (вместе с оценками) в список.

Из списка выбирается подмножество с минимальной оценкой (это Ω_{15} с оценкой 8).

Подмножество Ω_{15} разбивается на 3 подмножества:



Вычисляются нижние границы вновь образованных подмножеств.

Подмножество Ω_{152} :

$$\eta(\Omega_{152})=c_{15}+c_{52}+\min_{j\neq 1,5,2}\{c_{2,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,5,2\\ l\neq 5,2}}\{c_{i,l}\}=1+11+2+2\times 3=20.$$

Подмножество Ω_{153}:

$$\eta(\Omega_{153})=c_{15}+c_{53}+\min_{j\neq 1,5,3}\{c_{3,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,5,3\\ l\neq 5,3}}\{c_{i,l}\}=1+1+8+2\times 2=14.$$

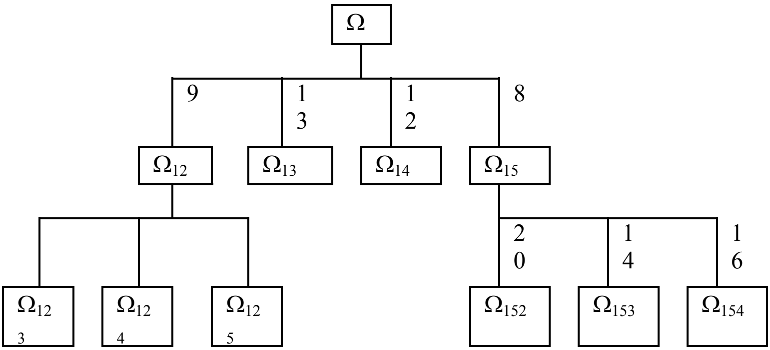
Подмножество Ω_{154}:

$$\eta(\Omega_{154})=c_{15}+c_{54}+\min_{j\neq 1,5,4}\{c_{4,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,5,4\\ l\neq 5,4}}\{c_{i,l}\}=1+8+3+2\times 2=16.$$

Заносим подмножества (вместе с оценками) в список.

Из списка выбирается подмножество с минимальной оценкой (это Ω_{12} с оценкой 9).

Подмножество Ω_{15} разбивается на 3 подмножества:



Вычисляются нижние границы вновь образованных подмножеств.

Подмножество Ω_{123}:

$$\eta(\Omega_{123})=c_{12}+c_{23}+\min_{j\neq 1,2,3}\{c_{3,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,2,3\\ l\neq 2,3}}\{c_{i,l}\}=4+9+1+2\times 1=16.$$

Подмножество Ω_{124}:

$$\eta(\Omega_{124})=c_{12}+c_{24}+\min_{j\neq 1,2,4}\{c_{4,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,2,4\\ l\neq 2,4}}\{c_{i,l}\}=4+2+3+2\times 1=11.$$

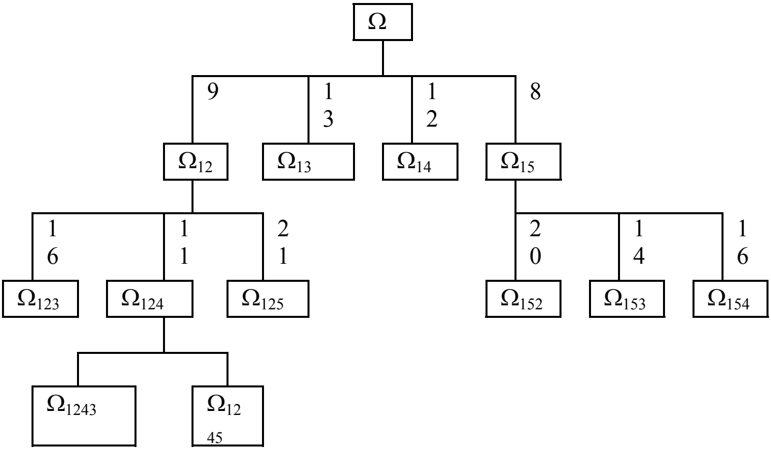
Подмножество Ω_{125}:

$$\eta(\Omega_{125})=c_{12}+c_{25}+\min_{j\neq 1,2,5}\{c_{5,j}\}+(5\text{-}3)\min_{\substack{i\neq 1,2,5\\ l\neq 2,5}}\{c_{i,l}\}=21.$$

Заносим подмножества (вместе с оценками) в список.

Из списка выбирается подмножество с минимальной оценкой (это Ω_{124} с оценкой 11).

Подмножество Ω_{124} разбивается на 2 подмножества:



Вычисляются нижние границы вновь образованных подмножеств.

Подмножество Ω_{1243} :

$$\eta(\Omega_{1243})=c_{12}+c_{24}+c_{43}+\min_{j\neq 1,2,4,3}\{c_{3,j}\}+(5\text{-}4)+\min_{\substack{i\neq 1,2,4,3\\ l\neq 2,3,4}}\{c_{i,l}\}=$$

$=4+2+3+1+1=11.$

Это подмножество соответствует полному маршруту:

$1\to 2\to 4\to 3\to 5\to 1.$

При этом оценка этого маршрута – минимальная.

Действительно, $\eta(\Omega_{1245})$ можно не вычислять, так как исходная оценка $\eta(\Omega_{124})=11$ при пересчете может только ухудшиться.

Для справки: $\eta(\Omega_{1245})=26$.

Контрольные вопросы и задачи к разделу 2.4

1. Приведите основные принципы метода ветвей и границ.

2. Что такое бесперспективное подмножество решений? В чем заключается критерий определения перспективности подмножества решений?

3. Дайте определение рекорда в методе Ленд и Дойг.

4. Какие преимущества даёт использование штрафов в вычислительной схеме метода Ленд и Дойг?

5. Напишите формулу вычисления оценки нижней границы в методе ветвей и границ для решения задачи коммивояжера.

6. Решите задачу ЛЦП методом Ленд и Дойг (графически):

$$x_1 + 2x_2 \to \max$$
$$x_1 + 3x_2 \le 15$$
$$7x_1 + x_2 \le 35$$
$$x_1, x_2 \ge 0, \textit{целые}.$$

7. В процессе решения задачи ЛЦП методом Ленд и Дойг была получена оптимальная симплекс-таблица частной задачи:

			1	1	1	2
Баз	$C_{баз}$	A_0	A_1	A_2	A_3	A_4
A_3	1	5.5	1	0.5	1	0
A_4	2	7	3	7	0	1
Табл.N		19.5	6	13.5	0	0

Проведите ветвление частной задачи с вычислением штрафов.

8. Решите задачу коммивояжера методом ветвей и границ.

Город	**1**	**2**	**3**	**4**
1	-	3	1	5
2	2	-	2	1
3	3	5	-	2
4	2	2	3	-

9. Решите задачу коммивояжера методом ветвей и границ.

Город	**1**	**2**	**3**	**4**	**5**
1	-	2	1	2	1
2	3	-	4	1	3
3	4	2	-	1	5
4	2	5	3	-	1
5	4	3	1	2	-

2.5. Приближенные методы решения дискретных задач

2.5.1. Обзор приближенных методов

На примере метода отсечения мы столкнулись с теми принципиальными трудностями, которые отличают задачи ЛП от задач ЛЦП.

В ряде случаев для получения решения задачи ЛЦП можно идти по пути округления оптимального решения задачи ЛП.

Например, если речь идет о планировании выпуска достаточно крупных партий изделий. Здесь округление вполне правомерно, тем более, что исходная информация, как правило, задается неточно, с определенными "допусками".

Однако в очень большом количестве случаев объект управления не описывается моделью ЛП – требование целочисленности параметров управления является принципиальным.

Например, задача о коммивояжере. Нецелочисленное ее решение не имеет никакого смысла.

Первый вопрос, на который следует ответить, ставится так – решать ли задачу точно или приближенно. Соответственно, существующие методы можно условно разбить на две группы: точные и приближенные.

В настоящее время интенсивно развиваются и используются на практике приближенные методы, которые дают возможность за приемлемое время получать "достаточно хорошее" решение. Это особенно важно в задачах оперативного планирования и управления.

Традиционные приближенные методы можно разделить на два класса:

1 приближенные методы, порожденные известными точными методами;

2 методы, с самого начала ориентированные на приближенное решение задачи.

Относительно первой группы следует отметить, что любой точный метод решения ЛЦП задачи, монотонно увеличивающий (в задачах на максимум ЦФ) значение целевой функции, может быть прерван на произвольной итерации и может рассматриваться, как приближенный метод.

В этой связи, следует еще раз отметить, что на основе метода отсечения едва ли удастся построить удовлетворительный приближенный метод решения целочисленной задачи. Гораздо более перспективной основой для конструирования приближенных алгоритмов является рассмотренный в разделе 2.4 метод ветвей и границ.

Рассмотрим один из методов, с самого начала ориентированный на приближенное решение задачи ЛЦП.

2.5.2. Приближенный метод решения задачи целочисленного программирования

Дана задача ЛЦП:

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

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i,\ (i=\overline{1,m}),$$

$$x_j \ge 0,\ x_j - \textit{целые}\ (j=\overline{1,n}).$$

Здесь все $a_{ij} \ge 0$, $c_j \ge 0$, $b_i \ge 0$.

Метод [8] представляет собой пошаговый процесс движения к искомому решению из некоторой начальной точки, в качестве которой принимается начало координат $\boldsymbol{X^0}=(0, 0, \dots ,0)$. Эта точка удовлетворяет всем ограничениям задачи.

На каждом шаге этого процесса увеличивается значение только одной переменной на единицу. Значения остальных переменных остаются неизменными. Процесс завершается в тот момент, когда ни одну из координат полученной точки (решения) нельзя увеличить: новое решение становится недопустимым.

<u>Шаг 0</u>. Принимаем $k = 0$, $X^k=(x_1^k, x_2^k, \dots, x_n^k)=(0, 0, \dots ,0)$. $b_i^k = b_i\ (i=\overline{1,m})$.

<u>Шаг 1</u>. Решаемая задача ЛЦП$_k$ имеет вид:

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

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i^k,\ (i=\overline{1,m}),$$

$$x_j \ge 0,\ x_j - \textit{целые}\ (j=\overline{1,n}).$$

Определяем верхние границы переменных:

$d_j^{k+1} = \min\limits_i \left\{ \dfrac{b_i^k}{a_{ij}} \right\},\ j=\overline{1,n}$. (На сколько можно увеличить значение переменной x_j, чтобы не нарушить ни одного ограничения задачи ЛЦП$_k$.)

Шаг 2. Если все $d_j^{k+1} < 1 \ (j = \overline{1,n})$, то процесс завершен – получено приближенное решение задачи: $X^* = X^k = (x_1^k, x_2^k, ..., x_n^k)$, $Z^* = \sum_{j=1}^{n} c_j x_j^k$.

Конец.

Шаг 3. Для переменных x_j , имеющих границу $P_j^{k+1} \ge 1$, вычисляются *приоритеты*:

$P_j^{k+1} = c_j d_j^{k+1} \ (j = \overline{1,n}\,;\ d_j^{k+1} \ge 1)$ – "вклад" в ЦФ, который обеспечивается увеличением переменной x_j до максимально возможного значения. Выбирается переменная, обладающая максимальным приоритетом: $P_\nu^{k+1} = \max_j \{P_j^{k+1}\}$.

Шаг 4. В новом решении $\boldsymbol{X^{k+1}} = (x_1^{k+1}, x_2^{k+1}, ..., x_n^{k+1})$ переменную x_ν увеличиваем на единицу, остальные переменные остаются прежними:

$$x_\nu^{k+1} = x_\nu^k + 1,\ \ x_j^{k+1} = x_j^k \ (j \ne \nu,\ j = \overline{1,n}) .$$

Пересчитываем новые значения свободных членов в ограничениях задачи ЛЦП$_{k+1}$:

$$b_i^{k+1} = b_i^k - a_{i\nu} \ (i = \overline{1,m}) .$$

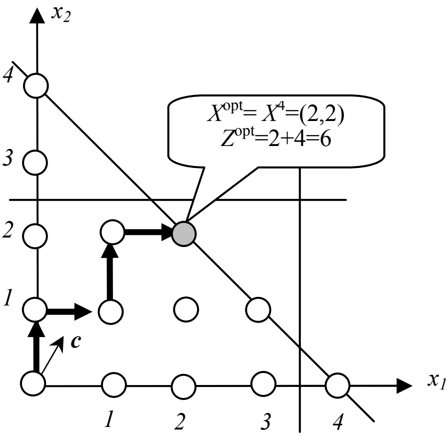
Полагаем $k=k+1$ и переходим к шагу 1.

Пример 2.20

$x_1 + 2x_2 \to \max$

$2x_2 \le 5$	3	3	1	1
$x_1 + x_2 \le 4$	3	2	1	0
$2x_1 \le 7$	7	5	5	3

$x_{1,2} \ge 0$, целые.



Шаг 0. $x_1^0 = 0,\ x_2^0 = 0;\ \ b_1^0 = 5,\ b_2^0 = 4,\ b_3^0 = 7$.

Шаг 1. $d_1^1 = \min\limits_i \left\{ \dfrac{b_i^0}{a_{i1}} \right\} = \min\{4:1,\, 7:2\} = \dfrac{7}{2};\; P_1^1 = \dfrac{7}{2} \times 1 = \dfrac{7}{2}$

$d_2^1 = \min\{5:2,\, 4:1\} = \dfrac{5}{2};\; P_2^1 = \dfrac{5}{2} \times 2 = 5 \;\Rightarrow x_2 \uparrow$

$x_1^1 = x_1^0 = 0,\; x_2^1 = x_2^0 + 1 = 0 + 1 = 1;\;\; X^1 = (0, 1)$

$b_1^1 = b_1^0 - 2 = 5 - 2 = 3,\; b_2^1 = b_2^0 - 1 = 4 - 1 = 3,\; b_3^1 = b_3^0 = 7$.

Шаг 2. $d_1^2 = \min\{3:1,\, 7:2\} = 3;\; P_1^2 = 3 \times 1 = 3 \Rightarrow$произвольно $x_1 \uparrow$

$d_2^1 = \min\{3:2,\, 3:1\} = \dfrac{3}{2};\; P_2^2 = \dfrac{3}{2} \times 2 = 3$

$x_1^2 = x_1^1 + 1 = 0 + 1 = 1,\; x_2^2 = x_2^1 = 1;\;\; X^2 = (1, 1)$

$b_1^2 = b_1^1 = 3,\; b_2^2 = b_2^1 - 1 = 3 - 1 = 2,\; b_3^2 = b_3^1 - 2 = 7 - 2 = 5$.

Шаг 3. $d_1^3 = \min\{2:1,\, 5:2\} = 2;\; P_1^3 = 2 \times 1 = 2$

$d_2^3 = \min\{3:2,\, 2:1\} = \dfrac{3}{2};\; P_2^3 = \dfrac{3}{2} \times 2 = 3 \;\;\Rightarrow x_2 \uparrow$

$x_1^3 = x_1^2 = 1,\; x_2^3 = x_2^2 + 1 = 1 + 1 = 2;\;\; X^3 = (1, 2)$

$b_1^2 = b_1^1 = 3,\; b_2^2 = b_2^1 - 1 = 3 - 1 = 2,\; b_3^2 = b_3^1 - 2 = 7 - 2 = 5$.

Шаг 4. $d_1^4 = \min\{1:1,\, 5:2\} = 1;\; P_1^4 = 1 \times 1 = 1 \;\Rightarrow x_1 \uparrow$

$d_2^4 = \min\{1:2,\, 1:1\} = \dfrac{1}{2} \;\; < 1$

$x_1^4 = x_1^3 + 1 = 1 + 1 = 2,\; x_2^4 = x_2^3 = 2;\;\; X^4 = (2, 2)$

$b_1^2 = b_1^1 = 3,\; b_2^2 = b_2^1 - 1 = 3 - 1 = 2,\; b_3^2 = b_3^1 - 2 = 7 - 2 = 5$.

Шаг 5. $d_1^5 = \min\{0,\, 3:2\} = 0 \;< 1$

$d_2^5 = \min\{1:2,\, 0\} = 0 \;\; < 1$. Конец.

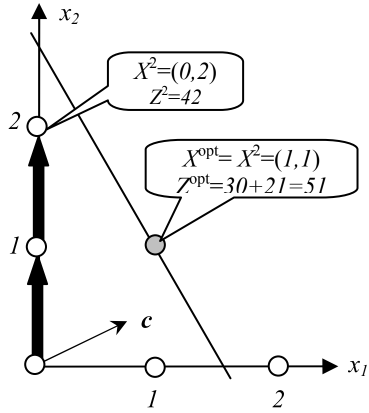
$X^{opt} = X^4 = (2,2)$

$Z^{opt} = 2 + 4 = 6$

В примере 2.20 полученное решение совпадает с оптимальным решением задачи. Однако можно привести простой пример, когда алгоритм не дает оптимального решения.

Пример 2.21

$$30x_1+21x_2 \to \max$$
$$3x_1+\ 2x_2 \le 5$$
$x_{1,2} \ge 0$, целые.



2.5.3. Метод локальной оптимизации

Как было показано в п. 2.1.4, количество комбинаций в экстремальных комбинаторных задачах растет экспоненциально при росте их размерности, что при больших размерностях делает весьма проблематичным использование точных методов, таких, как метод ветвей и границ. Вместе с тем, несмотря на то, что количество допустимых комбинаций в экстремальных задачах комбинаторного типа очень большое, оно все-таки конечное. Именно это обстоятельство используется в комбинаторных методах, основная идея которых – замена полного перебора всех допустимых вариантов (комбинаций) частичным перебором. Таким методом, в частности, является метод локальной оптимизации, в котором осуществляется последовательный переход от одной комбинации к другой, доставляющей ЦФ лучшее значение.

Итак, общая постановка задачи заключается в следующем.

Дано множество комбинаций $G=\{ x_i \}, i=1,2,\ldots,N.$

Определена функция $f(x_i)$ – целевая функция.

Требуется найти комбинацию, $x_s \in G$ такую, что

$$f(x_s) = \max_{r_i}\{f(x_i) \mid x_i \in G\}.$$

Как отмечалось, метод локальной оптимизации последовательно, начиная с некоторой начальной комбинации $x_0 \in G$, переходит к комбинациям $x_1, x_2,$ и т.д. При этом $f(x_1) < f(x_2) < ...$. Этот процесс продолжается до тех пор, пока удается найти комбинацию, доставляющую ЦФ значение большее, чем на последней найденной комбинации.

Принципиально важным понятием для организации этого процесса является понятие окрестности комбинации.

Для каждой комбинации $x_i \in G$ определим некоторое подмножество комбинаций $M(x_i) \subseteq G$, причем $x_i \in M(x_i)$.

Подмножество $M(x_i)$ назовем *окрестностью комбинации* x_i .

Комбинацию $x_j \in M(x_i) \setminus x_i$ назовем *соседней* с x_i .

Иногда (для наглядности) окрестности комбинаций задают в виде графа соседства, вершинам которого соответствуют комбинации. Дуга, направленная из вершины x_i в вершину x_j , указывает на то, что комбинация $x_j \in M(x_i) \setminus x_i$ является соседней с x_i (рис. 2.28).

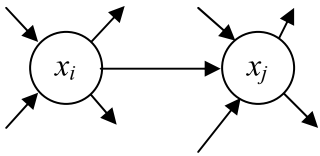


Рис.2.28. Соседние вершины

Комбинацию x_j назовем *локально-оптимальной*, если:

$$f(x_i) = \max_{x_j}\{f(x_j) \mid x_j \in M(x_i)\} .$$

Алгоритм

Шаг 0. Полагаем $k=0$. Выбираем в качестве начальной некоторую комбинацию из G. Для этой комбинации принимаем обозначение x_k .

Шаг 1. Строим окрестность комбинации x_k $M(x_k)$.

Шаг 2. Если $\max_{x_j}\{f(x_j) \mid x_j \in M(x_k)\} = f(x_k)$, то конец: x_k – локально-оптимальное решение. В противном случае – следующий шаг.

Шаг 3. Если $\max_{x_j}\{f(x_j) \mid x_j \in M(x_k)\} = f(x_\lambda) > f(x_k)$, то $k=k+1$; $x_k = x_\lambda$. Переходим к шагу 1.

Вполне очевидно, что за конечное число шагов будет найдена локально-оптимальная комбинация, однако эта комбинация может и не обеспечить глобального максимума ЦФ. Т.е., в принципе, процесс нужно продолжить. Здесь возможны различные варианты:

- в качестве начальной взять другую комбинацию и повторить работу по алгоритму;
- переопределить окрестности комбинаций (изменить граф соседства).

Ясно, что качество приближенного метода, каким является метод локальной оптимизации, существенно зависит от того, как определена окрестность. Так, если окрестность чрезмерно велика, перебор всех комбинаций такой окрестности может оказаться затруднительным. Если же окрестность слишком мала, процесс оказывается малоэффективным.

Вопрос определения окрестности существенно зависит от специфики задачи, от ее специальных свойств.

Так для известной задачи о коммивояжере разработано много алгоритмов, которые различаются определением окрестности. Рассмотрим один из таких алгоритмов.

Условия задачи приведены в таблице [8,9]. Предполагается, что исходным городом является город 1 – из него нужно выехать и в него же и вернуться, побывав в каждом из остальных городов только один раз.

Таким образом, любая перестановка чисел 2, 3, 4, 5 однозначно определяет некоторый маршрут.

Для определения понятия окрестности введем понятие соседних маршрутов (перестановок).

Две перестановки будем называть соседними, если одну из них можно получить перестановкой двух соседних элементов. При этом первый и последний элементы будем также считать соседними.

Пример 2.22

Возьмем, в качестве исходной комбинации комбинацию x_0 = =(2,3,4,5). Этой комбинации соответствует маршрут: 1→2→3→4→5→1. Стоимость маршрута:

$$f(x_0) = \underset{1\to2}{10} + \underset{2\to3}{13} + \underset{3\to4}{8} + \underset{4\to5}{13} + \underset{5\to1}{6} = \mathbf{50}$$

№	1	2	3	4	5
1		10	6	1	8
2	7		13	1	2
3	1	19		8	27
4	2	3	4		13
5	6	1	23	1	

$x_1 = (3,2,4,5);\quad f(x_1) = 45$

$$M(x_0) = \begin{cases} 3\,2\,4\,5: & f = 6+19+1+13+6 = 45 \\ 2\,4\,3\,5: & f = 10+1+4+27+6 = 48 \\ 2\,3\,5\,4: & f = 10+13+27+1+2 = 53 \\ 5\,3\,4\,2: & f = 8+23+8+3+7 = 49 \end{cases}$$

$$M(x_1) = \begin{cases} 2\,3\,4\,5: & f = 10+13+8+13+6 = 50 \\ 3\,4\,2\,5: & f = 6+8+3+2+6 = 25 \\ 3\,2\,5\,4: & f = 6+19+2+1+2 = 30 \\ 5\,2\,4\,3: & f = 8+1+1+4+1 = 15 \end{cases}$$

$x_2 = (5,2,4,3);\ f\left(x_2\right) = 15$

$$M(x_2) = \begin{cases} 2\,5\,4\,3: & f = 10+2+1+4+1 = 18 \\ 5\,4\,2\,3: & f = 8+1+3+13+1 = 26 \\ 5\,2\,3\,4: & f = 8+1+13+8+2 = 32 \\ 3\,2\,4\,5: & f = 6+19+1+13+6 = 45 \end{cases}$$

Лучшей оказалась комбинация x_2=(5,2,4,3), образующая окрестность. $f^{opt} = 15$.

Разработано очень большое количество алгоритмов решения задачи о коммивояжере, основанных на различных понятиях окрестности.

В частности, представляют определенный интерес алгоритмы, в которых при переходе от одной комбинации к другой понятие окрестности переопределяется.

Примером подобного алгоритма может служить следующий. Случайным образом генерируется некоторая комбинация $(i_1, i_2, ..., i_n)$. Окрестность этой комбинации образуется перестановками тройки i_1, i_2, i_3. В построенной таким образом окрестности выбирается лучшая комбинация. Ее окрестность образуется перестановками тройки i_2, i_3, i_4. Для следующей комбинации – i_3, i_4, i_5 и т.д.

Контрольные вопросы и задачи к разделу 2.5

1. Сравните два приближенных алгоритма решения задачи линейного целочисленного программирования: алгоритм, основанный на методе ветвей и границ, и алгоритм пошагового движения в сторону улучшения целевой функции.
2. Какой вид должна иметь задача, чтобы её можно было отнести к группе комбинаторных задач экстремального типа?
3. Можно ли любую задачу экстремального типа приближенно решить с помощью метода ветвей и границ?
4. Дайте определение окрестности комбинации в методе локальной оптимизации.
5. Почему метод локальной оптимизации является приближенным методом решения оптимизационных задач и не гарантирует получения абсолютного оптимального значения на допустимом множестве?
6. Решите задачу ЛЦП приближенным методом :

$$2x_1 + x_2 \to \max$$
$$10x_1 + 6x_2 \le 23$$
$$14x_1 + x_2 \le 21$$
$$x_1, x_2 \ge 0, \textit{целые}.$$

7. Решите задачу ЛЦП приближенным методом и графически. Сравните полученные результаты:

$$70x_1 + 41x_2 \to \max$$
$$7x_1 + 4x_2 \le 11$$
$$x_1 \le 1$$
$$x_2 \le 5$$
$$x_1, x_2 \ge 0, \textit{целые.}$$

8. Решите задачу коммивояжера методом локальной оптимизации:

Город	**1**	**2**	**3**	**4**	**5**
1	-	5	7	5	8
2	2	-	9	3	5
3	7	7	-	2	3
4	4	5	4	-	8
5	3	5	2	4	-

9. Решите задачу коммивояжера методом локальной оптимизации:

Город	**1**	**2**	**3**	**4**
1	-	2	3	1
2	4	-	1	2
3	1	1	-	1
4	3	2	3	-

2.6. Булево программирование

Основная задача ЛЦП(б) имеет форму [8,9]:

$$\sum_{j=1}^{n} c_j x_j \to \max \quad (c_j \ge 0, j=1,2,...,n) \tag{2.37}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i \quad (i=1,2,...,m) \tag{2.38}$$

$$x_j \in \{0,1\} \quad (j=1,2,...,n). \tag{2.39}$$

Любую задачу ЛЦП(б) можно свести к форме основной задачи, используя простейшие преобразования.

Равенство в ограничениях заменяется двумя нестрогими неравенствами.

Неравенство смысла "≥" умножением на "-1" левой и правой части преобразуется в нестрогое неравенство смысла "≤".

Неотрицательность коэффициента c_j обеспечивается заменой переменных:

$$y_j=1 - x_j \text{, если } c_j < 0.$$

2.6.1. Дерево ветвления в задачах булева программирования

Для обоснования различных методов решения задач ЛЦП(б) большое методологическое значение имеет представление множества вариантов решений задачи (2.37)-(2.39) в виде "дерева ветвлений" (рис. 2.29).

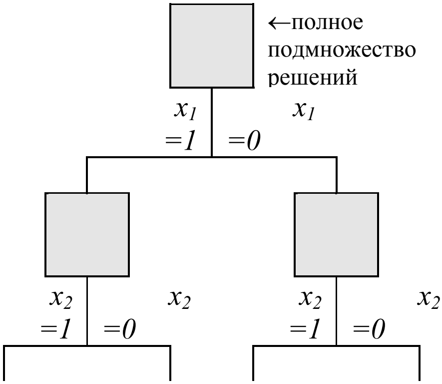


Рис.2.29. Дерево ветвлений

Если не учитывать ограничений, то исходное множество возможных решений будет включать 2^n вариантов.

Последовательность конкретных значений переменных образует ветвь дерева – по существу, это некоторый двоичный n-разрядный код.

Ясно, что некоторые ветви не удовлетворяют системе ограничений (2.38). Такие ветви исключаются из дерева ветвлений.

В ряде случаев удается установить, что все ветви, исходящие из некоторой вершины, с которой связано вполне определенное подмножество вариантов, соответствуют недопустимым решениям.

То есть это подмножество пустое – оно не содержит ни одного допустимого решения.

Возникает вопрос, как, не опускаясь до отдельных элементов подмножества, т.е., не просматривая все ветви, исходящие из соответствующей вершины, установить, что это подмножество пустое?

Для этого необходимо сформулировать некоторый признак пустого подмножества [8,9]. Этот признак введем следующим образом.

Преобразуем систему неравенств

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i \ , \ (i=1,2,...,m)$$

таким образом, чтобы все слагаемые в левой части этой системы стали неотрицательными. Как это сделать?

Допустим, в i-м ограничении ($i=1,2,...,m$) имеется отрицательный коэффициент $a_{ij} < 0$. Тогда вместо x_j в i-е ограничение вводится переменная $x'_j = 1 - x_j$.

Очевидно, что x'_j – это также булева переменная.

При этом замена переменных в других ограничениях не производится.

$$a_{i1}x_1 + ... - a_{ij}x_j + ... + a_{in}x_n \le b_i \ (a_{ij} \ge 0)$$

$$a_{i1}x_1 + ... + a_{ij}x'_j + ... + a_{in}x_n \le b_i + a_{ij}.$$

В результате таких преобразований для всех отрицательных коэффициентов, все коэффициенты левой части должны стать неотрицательными. Коэффициенты правых частей также должны быть неотрицательными.

Очевидно, что если некоторое $b'_i = b_i - a_{ij} < 0$, то i-е ограничение противоречиво. Ему не может удовлетворить ни один набор значений переменных. Т.е. соответствующее множество решений – пустое. Задача не имеет решений. Это и есть признак пустого множества.

Назовем операцию приведения системы ограничений к виду, у которого все $a_{ij} > 0$, операцией приведения системы ограничений к стандартному виду [8,9].

Пример 2.23

Привести к стандартному виду задачу:

$$\begin{gathered} 2x_1 - 4x_2 \to \max \\ -2x_1 + 5x_2 \le 4 \\ 5x_1 - 3x_2 \le 3 \\ x_{1,2} \in \{0,1\} \end{gathered}$$

Основная: $y_2 = 1 - x_2$

$$\begin{gathered} 2x_1 + 4y_2 - 4 \to \max \\ -2x_1 - 5y_2 \le -1 \\ 5x_1 + 3y_2 \le 6 \\ x_1, y_2 \in \{0,1\} \end{gathered}$$

Стандартная

$$\begin{gathered} 2x_1 + 4y_2 - 4 \to \max \\ 2x_1' + 5y_2' \le 6 \\ 5x_1 + 3y_2 \le 6 \\ x_1, x_2, y_2, y_2' \in \{0,1\} \end{gathered}$$

Пусть зафиксированы значения первых k переменных: $\overline{x}_1, \overline{x}_2, ..., \overline{x}_k$. Этому фиксированному набору нулей и единиц соответствует некоторая вершина в дереве ветвлений. Подставим эти значения в систему ограничений задачи (2.38) ЛЦП(б):

$$\sum_{j=1}^{k} a_{ij}\overline{x}_j + \sum_{j=k+1}^{n} a_{ij}x_j \le b_i \,,\ (\, i = \overline{1,m}\,)\,.$$

Имеем систему:

$$\sum_{j=k+1}^{n} a_{ij}x_j \le b_i - \sum_{j=1}^{k} a_{ij}\overline{x}_j \,.$$

Придадим этой системе стандартную форму:

$$\sum_{j=k+1}^{n} a_{ij}x_j^i \le b'_i \,.$$

Очевидно, что если какой-нибудь свободный член станет отрицательным, например $b'_i < 0$, то i-е ограничение – противоречивое. То есть вершину дерева, соответствующую фиксированному набору $\overline{x}_1, \overline{x}_2, ..., \overline{x}_k$, можно отбросить вместе со всеми исходящими из нее ветвями.

В результате такого отсеивания недопустимых ветвей можно существенно сократить размерность задачи.

Но в практических ситуациях полный перебор оставшихся ветвей едва ли может привести к оптимальному решению в приемлемое время.

Здесь приведен просто прием, который используется во многих алгоритмах решения ЛЦП(б) задачи. Первым таким алгоритмом является алгоритм плотного заполнения.

2.6.2. Алгоритм плотного заполнения

Этот алгоритм, ориентированный на поиск допустимого решения ЛЦП(б) задачи, служит в качестве основы многих алгоритмов.

Система ограничений приводится к стандартной форме.

Если в некотором ограничении срабатывает признак пустого подмножества (отрицательная правая часть соответствующего неравенства), то задача не имеет решения – множество ее допустимых решений пусто.

Затем переменным задачи поочередно приписываются единичные значения. Одновременно пересчитываются правые части системы ограничений и проверяется признак пустого подмножества.

Пусть переменным $x_1, x_2, ..., x_{k-1}$ уже присвоены конкретные значения. Система ограничений имеет стандартную форму:

$$\sum_{j=k}^{n} a_{ij} x_j^i \le b_i^{(k-1)} \quad (i = \overline{1,m}), \tag{2.40}$$

$$x_j^i \in \{x_j, x_j'\};\; x_j = 1 - x_j';\; x_j \in \{0,1\};\; x_j \in \{0,1\};\; i = \overline{1,m}.$$

Рассмотрим общий шаг *k*.

<u>Шаг *k*</u>. Попытаемся приписать переменной x_k единичное значение. Для этого подставим x_k =1 в ограничения (2.40), учитывая вхождение этой переменной в каждое ограничение в прямом (x_k) или в инверсном виде (x_k).

Пересчитываем правые части ограничений:

$$\sum_{j=k+1}^{n} a_{ij} x_j^i \le b_i^{(k-1)} - a_{ik}\beta_{ik} = b_i^{(k)} \quad (i = \overline{1,m}), \tag{2.41}$$

где $\beta_{ik} = \begin{cases} 0, & \textit{если } x_k \textit{ входит в ограничение в виде } x_k; \\ 1, & \textit{если } x_k \textit{ входит в ограничение в виде } x_k'. \end{cases}$

Если все $b_i^{(k)} \geq 0$, принимаем $x_k = 1$ и выполняем шаг $k0$. В противном случае – шаг $k1$.

Шаг *k*0. Если $k=n$, то получено допустимое решение задачи. Конец. В противном случае выполняется шаг $k+1$.

Шаг *k*1. Попытаемся приписать переменной x_k нулевое значение. Для этого подставим $x_k = 0$ (x_k) в ограничения (2.40) и пересчитаем правые части ограничений.

Если все $b_i^{(k)} \geq 0$, принимаем $x_k = 0$ и выполняем шаг k0. В противном случае – шаг k2.

Шаг *k*2. Переменной x_k нельзя приписать ни единичного, ни нулевого значения.

Последовательно (в обратном порядке) просматриваем переменные $x_1, x_2, ..., x_{k-1}$ до тех пор, пока не обнаружим некоторую переменную x_k , которой приписано единичное значение. Если такой переменной нет, то конец – задача не имеет допустимых решений. В противном случае в исходную систему ограничений подставляем зафиксированные ранее значения переменных $x_1, x_2, ..., x_{k-1}$. Полагаем $k=s$, приводим систему ограничений к стандартному виду и выполняем шаг $k1$.

Пример 2.24

Используя алгоритм плотного заполнения, найти допустимое решение следующей задачи ЛЦП(б).

$$4y_1 + 2y_2 - y_3 \to \max$$

$$2y_1 + y_2 + y_3 \leq 2$$

$$y_1 - y_2 + y_3 \geq 1$$

$$y_j \in \{0,1\}, \quad j = \overline{1,3}$$

Произведя подстановку $y_1 = x_1$, $y_1 = x_1$, $y_1 = x_1$, приводим задачу к форме основной задачи ЛЦП(б):

$$4x_1 + 2x_2 + x_3 - 1 \to \max$$

$$2x_1 + x_2 - x_3 \le 1$$

$$-x_1 + x_2 + x_3 \le 0$$

$$x_j \in \{0,1\}, \quad j = \overline{1,3}$$

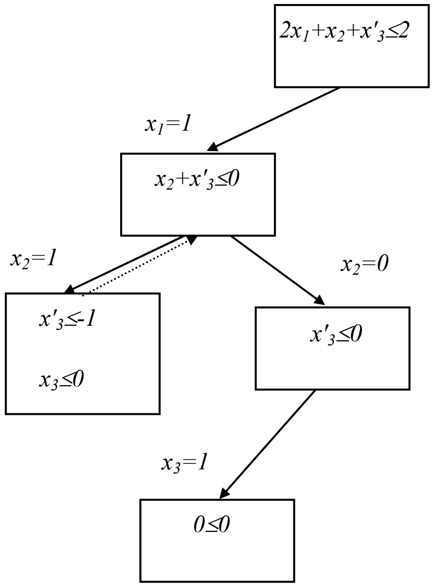
Приводим систему ограничений к стандартному виду:

$$4x_1 + 2x_2 + x_3 - 1 \to \max$$

$$2x_1 + x_2 + x'_3 \le 2$$

$$x'_1 + x_2 + x_3 \le 1$$

$$x_j, x'_j \in \{0,1\}, \quad x_j = 1 - x'_j, \quad j = \overline{1,3}$$



Т.е. $y_1 = x_1$, $y_1 = x_1$, $x_3 = 1$. Решение исходной задачи:

$$y_1 = x_1 = 1$$

$$y_2 = x_2 = 0 \qquad\qquad \text{ЦФ=4.}$$

$$y_3 = 1 - x_3 = 0$$

Этот алгоритм является основой многих алгоритмов решения ЛЦП(б)-задач. Его главный недостаток – резкое увеличение времени работы, если задача не имеет допустимых решений.

2.6.3. Метод Фора и Мальгранжа

Этот метод используется для решения задачи ЛЦП(б), в которой все коэффициенты целые.

$$F(x)=\sum_{j=1}^{m} c_j x_j \rightarrow \max \;\; (c_j \text{ – целые, } j=1,2,...,n) \tag{2.42}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i \;\; (i=1,2,...,m) \tag{2.43}$$

$$x_j \in \{0,1\} \;\; (j=1,2,...,n). \tag{2.44}$$

Приведем задачу к виду основной задачи ($c_j \ge 0$, $j=1,2,...,n$) и попытаемся найти любое ее допустимое решение, используя, например, алгоритм плотного заполнения.

Пусть задача имеет допустимое решение X^0. $F(X^0)$ – значение ЦФ на этом решении.

К ограничениям задачи (2.43) добавим новое ограничение:

$$\sum_{j=1}^{n} c_j x_j \ge F(X^0)+1 . \tag{2.45}$$

Попытаемся найти допустимое решение задачи (2.42)-(2.45). Пусть новая задача имеет допустимое решение X^1. $F(X^1)$ – значение ЦФ на этом решении.

Заменим ограничение (2.45) новым ограничением: $\sum_{j=1}^{n} c_j x_j \ge F(X^1)+1$ и снова попытаемся найти допустимое решение задачи (2.43)-(2.45).

Этот процесс продолжается до тех пор, пока на некотором шаге k+1 не выяснится, что задача с дополнительным ограничением $\sum_{j=1}^{m} c_j x_j \ge F(X^k)+1$ не имеет допустимых решений. Очевидно, что решение предыдущей задачи X^k – оптимальное решение исходной задачи, а $F(X^k)$ – оптимальное значение ЦФ.

2.6.4. Аддитивный алгоритм

Этот метод – один из наиболее популярных и мощных для решения задач ЛЦП(б) вида (2.42)-(2.44).

Рассмотрим два подмножества множества N – ω и v такие, что $\omega \cup v = N$ и $\omega \cap v = \varnothing$.

Каждой переменной x_j ($j \in \omega$) придадим конкретное значение $\overline{x}_j$ (0 или 1). Эти переменные будем называть *фиксированными*.

Переменные x_j ($j \in v$) оставим в прежнем виде и назовем их *свободными*.

Формально определим вектор $y=(y_1, y_2, \ldots, y_n)$, состоящий из следующих компонент:

$$y_j = \begin{cases} \overline{x}_j, & \text{если } \; j \in \omega; \\ x_j, & \text{если } \; j \in \text{v}. \end{cases}$$

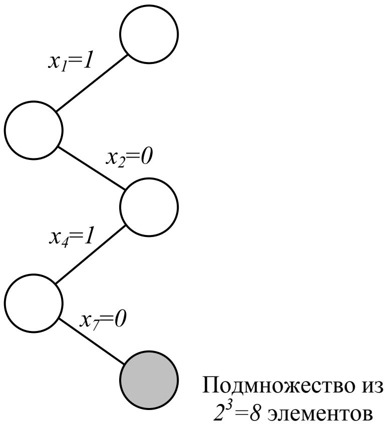
Будем называть этот вектор *частичным решением*.

Каждому частичному решению поставим в соответствие вершину дерева ветвлений или, что то же самое, определенное подмножество решений. При этом путь от корня к этой вершине определяется значениями фиксированных переменных.

Пример 2.25

$y=(1,0,x_3,1,x_5,x_6,0)$. Здесь $\omega=\{1,2,4,7\}$, $v=\{3,5,6\}$.

Для этого частичного решения дерево ветвлений выглядит следующим образом:



Если $\omega=\varnothing$, то имеет место частный случай частичного решения – все переменные свободные ($v\equiv N$).

Любому частичному решению соответствует вполне определенная задача ЛЦП(б), которая получается из исходной задачи в результате подстановки в нее фиксированных значений переменных данного частичного решения и приведения полученной задачи к стандартной форме.

$$\sum_{j\in\omega} c_j \bar{x}_j + \sum_{j\in v} c_j x_j \to \max$$

$$\sum_{j\in v} a_{ij} x_j \le b_i - \sum_{j\in\omega} a_{ij}\bar{x}_j \ (i=\overline{1,m})$$

$$x_j \in \{0,1\}, \ \ j\in v$$

$$\Downarrow$$

$$\sum_{j\in\omega} c_j \bar{x}_j + \sum_{j\in v} c_j x_j \to \max$$

$$\sum_{j\in v} a_{ij} x_j^i \le \tilde{b}_i, \ \ (i=\overline{1,m})$$

$$x_j^i \in \{x_j, x'_j\}, \ x'_j = 1 - x_j, \ x_j \in \{0,1\}, \ j\in v$$

Очевидно, что если некоторое $\tilde{b}_i < 0$, то соответствующее ограничение противоречиво.

Следовательно, данному частичному решению соответствует недопустимая задача: все ветви, выходящие из вершины можно отбросить (как это делается в алгоритме плотного заполнения).

В противном случае (все $\tilde{b}_i \ge 0, \ i=\overline{1,m}$) можно предпринять попытку расширить частичное решение.

Рассмотрим некоторое ограничение задачи:

$$\sum_{j\in v} a_{ij} x_j^i \le \tilde{b}_i, \ \ (i=\overline{1,m}) .$$

Допустим, в этом ограничении существует коэффициент

$$a_{ij} > \tilde{b}_i . \tag{2.46}$$

Это говорит о том, что переменная x_j не может принять значение "1", если она входит в данное ограничение в прямом виде (в виде x_j). Если же переменная x_j входит в ограничение в инверсном виде – в виде x'_j ($x'_j = 1 - x_j$), то x_j не может принять значение "0".

Таким образом, наличие условия (2.46) позволяет приписать переменной x_j только определенное значение.

Придадим этой переменной соответствующее значение. Фактически имеем новое частичное решение: один элемент j из подмножества v перешел в подмножество ω.

Для полученного нового частичного решения построим задачу в стандартной форме и проведем аналогичный анализ: сначала на допустимость ($\exists\ \tilde{b}_i < 0$), затем – на предмет существования коэффициента $a_{ij} > \tilde{b}_i$. Если признака недопустимости задачи нет (все $\tilde{b}_i \geq 0$, $i = \overline{1,m}$) и $\exists\, a_{ij} > \tilde{b}_i$, придадим переменной x_j соответствующее значение и повторим процедуру для нового частичного решения и т.д.

В результате рано или поздно возникнет одна из трех ситуаций:

1. На некотором шаге установлена недопустимость задачи.
2. Нет признака пустого подмножества, но условие (2.46) не выполняется – получено новое частичное решение.
3. Все элементы из подмножества v перешли в подмножество ω (получено полное решение задачи).

Совокупность соответствующих операций до возникновения одной из перечисленных ситуаций назовем *расширением частичного решения*.

Рассмотрим подробнее эти ситуации.

1. На некотором шаге установлена недопустимость задачи.

В этом случае исходное частичное решение недопустимо (рис. 2.30).

2. Нет признака пустого подмножества, но условие (2.46) не выполняется. Получено новое частичное решение, которое нельзя расширить. В этом случае принудительно производится ветвление. Из одного частичного решения формируются два новых частичных решения: любой свободной переменной сначала приписывается значение "0", а затем "1".

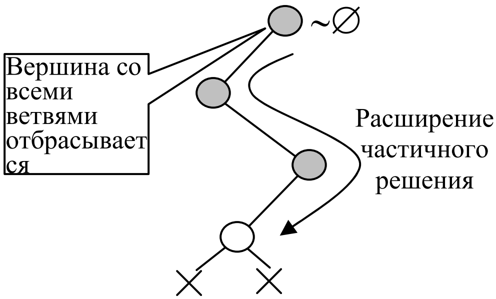
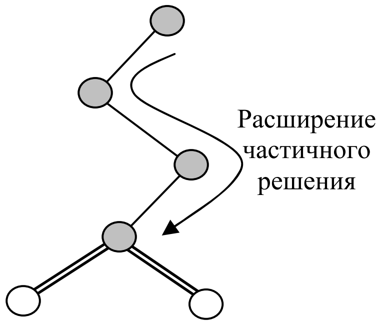


Рис.2.30. Случай недопустимости задачи

Далее предпринимается попытка расширить оба новых решения.



Новые частичные решения

Рис.2.31. Случай ветвления

Очевидно, что нужно провести расширение сначала одного нового решения, а затем – другого. То есть необходимо создать список частичных решений.

3. Получено полное решение задачи, т.е., допустимое решение исходной задачи.

Возникает вопрос, как наиболее эффективно использовать полученную информацию. Вот здесь-то и используется механизм, который уже рассматривался в методе Фора и Мальгранжа.

Этот механизм позволяет ускорить процесс отсеивания частичных решений, которые не только являются недопустимыми, но и, что самое интересное, заведомо неоптимальными.

Работа этого механизма будет рассмотрена несколько позже. Сейчас же приведем важнейшее в аддитивном алгоритме определение (уже знакомое из метода ветвей и границ).

Допустимое решение $X^* = (x_1^*, x_2^*, ..., x_n^*)$, которое доставляет ЦФ значение Z^* – большее, чем любое известное к данному моменту решение, называется *рекордом*.

Рекорды мы будем регистрировать – запоминать только последний рекорд.

В принципе, можно каждое полное частичное решение сравнивать с рекордом: если рекорд не улучшился, отбрасывать это решение.

Мы же воспользуемся механизмом метода Фора и Мальгранжа.

После того, как получен очередной рекорд (X^*, Z^*), в систему ограничений каждой из задач, которая соответствует некоторому, еще не расширенному решению, будем добавлять ограничение:

$$\sum_{j=1}^{n} c_j x_j \ge F(X^*) + 1 .$$

Таким образом, каждое полное расширенное решение будет соответствовать новому рекорду.

Ниже приводится общая схема аддитивного алгоритма.

Шаг 1. Положим $\omega=\varnothing$, $v\equiv N$ (все переменные – свободные). Придадим задаче стандартную форму и занесем ее в предварительно очищенный список задач.

Шаг 2. Если список задач пуст, то проверяем, был ли ранее зафиксирован рекорд. Если рекорд был зафиксирован, задача решена: рекорд X^* – оптимальное решение, Z^* – оптимальное значение ЦФ. Конец.

Шаг 3. Если при пустом списке рекорд не зафиксирован, задача не имеет допустимых решений. Конец.

Шаг 4. Если список задач не пуст, выберем из него любую задачу. По признаку пустого подмножества ($\exists\, \tilde{b}_i < 0$) проверяем допустимость задачи, и, если признак есть, переходим к шагу 2. Если признака нет (все $\tilde{b}_i \ge 0,\ i = \overline{1,m}$), выполняется следующий шаг.

Шаг 5. Производим расширение частичного решения, которому соответствует выбранная задача. Если устанавливается факт недопустимости задачи, соответствующей расширенному решению ($\exists$ $\tilde{b}_i < 0$) переходим к шагу 2.

Шаг 6. Если расширенное частичное решение полное, то получен новый рекорд (X^*, Z^*).

Во все задачи списка вносим дополнительное ограничение

$$\sum_{j=1}^{n} c_j x_j \geq F(X^*) + 1,$$

учитывая соответствующие частичные решения. Всем задачам списка придаем стандартную форму и переходим к шагу 2.

Шаг 7. Имеем неполное расширенное частичное решение. Возьмем любую свободную переменную этого решения x_k ($k \in v$). Полагая $x_k=1$ и $x_k=0$ ($v=v \backslash k$, $\omega=\omega \cup k$) формируем два частичных решения. Каждому из этих частичных решений ставим в соответствие задачу в стандартной форме (естественно, с учетом дополнительного ограничения, если ранее был зафиксирован рекорд). Заносим эти задачи в список и переходим к шагу 2.

Пример 2.26

Используя аддитивный алгоритм, решить задачу ЛЦП(б):

$$10x_1 + 2x_2 + 3x_3 + 14x_4 \to \max$$

$$-4x_1 - 5x_2 - 10x_3 + 4x_4 \leq 11$$

$$9x_1 + 4x_2 + 3x_3 - 5x_4 \leq 5$$

$$x_j \in \{0,1\},\ j = \overline{1,4}.$$

Положим $\omega=\varnothing$, $v=\{1,2,3,4\}$ – все переменные частичного решения – свободные. Это решение имеет вид x_1, x_2, x_3, x_4 .

Задача имеет вид основной задачи. Придадим ей стандартную форму.

$$10x_1 + 2x_2 + 3x_3 + 14x_4 \to \max$$

$$4x'_1 + 5x'_2 + \boldsymbol{10x'_3} + 4x_4 \leq \boldsymbol{8}$$

$$9x_1 + 4x_2 + 3x_3 + 5x'_4 \leq 10$$

Занесем эту задачу в список (для упрощения записи спецификация переменных опущена).

Выберем задачу из списка. Ее частичное решение x_1, x_2, x_3, x_4 . Признака пустого подмножества нет. Начнем расширять частичное решение.

$x'_3 = 0 \;\Rightarrow x_3 = 1$. Новое ЧР $x_1, x_2, 1, x_4$.

Новая задача:

$$10x_1 + 2x_2 + 14x_4 + 3 \to \max$$

$$4x'_1 + 5x'_2 + \quad 4x_4 \le 8$$

$$\boldsymbol{9}x_1 + 4x_2 + \; 5x'_4 \le 7$$

$x_1 = 0 \;\Rightarrow x'_1 = 1$. Новое ЧР $0, x_2, 1, x_4$.

Новая задача:

$$2x_2 + 14x_4 + 3 \to \max$$

$$\boldsymbol{5}x'_2 + \quad 4x_4 \le \boldsymbol{4}$$

$$4x_2 + \; 5x'_4 \le 7$$

$x'_2 = 0 \;\Rightarrow x_2 = 1$. Новое ЧР $0,1,1,x_4$.

Новая задача:

$$14x_4 + 5 \to \max$$

$$4x_4 \le 4$$

$$\boldsymbol{5}x'_4 \le \boldsymbol{3}$$

$x'_4 = 0 \;\Rightarrow x_4 = 1$. Новое ЧР $0,1,1,1$ полное. Получен рекорд.

$$X^* = (0,1,1,1),\; Z^* = 19.$$

Список задач пустой. Рекорд зафиксирован. Следовательно, имеем оптимальное решение ЛЦП(б)-задачи:

$$X^{opt} = (0,1,1,1),\; Z^{opt} = 19.$$

2.6.5. Моделирование логических высказываний в задачах булева программирования

Логическое высказывание T_j, которое может быть либо истинным, либо ложным, можно моделировать бинарной (булевой) переменной x_j:

$$x_j = \begin{cases} 1, & \textit{если } T_j \textit{ истинно}, \\ 0, & \textit{если } T_j \textit{ ложно}. \end{cases}$$

Отрицание ($\rceil T_j$) моделируется, как ($1 - x_j$).

Промоделируем некоторые высказывания.

1. "T_1 *или* T_2 *или ...или* T_k *истинно*". В этом случае нужно сформулировать ограничение, которое "заставит" принять единичное значение по крайней мере одну переменную. Таким ограничением может служить следующее:

$$x_1 + x_2 + ... + x_k \geq 1.$$

2. "T_1 *и* T_2 *и ... и* T_k *истинно*". Модель этого высказывания имеет вид:

$$x_1 + x_2 + ... + x_k \geq k .$$

3. "*Одно и только одно из высказываний* T_1 *и* T_2 *и ... и* T_k *истинно*":

$$x_1 + x_2 + ... + x_k = 1.$$

А теперь рассмотрим сложные высказывания.

4. "T_{k+1} *истинно тогда и только тогда, когда все высказывания* T_1 *и* T_2 *и ... и* T_k *истинны*": $T_{k+1} = T_1 \wedge T_2 \wedge ... \wedge T_k$. Это высказывание моделируется двумя ограничениями:

а) $x_1 + x_2 + ... + x_k \geq kx_{k+1}$ – если хотя бы l высказываний ложны, то $k - l \leq kx_{k+1}$, откуда следует, что x_{k+1} *=0,*

б) $x_1 + x_2 + ... + x_k - (k-1) \leq x_{k+1}$ – если все высказывания истинны, то $k - l \leq kx_{k+1}$, откуда следует, что x_{k+1} *=1.*

5. "T_{k+1} *истинно тогда и только тогда, когда хотя бы одно из высказываний* T_1 *и* T_2 *и ... и* T_k *истинно*":

$T_{k+1} = T_1 \wedge T_2 \wedge ... \wedge T_k$. Это высказывание также моделируется двумя ограничениями:

а) $x_1 + x_2 + ... + x_k \leq kx_{k+1}$ – если хотя бы одно из высказываний истинно, то x_{k+1} *=1*;

б) $x_1 + x_2 + ... + x_k \geq x_{k+1}$ – если все высказывания ложны, то $k - l \leq kx_{k+1}$, откуда следует, что x_{k+1} *=0***.**

Введенная таким образом формализация уже вполне достаточна, чтобы моделировать высказывания любой сложности.

Пример 2.27

Возьмем импликацию:

$$T_3 = T_1 \to T_2 .$$

(T_3 ложно тогда и только тогда, когда T_1 истинно, а T_2 ложно). Эквивалентная запись этого высказывания имеет вид:

$$T_3 = \overline{T_1} \vee T_2 .$$

Высказывание $T_3 = \overline{T_1} \vee T_2$ моделируется как ($1- x_j$), а дизъюнкция моделируется, как уже было показано в пп.5 – двумя ограничениями:

$$(1 - x_1) + x_2 \le 2x_3 ,$$

$$\left(1 - x_1\right) + x_2 \ge x_3 .$$

Итак, возможность моделирования высказываний любой сложности открывает широкие перспективы для моделирования таких сложных ситуаций, ярким примером одной из которых является оптимизация планирования взаимоувязанного комплекса работ по созданию сложной системы.

Необходимо наладить выпуск продукции и обеспечить максимальную прибыль от ее реализации.

Сразу возникает масса альтернативных вариантов, где размещать производство и какую технологию использовать для выпуска продукции.

Для каждого альтернативного варианта возникает свой круг проблем:

- Ресурсное обеспечение (материальные, трудовые, финансовые ресурсы). Здесь также возможны варианты.
- Оптимальная кооперация (каких смежников привлекать, что оставлять смежникам, что производить самим).
- Инфраструктура (сбыт продукции). Какие требуются коммуникации.
- Экология, социальное развитие, капитальные вложения в непроизводственную сферу.
- Возможность переориентации производства при насыщении рынка.

При решении подобных проблем вся плановая информация, при хорошей ее структуризации, развертывается в граф, вершинам которого соответствуют работы по созданию отдельных подсистем (рис. 2.32).

Основная логика этого графа – это "И/ИЛИ". Построение же графа идет сверху вниз.

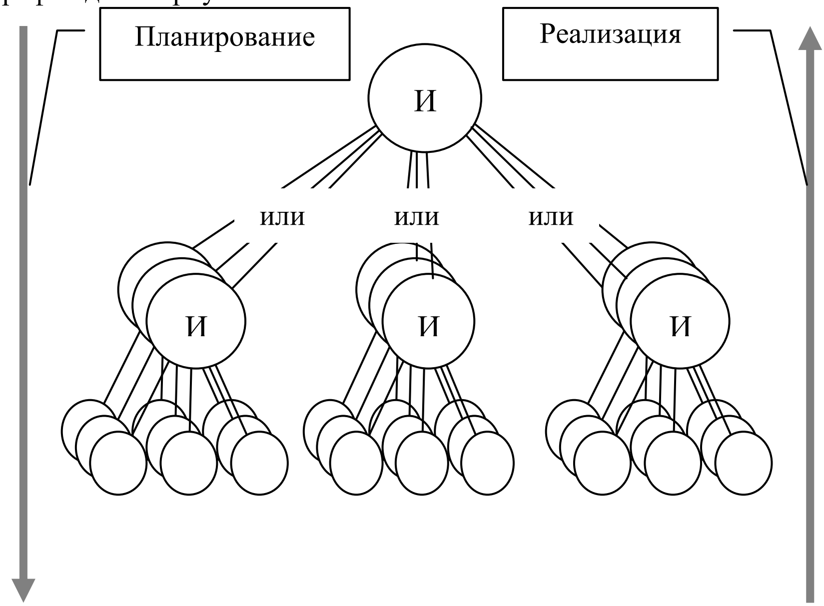


Рис.2.32. Дерево целей

В процессе планирования этот граф последовательно (сверху вниз) преобразуется в граф, имеющий логику "И". То есть для каждой цели (подцели) выбирается свой вариант реализации из множества альтернативных вариантов.

Полученный таким образом граф преобразуется в исполнительный план, в план-график, для работы с которым используются методы сетевого планирования.

Вот здесь-то следует подчеркнуть одну важную особенность булева программирования.

Модели ЛЦП(б) позволяют учитывать взаимную обусловленность не только работ, относящихся к одному комплексу (работа такая-то не может быть выполнена до тех пор, пока не будет выполнены все работы, связанные с достижением целей нижнего уровня), но и работ, относящихся к разным комплексам.

Это как раз и выражается определенными логическими высказываниями, которые включаются в модель задачи в виде линейных ограничений с булевыми переменными.

Остановимся на основных типах высказываний, задающих взаимообусловленность плановых решений и на моделях этих высказываний – на вопросах моделирования логических связей.

2.6.6. Моделирование логических связей в задачах булева программирования

В качестве объекта планирования возьмем самый общий объект – большую сложную систему, состоящую из ряда подсистем.

Пусть x_j – булева переменная, значение которой определяет включение в план (исключение из плана) разработку некоторой подсистемы A_j :

$$x_j = \begin{cases} 1, & \textit{разработка } A_j \textit{ включается в план}, \\ 0, & \textit{разработка } A_j \textit{ не включается в план}. \end{cases}$$

То есть переменная x_j моделирует атомарное высказывание A_j: "*создать подсистему* A_j".

Информация, предоставляемая для разработки плана создания системы, помимо перечня вариантов проектов и их составляющих, содержит также сведения об их логических взаимосвязях. Приведем примеры моделирования типичных логических связей.

1. *Создание любой системы из множества* $A_1, A_2, \ldots, A_k$ *исключает создание других систем этого множества.*

Это так называемое "альтернативное разбиение", которое соответствует высказыванию:

$$T_1 \underline{\cup} T_2 \underline{\cup} \ldots \underline{\cup} T_k .$$

Это высказывание ложно тогда и только тогда, когда истинными являются более одной пропозициональной переменной.

Высказывание моделируется следующим ограничением:

$$x_1 + x_2 + \ldots + x_k \le 1 .$$

2. *Система* A_i *может быть создана только после того, как будет создана система* $A_1, A_2, ..., A_k$.

Высказывание: $T_i \to T_k$. Модель этого высказывания:

$$x_k \geq x_i .$$

3. *Подсистему* A_i *можно начать разрабатывать только в том случае, если будут разработаны все подсистемы множества* $A_1, A_2, ..., A_k$.

Высказывание: $T_i \to T_k$. Модель этого высказывания:

$$x_1 + x_2 + ... + x_k \geq k x_{k+1} .$$

4. *Подсистему* A_i *можно начать разрабатывать только в том случае, если будет разработана по крайней мере одна из подсистем множества* $A_1, A_2, ..., A_k$.

Высказывание: $T_i \to T_k$. Модель этого высказывания:

$$x_1 + x_2 + ... + x_k \geq x_{k+1} .$$

В завершение этого раздела необходимо подчеркнуть следующее обстоятельство. Сложные предложения, включенные в информацию, представленную для разработки плана, всегда имеют значение "истинно".

Задача разработки плана является в определенном смысле обратной той задаче, которая решается с помощью таблиц истинности – по известным значениям пропозициональных переменных найти значения сложных логических выражений. В нашем же случае требуется найти значения пропозициональных переменных, обеспечивающие истинность логических выражений, включенных в математическую модель задачи в виде соответствующих ограничений.

Пример 2.28

На модернизацию АЭС, состоящей из трех энергоблоков, выделена сумма А.

Для каждого энергоблока i ($i = 1, 2, 3$) подготовлено m_i проектов модернизации: $P_{i1}, P_{i2}, ..., P_{i\,m_i}$, из которых может быть выбран только один вариант (а может, и ни одного).

Реализация проекта P_{ij} требует затрат на сумму a_{ij} и дает дополнительный доход в сумме c_{ij}.

Построить модель задачи выбора проекта модернизации, обеспечивающего максимальный дополнительный доход при условии, что проект P_{12} может быть реализован только после того, как будут реализованы проекты P_{21} и P_{31}.

Решение.

Вводим булевы переменные $x_{ij} \in \{0,1\}$, такие, что:

$$x_{ij} = \begin{cases} 1, & \textit{если } i-\textit{й энергоблок модернизируется по } j-\textit{му проекту} \\ 0, & \textit{в противном случае.} \end{cases}$$

Моделируем дополнительное ограничение:

$x_{21} + x_{31} \geq 2x_{12}$.

Окончательно:

$$\sum_{i=1}^{3}\sum_{j=1}^{m_j} c_{ij}x_{ij} \to \max,$$

$\displaystyle\sum_{i=1}^{3}\sum_{j=1}^{m_j} a_{ij}x_{ij} \leq A$ – ограничение на затраты на модернизацию.

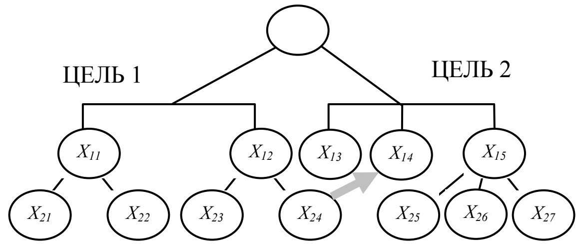
$\displaystyle\sum_{j=1}^{m_i} x_{ij} \leq 1$, $i = 1, 2, 3,$ – не более одного варианта модернизации.

$x_{21} + x_{31} \geq 2x_{12}$, – модель дополнительного ограничения.

$x_{ij} \in \{0,1), \ i = 1, 2, 3; \ j = \overline{1, m_i}$.

Пример 2.29

Задан граф планирования.



1. Главная цель может быть достигнута, если будет достигнута ЦЕЛЬ1 и ЦЕЛЬ2.
2. ЦЕЛЬ1 можно достичь, только решив одну из задач X_{11} или X_{12}.
3. ЦЕЛЬ2 можно достичь, только решив одну из трех задач X_{13}, или X_{14}, или X_{15}
4. Для решения задачи X_{11} необходимо решить задачу X_{21} и задачу X_{22} .
5. Для решения задачи X_{12} необходимо решить задачу X_{23} и задачу X_{24}.
6. Для решения задачи X_{15} необходимо решить все три задачи X_{25}, X_{26} и X_{27}.
7. Кроме того, есть еще дополнительное ограничение: задачу X_{14} можно решить только при условии, что задача X_{24} решена.

Необходимо сформулировать систему ограничений, при помощи которых можно смоделировать перечисленные логические связи.

Решение.

В данном случае можно предложить целый ряд вариантов. Рассмотрим один из них.

Одна из двух задач X_{11} или X_{12} должна быть обязательно решена:

$$X_{11} + X_{12} = 1.$$

Одна из трех задач X_{13} , X_{14} или X_{15} должна быть обязательно решена:

$$X_{13} + X_{14} + X_{15} = 1.$$

Для решения задачи X_{11} необходимо решить задачу X_{21} и задачу X_{22}:

$$X_{21} + X_{22} \geq 2X_{11}.$$

Для решения задачи X_{12} необходимо решить задачу X_{23} и задачу X_{24}:

$$X_{23} + X_{24} \geq 2X_{12}.$$

Для решения задачи X_{15} необходимо решить все три задачи X_{25}, X_{26} и X_{27}:

$$X_{25} + X_{26} + X_{27} \geq 2X_{15}.$$

Наконец, дополнительное ограничение: задачу X_{14} можно решить только при условии, что задача X_{24} решена:

$$X_{24} \geq X_{14}.$$

Контрольные вопросы и задачи к разделу 2.6

1. Сформулируйте признак пустого подмножества в задаче булева программирования.

2. Чем отличаются основная и стандартная формы задач булева программирования?

3. Сравните метод плотного заполнения с полным перебором всех комбинаций булевых переменных.

4. Зачем в методе Фора и Мальгранжа наложено требование целочисленности коэффициентов целевой функции?

5. В чем заключается процесс расширения частичного решения в аддитивном алгоритме?

6. Постройте стандартную форму задачи булева программирования:

$$x_1 - x_2 + 2x_3 - x_4 \to \max$$
$$x_1 - 2x_2 + x_3 \le 5$$
$$x_1 + x_2 - 2x_3 + 3x_4 \ge 3$$
$$x_1 + 3x_2 + x_3 - 4x_4 \le 2$$
$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

7. Получите допустимое решение задачи булева программирования методом плотного заполнения:

$$x_1 + 2x_2 - 3x_3 \to \max$$
$$x_1 + 2x_2 - x_3 \ge 2$$
$$x_1 + x_2 + x_3 \le 2$$
$$-x_1 + x_2 - x_3 \le 0$$
$$x_1, x_2, x_3 \in \{0,1\}$$

8. Решите задачу булева программирования методом Фора и Мальгранжа:

$$2x_1 + x_2 + x_3 \to \max$$
$$x_1 + x_2 = 1$$
$$x_1 + x_2 - x_3 \ge 1$$
$$x_1, x_2, x_3 \in \{0,1\}$$

8. Решите задачу булева программирования аддитивным методом:

$$x_1 + 2x_2 - x_3 + x_4 \to \max$$
$$-2x_1 + x_2 + x_3 + x_4 \le 0$$
$$-x_1 + x_2 - x_3 + x_4 \ge 1$$
$$x_1, x_2, x_3, x_4 \in \{0,1\}$$

10. При проектировании систем автоматической защиты ядерного реактора допускается использовать 4 разновидности автоматических устройств реагирования $У_j$ (установлено может быть не более одного устройства каждого вида). Каждое устройство обладает весом m_j. Конструкция активной зоны не допускает установку защитных устройств с общим весом, превышающим *M*. Установка устройства вида *j* обойдется в c_j рублей. Необходимо определить перечень устройств, включаемых в проект, обеспечивающий проекту минимальные затраты по данной статье, при соблюдении следующих требований:

1.Из соображений надежности должно быть установлено не менее трех защитных устройств;

2.Совместное использование устройств $У_1$ и $У_2$ исключает возможность использование устройства $У_3$;

3.Использование устройства $У_4$ возможно только при использовании устройства $У_2$.

2.7 Динамическое программирование

2.7.1. Общие принципы задач динамического программирования

Динамическое программирование является методом решения оптимизационных задач, основанным на декомпозиции самого вычислительного алгоритма. Основным принципом динамического программирования является разбиение процесса решения задачи на группу последовательных этапов, связанных между собой однонаправленным потоком данных (от предыдущего этапа к последующему). В настоящее время методы динамического программирования получили широкое распространение в большом количестве прикладных областей, начиная математическими задачами анализа графов и заканчивая оптимизацией инвестиционных затрат и теорией управления запасами. Интерес к динамическому программированию обусловлен такими его качествами, как абстрактность и фундаментальность идеи, позволяющая применять их в широких масштабах, а также потенциальная возможность снижения вычислительной сложности решаемых задач за счет их декомпозиции. С другой стороны, при реализации алгоритма динамического программирования для конкретной задачи, необходимо провести её серьезный анализ, и обосновать для неё основные принципы, что является трудной неформализуемой задачей. Зачастую конкретная реализация алгоритма весьма сложна для понимания. Как уже говорилось, метод динамического программирования является абстрактным, подобно методу ветвей и границ. Это означает, что суть самого метода составляет набор основных абстрактных принципов, которые необходимо формализовать для решения конкретной задачи. Рассмотрим эти принципы.

Пусть существует некоторая задача, решаемая с помощью динамического программирования. Процесс решения этой задачи должен быть разбит на последовательные этапы. Состоянием системы x_i на этапе i будем называть некоторый набор данных, получаемый в процессе работы этапа i. Можно выделить основные принципы методов динамического программирования:

- Последовательность этапов – все этапы выполняются друг за другом в жестко заданной последовательности;
- Передача состояния от текущего этапа к последующему – генерируемое в процессе работы этапа i состояние передается этапу $i+1$;
- Рекуррентная природа вычислений. Каждый этап i представляет собой некоторую функцию, которая получает на вход вектор состояния предыдущего этапа x_{i-1}, по определенному правилу выделяет некоторые альтернативы, выбирает из них лучшие и формирует новое состояние x_i:

$$x_i = \Psi(x_{i-1})\,;$$

- *Принцип оптимальности Беллмана*. Оптимальная стратегия на каждом отдельном этапе определяется исключительно на основании текущего состояния, независимо от состояний и оптимальных стратегий на предшествующих этапах.

Указанные принципы можно понимать достаточно широко. Фактически, любой итерационный алгоритм может быть истрактован как метод динамического программирования. Основным признаком правильного подхода к формализации принципов является низкая степень роста объема данных, образующих состояние системы на последующих этапах. Как и метод ветвей и границ, метод динамического программирования должен сужать количество рассматриваемых альтернатив с полного количества комбинаций (полный перебор) до некоторого допустимого, которое делает возможным (реалистичным) применение современных вычислительных средств (направленный перебор). Без обретения этого качества метод динамического программирования, применяемый на *NP*-полной задаче, ничем не отличается от полного перебора, что делает невозможным его применение на задачах больших размерностей. Для его обеспечения и введены принципы рекуррентности и оптимальности. Принцип рекуррентности ограничивает информацию, которую может использовать вычислительный процесс на некотором этапе той, которая хранится в состоянии системы. Однако само понятие состояния системы тождественно понятию информации в широком смысле, что позволяет, например, хранить в состоянии системы на шаге k состояния на всех предыдущих шагах $(x_1, \dots x_{k-1})$. Для наложения ограничений на понятие состояния системы введен принцип оптимальности, который декларирует независимость оптимизационной стратегии на шаге k от стратегий на предыдущих шагах. Косвенно это приводит к невозможности экспоненциального роста состояния системы, поскольку в последнем случае это означало бы зависимость стратегии на шаге k от альтернатив, рассмотренных на предыдущих шагах.

Итак, для решения конкретной задачи методами динамического программирования, необходимо ответить на следующие вопросы:

- Что будет пониматься под отдельным этапом решения задачи?
- Какие альтернативы возникают на каждом этапе решения?

- Как вычислять числовую меру для альтернатив, на основании которой эти альтернативы сравниваются между собой?
- Какая информация должна образовывать состояние x_i на шаге i?
- Каков принцип остановки алгоритма, и существует ли он вообще?

Ответив на эти вопросы, можно работать по следующей обобщенной схеме динамического программирования.

Шаг 1. Положить $k=1$.

Шаг 2. Выполняется k-й этап решения задачи. На основании состояния предыдущего этапа x_{k-1} определить альтернативы, вычислить меру альтернатив. С помощью меры отбросить худшие альтернативы. На основании лучших альтернатив сформировать новое состояние системы x_k.

Шаг 3. Если срабатывает признак остановки алгоритма (например, текущий этап – последний), то из состояния системы на последнем этапе извлекается оптимальное решение задачи. В противном случае положить $k=k+1$ и выполнить шаг 2.

2.7.2. Задача поиска кратчайшего маршрута в графе

Простейшим примером задачи, которую можно решить с помощью динамического программирования, является задача определения кратчайшего пути в графе прямого распространения.

Пусть задан взвешенный ориентированный граф G со специальной структурой. Каждой вершине графа поставлен в соответствие слой – натуральное число от 1 до N. Вершина не может относиться к двум слоям одновременно, равно как и не может быть вершин, не относящихся к какому-либо слою. Дуги в графе существуют только между вершинами двух соседних слоев, причем направление дуги соответствует переходу от вершины с меньшим номером слоя к вершине с большим номером. Первый и последний слои графа содержат ровно одну вершину. Необходимо найти кратчайший маршрут в графе, начинающийся с вершины на первом слое и заканчивающийся вершиной на последнем.

Заданная структура графа определяется требованием отсутствия дуг обратной связи, чтобы в задаче не существовало маршрутов бесконечной длины (циклов). Любой граф, удовлетворяющий этому требованию, может быть преобразован в граф с заданной структурой. Пусть на слое i графа G находится m_i вершин

$r_j^i, j=1..m_i, i=1..N$. Пусть также вес дуги, соединяющей вершины $r_{j_1}^i$ и $r_{j_2}^{i+1}$, равен $c_{j_1 j_2}^i$ (рис. 2.33).

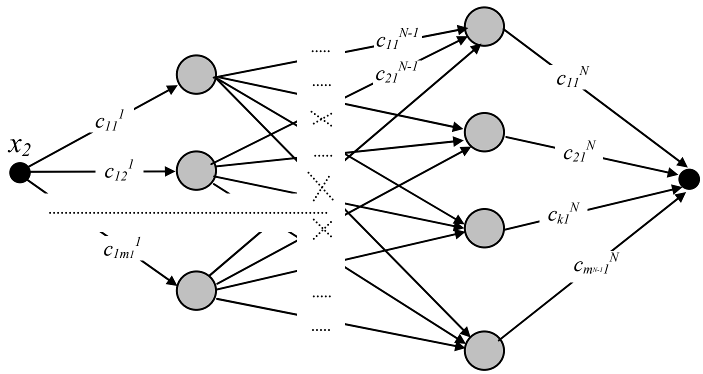


Рис.2.33. Граф прямого распространения сигнала

Произведем формализацию необходимых для применения динамического программирования принципов:

1. Этапом решения задачи *k* будем считать процедуру анализа дуг между слоями *k* и *k+1*.
2. Альтернативами на этапе *k* являются дуги между слоями *k* и *k+1*.
3. Состояние системы на первом этапе представляет собой вектор $x_1 = (s_1^{(1)}, ..., s_{m_2}^{(1)})$, где $s_j^{(1)}$ – дуга из начальной вершины в вершину *j* второго слоя. Элементы вектора состояния системы на этапе *k* определяется по следующему принципу. Для каждой вершины r_l^{k+1} из всех альтернативных дуг, в неё входящих, выбирается такая вершина r_t^k, чтобы выполнялось условие:

$$f^t(x_{k-1}) + c_{tl}^k = \min_{j=1..m_k} (f^j(x_{k-1}) + c_{jl}^k)$$

где $f^j(x_{k-1}) = \left| s_j^{(k)} \right|$ – длина маршрута $s_j^{(k)}$. С точки зрения принципов динамического программирования, величина $f^j(x_{k-1}) + c_{jl}^k$ и есть та самая числовая мера, с помощью которой выбирается альтернатива (в данном случае – с помощью минимизации этой меры).

Элемент $s_l^{(k)}$ вектора состояния x_k, представляющий собой кратчайший маршрут из начальной вершины в вершину r_l^{k+1}, получается добавлением к маршруту $s_t^{(k-1)}$ дуги из вершины r_t^k в вершину r_l^{k+1}.

Таким образом, алгоритм динамического программирования может быть упрощенно записан:

Шаг 1. Положить *k=1*, сформировать вектор $x_1 = (s_1^{(1)},...,s_{m_2}^{(1)})$. Значение целевых функций $f^j(x_1) = c_{1j}^1$. Увеличить k на 1.

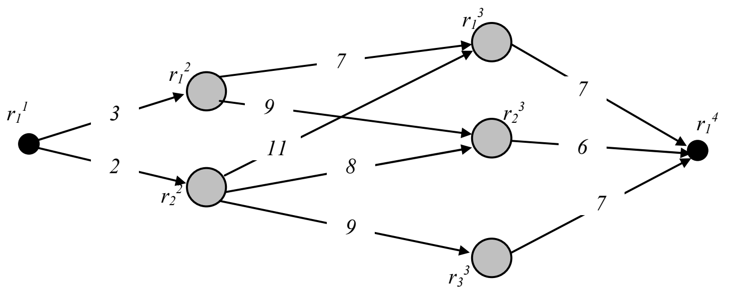
Шаг 2. Имеется вектор $x_{k-1} = (s_1^{(k-1)},...,s_{m_k}^{(k-1)})$. Каждый элемент этого вектора $s_j^{(k-1)}$ представляет собой кратчайший маршрут из начальной вершины в вершину r_j^k. Если $k=N$, то конец – получен самый короткий маршрут. Его длина записана в единственном элементе вектора x_k.

Шаг 3. Сформировать новый вектор состояния $x_k = \left(s_1^{(k)},...,s_{m_{k+1}}^{(k)}\right)$. Для каждой вершины r_l^{k+1} слоя *k+1* определить соответствующий элемент $s_l^{(k)}$ вектора состояния x_k.

Шаг 4. Для каждой вершины r_l^{k+1} вычислить значение целевой функции $f^1(x_k) = \left|s_l^{(k)}\right|$. Перейти к шагу 2.

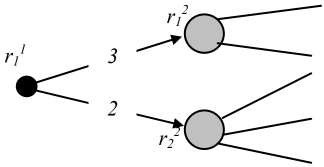
Пример 2.30

Задан граф:



Найти кратчайший маршрут из вершины r_1^1 в вершину r_1^4.

ЭТАП 1

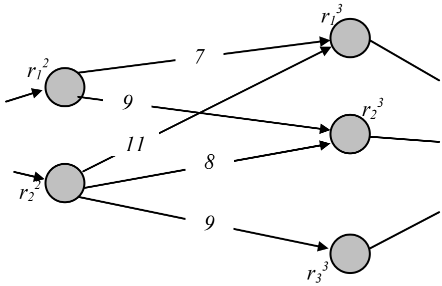


Формируем вектор состояния $x_1 = (s_1^{(1)}, s_2^{(1)})$. Его элементы:

$s_1^{(1)} = r_1^1 \to r_1^2, f^1(x_1) = 3$

$s_2^{(1)} = r_1^1 \to r_2^2, f^2(x_1) = 2$

ЭТАП 2



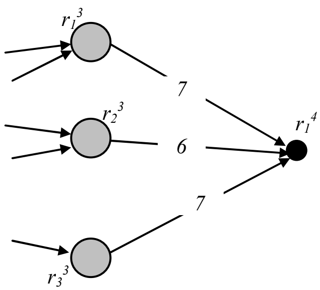
Формируем вектор состояний $x_2 = (s_1^{(2)}, s_2^{(2)}, s_3^{(2)})$:

$\min(3+7, 2+11) = 10 \Rightarrow s_1^{(2)} = r_1^1 \to r_1^2 \to r_1^3, f^1(x_2) = 10$

$\min(3+9, 2+8) = 10 \Rightarrow s_2^{(2)} = r_1^1 \to r_2^2 \to r_2^3, f^2(x_2) = 10$

$\min(2+9) = 11 \Rightarrow s_3^{(2)} = r_1^1 \to r_2^2 \to r_3^3, f^3(x_2) = 11$

ЭТАП 3



Формируем вектор состояний $x_3 = (s_1^{(3)})$:

$\min(10+7,10+6,11+7) = 16 \Rightarrow s_1^{(3)} = r_1^1 \to r_2^2 \to r_2^3 \to r_1^4, f^1(x_3) = 16$.

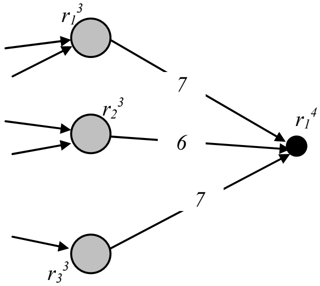
Мы оказались в конечной вершине, поэтому получено оптимальное решение задачи: получен кратчайший маршрут $r_1^1 \to r_2^2 \to r_2^3 \to r_1^4$, длина которого равна 16.

2.7.3. Прямой и обратный ход в задаче динамического программирования

Метод, приведенный выше, работает в прямом направлении: каждый этап добавляет к решению задачи свой фрагмент, и на последнем этапе получается окончательное оптимальное решение. Такой метод получил название алгоритма прямого хода. Однако поскольку этапы вычислений в динамическом программировании выполняются независимо друг от друга и связаны друг с другом только входными и выходными данными, можно формализовать алгоритм, работающий в обратном направлении. Такой алгоритм называется алгоритмом обратного хода. Несмотря на некоторую сложность для человеческого восприятия, алгоритм обратного хода весьма активно используется в математике, и в ряде случаев его использование менее ресурсоемко, чем у алгоритма прямого хода. Основные принципы динамического программирования для алгоритма обратного хода остаются без изменения. Основным его отличием от алгоритма прямого хода заключается в том, что если последний начинает свою работу с пустого решения и добавляет ему по мере работы недостающие фрагменты, то алгоритм обратного хода начинает с предпосылки, что задача решена и, оперируя абстрактным полным решением, постепенно добавляет в него конкретные величины. В ряде случаев (например, в задаче поиска кратчайшего маршрута) его отличие от метода прямого хода весьма иллюзорно, в других – может существенно изменить ход решения. Рассмотрим задачу обратного хода на предыдущем примере.

Пример 2.31

ЭТАП 1



Мы исходим из того, что уже находимся в конечной точке маршрута. Теперь необходимо уточнить все дуги, входящие в этот маршрут. Формируем начальный вектор состояний:

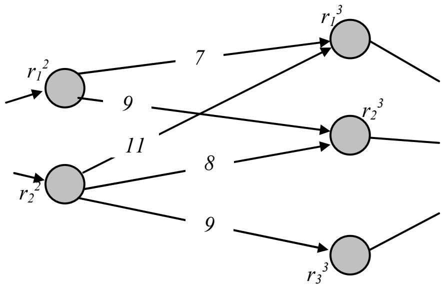
$$x_1 = (s_1^{(1)}, s_2^{(1)}, s_3^{(1)})$$

$$s_1^{(1)} = r_1^3 \to r_1^4, f^1(x_1) = 7$$

$$s_2^{(1)} = r_2^3 \to r_1^4, f^2(x_1) = 6$$

$$s_3^{(1)} = r_3^3 \to r_1^4, f^2(x_1) = 7$$

ЭТАП 2

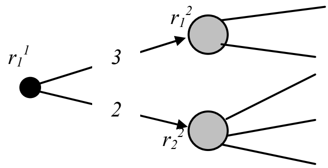


Формируем вектор состояний $x_2 = (s_1^{(2)}, s_2^{(2)})$:

$$\min(7+7, 6+9) = 14 \Rightarrow s_1^{(2)} = r_1^2 \to r_1^3 \to r_1^4, f^1(x_2) = 14$$

$$\min(6+11, 6+8, 7+9) = 14 \Rightarrow s_2^{(2)} = r_2^2 \to r_2^3 \to r_1^4, f^2(x_2) = 14$$

ЭТАП 3



Формируем вектор состояний $x_3 = (s_1^{(3)})$:

$\min(14+3, 14+2) = 16 \Rightarrow s_1^{(2)} = r_1^1 \to r_2^2 \to r_2^3 \to r_1^4, f^1(x_2) = 16$.

Поскольку мы оказались в начальной вершине, получено оптимальное решение задачи: получен кратчайший маршрут $r_1^1 \to r_2^2 \to r_2^3 \to r_1^4$, длина которого равна 16.

Оптимальное решение, получаемое с помощью динамического программирования, разумеется, не зависит от того, прямым или обратным ходом решалась задача.

2.7.4. Задача о замене

Задача о замене является практическим примером использования методов динамического программирования. Хотя её можно свести к задаче нахождения кратчайшего маршрута в графе, рекуррентная зависимость у этой задачи несколько более интересна, и мы рассмотрим её отдельно.

Рассмотрим предприятие, которое использует некоторое оборудование в процессе своей производственной деятельности. В начале каждого года решается вопрос о замене этого оборудования на новое. В первое время такая замена нецелесообразна, поскольку требует больших инвестиций, а оборудование является относительно свежим. Однако со временем издержки на обслуживание оборудования растут (поскольку оно стареет и чаще ломается), а доходы от использования оборудования снижаются (поскольку увеличиваются простои, связанные с ремонтом, а также, возможно, снижается производительность оборудования из-за старения). В какой-то момент оказывается выгоднее заменить оборудование вместо того, чтобы эксплуатировать старое.

Пусть максимальный срок службы оборудования (после которого оно заменяется в обязательном порядке) составляет n лет. Пусть также период планирования составляет m лет. Также пусть планирование начинается в начале года, когда возраст оборудования составляет a_0 лет. Пусть r_t – годовой доход, получаемый от эксплуатации оборудования возраста t, c_t – затраты на обслуживание оборудования возраста t, s_t – остаточная стоимость оборудования возрастом t лет, I – затраты на замену оборудования. Необходимо определить план замен оборудования, приносящий максимальный доход. Доход должен учитывать остаточную стоимость оборудования по окончании последнего года планирования (будем считать, что мы его продаем).

В процессе решения задачи на каждом этапе достраивается фрагмент дерева альтернатив, приведенный на рис. 2.34. Дерево представляет собой все возможные варианты возрастов оборудования на всех годах планирования. Ветвление дерева производится исходя из альтернатив замены (З) или использования (И) оборудования, с учетом того, что если его возраст составляет n лет, то оно может быть только заменено.

Исходя из условий задачи и рис. 2.34 очевидно, что на каждом этапе необходимо хранить в состоянии системы все возможные альтернативные сроки службы оборудования, а не только срок, получающийся из оптимального плана замен оборудования в предыдущих периодах.

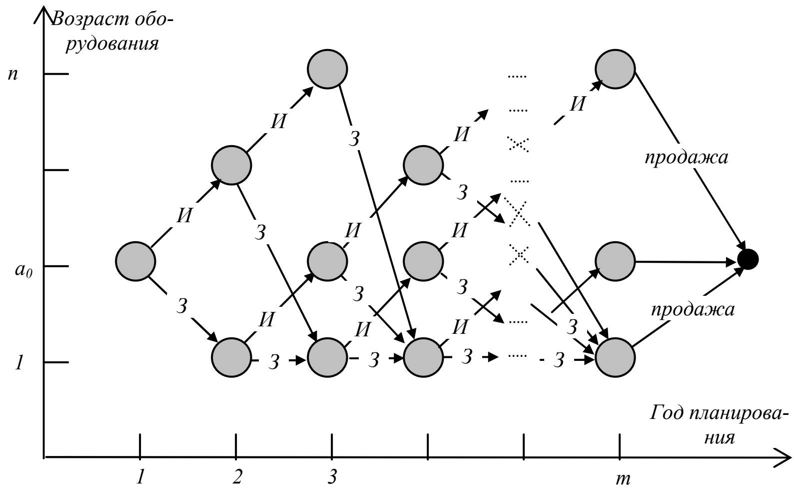


Рис.2.34. Дерево альтернатив

Формализуем основные принципы динамического программирования.

1. Этапом решения задачи k будем считать процедуру анализа k-го планового года.

2. Альтернативами на первом году планирования являются два возраста оборудования на начало года, исходя из двух вариантов: замены оборудования на новое, либо эксплуатации старого оборудования – 0 и a_0 соответственно. Альтернативами на этапе k являются все возможные действия с оборудованием (замена или использование), находящемся во всех возможных возрастах на начало года (дуги на рис. 2.34). Таким образом, альтернативы на разных этапах также рекуррентно связаны между собой.

3. Состояние системы определяется способом, аналогичным уже описанному в задаче нахождения кратчайшего маршрута. Отличие заключается в том, что в предыдущей задаче мы имели уже построенный граф в условии, а в данной задаче граф строится в процессе работы самого оптимизационного алгоритма, поэтому информации о плане замен (маршруте на рис. 2.34), наиболее прибыльно приводящего оборудование к данному возрасту на данном году планирования, недостаточно – необходимо знать также сам возраст оборудования, соответствующий этому плану (иначе придется делать лишние вычисления). Поэтому в элементе вектора состояния будем хранить пару: самый доходный план замен для достижения данного возраста оборудования, и сам возраст. Также совместим этот пункт с принципом вычисления числовой меры альтернатив. Будем считать что $x_k = \{x_{ki}\}, |\mathrm{x}_{\mathrm{ki}}|$ – возраст оборудования.

Состояние системы на первом этапе – массив $x_1=\{<\{З\},0>,<\{И\},a_0>\}$ – два возможных плана – замена оборудования (тогда его возраст на начало года равен 0), либо эксплуатация (возраст – a_0). Целевая функция этих альтернатив вычисляется по формулам:

- Для замены оборудования $f_З = r_0 - c_0 - I + s_{a_0}$.
- Для эксплуатации оборудования $f_И = r_{a_0} - c_{a_0}$.

Также учтем, что если $a_0=n$, то возможна только замена оборудования.

Состояние системы на этапе k $(k<m)$ вычисляется согласно следующему алгоритму.

<u>Шаг 1</u>. Положить $i=1$. Положить $x_k=\{\}$.

<u>Шаг 2</u>. Выбрать элемент i массива x_{k-1}. Пусть это будет элемент $<\{A_1..A_{k-1}\},y>$, где $A_j \in \{З, И\}$, y – возраст оборудования на начало года $k-1$ в случае плана замен $\{A_1..A_{k-1}\}$. Рассмотрим два случая: замену оборудования или его эксплуатацию.

<u>Шаг 3</u>. При замене оборудования его возраст на начало года k составит 0 лет, доход от такого варианта равен $f_З = f^i(x_{k-1}) + r_0 - c_0 - I + s_{y+1}$. Далее возможны два варианта:

- x_k не содержит $x_{kj} : \left|\mathrm{x_{kj}}\right| = 0$. Добавляем в x_k новый элемент $<\{A_1...A_{k-1},З\},0>$. Запомним величину $f_З$, соответствующую этому плану.
- Элемент с нулевым возрастом оборудования уже содержится в x_k ($\exists x_{kj} : \left|\mathrm{x_{kj}}\right| = 0$). В таком случае сравним запомненную для этого элемента величину дохода $f_З$ с вычисленной только что. Если вычисленное значение дохода превышает уже имеющееся, то заменим элемент x_{kj} на $<\{A_1,...,A_{k-1},З\},0>$.

<u>Шаг 4</u>. Если $y<n$, то можно эксплуатировать оборудование. В таком случае его возраст на начало года k составит $y+1$ лет, и доход от такой альтернативы будет равен $f_И = f^i(x_{k-1}) + r_{y+1} - c_{y+1}$. Добавим в x_k новый элемент $<\{A_1,...,A_{k-1},И\},y+1>$. Запомним величину $f_И$, соответствующую этому плану.

<u>Шаг 5</u>. Если i – последний элемент массива x_{k-1}, то конец – рассмотрены все альтернативы. В противном случае увеличиваем i на 1 и переходим к шагу 2.

Общая схема алгоритма выглядит следующим образом.

<u>Шаг 1</u>. Положить $k=1$.

<u>Шаг 2</u>. Вычислить состояние системы x_k и доходы, соответствующие находящимся в нем планам замен $f^i(x_k)$.

<u>Шаг 3</u>. Если $k=m-1$, то перейти к шагу 4. В противном случае положить $k=k+1$ и выполнить шаг 2.

<u>Шаг 4</u>. В последнем состоянии x_{m-1} содержатся все возможные возрасты оборудования на начало года $m-1$, а также планы замен, приводящие к этим возрастам и приносящие максимальные доходы $f^i(x_{m-1})$. Из массива x_{m-1} выбрать такую пару $x_{ml} = \left\langle \left\{ A_1^l, ..., A_{m-1}^l \right\}, y_l \right\rangle$, чтобы $f^l(x_m) + s_{y_l+1} = \max_i \{ f^i(x_m) + s_{y_i+1} \}$. План замен $\left\{ A_1^l, ..., A_{m-1}^l \right\}$ является планом максимального дохода, а возраст оборудования на год m составит $y_l + 1$.

Пример 2.32

Период планирования составляет 5 лет, в начале пятого года оборудование продается. Предельный срок эксплуатации оборудования равен 5 годам, стоимость замены оборудования – 10 млн руб. Возраст оборудования на первом году планирования равен 2 годам. Остальные необходимые величины приведены в таблице.

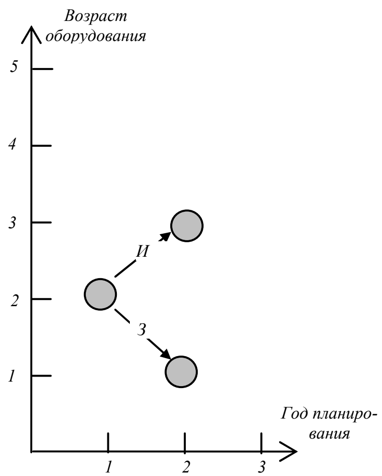
Возраст	Доход, тыс. руб.	Стоимость обсл., тыс. руб.	Остаточная стоимость, тыс.руб.
0	3000	300	
1	2800	500	7000
2	2600	700	6000
3	2500	1000	4000
4	2300	1300	2500
5			1500

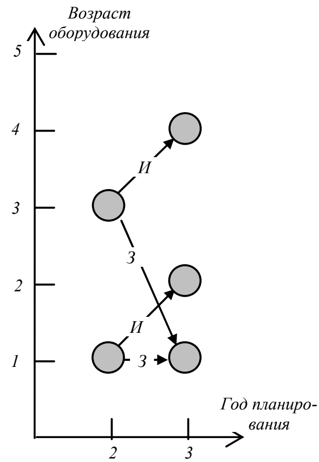
ЭТАП 1

Формируем состояние системы. Есть две альтернативы:

<{З},0>;$f1(x_1)$=6000+3000-300--10000=-1300.

<{И},2>;$f2(x_1)$=2600-700=1900.





ЭТАП 2

Формируем состояние системы. Всего возможны 4 альтернативы:

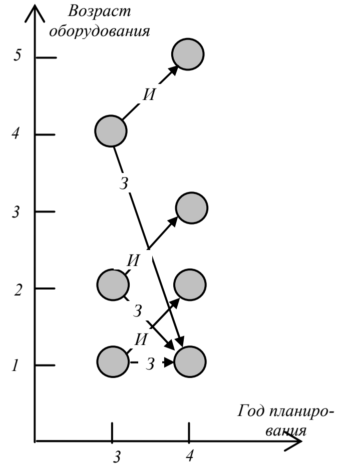
<{З,З},0>;*f*=-1300+7000+3000-
-300-10000=-1600.

<{И,З},0>;*f*=1900+3000+4000-
-300-10000=-1400.

<{З,И},1>;*f*= -1300+ 2800 -
-500=1000.

<{И,И},3>;*f*= 1900+ 2500 -
-1000=3400.

Первую альтернативу отбрасываем, поскольку она хуже второй, а приводит к тому же возрасту оборудования.



ЭТАП 3

6 альтернатив:

<{И,З,З},0>*f*=-1400+7000+3000-
-300 -10000= -1700.

<{И,З,И},1>*f*=-1400+2800-
-500=900.

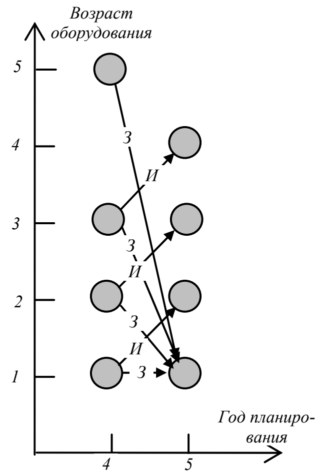
<{З,И,З},0> *f*=1000+6000+
3000-300-10000=-300.

<{З,И,И},2> *f*=1000+2600-
-700=2900.

<{И,И,З},0> *f*=3400+2500+3000-
-300-10000=-1400.

<{И,И,И},4>*f*=3400+2300-
-1300=4400.

Альтернативы 1 и 5 отбрасываем.



ЭТАП 4

7 альтернатив:

<{И,З,И,З},0>f=900+3000+6000--300-10000=-400.

<{И,З,И,И},2>f=900+2600--700=2800.

<{З,И,З,З},0>f=-300+7000++3000 -300-10000=-600.

<{З,И,З,И},1>f=-300+2800--500=2000.

<{З,И,И,З},0> f=2900+3000+4000-300--10000=-400.

<{З,И,И,И},3> f=2900+2500--1000=4400.

<{И,И,И,З},0> f=4400+1500++3000-10000-300=-1400.

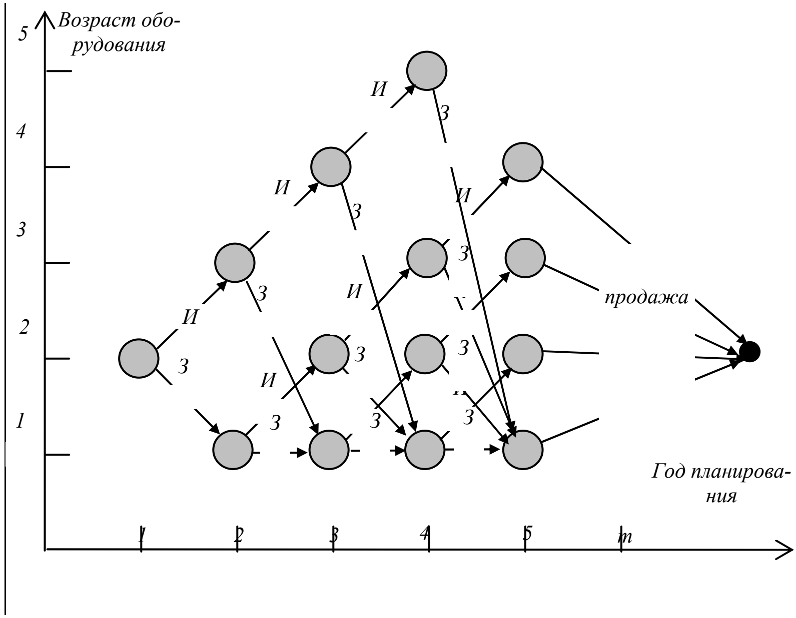
Альтернативы 3,5,7 отбрасываем.

В результате работы алгоритма на 6-м году планирования осталось 5 альтернативы:

Возраст	Альтернатива	Доход	Продажа	Итого
1	И,З,И,З	-400	7000	6600
2	З,И,З,И	2000	6000	8000
3	И,З,И,И	2800	4000	6800
4	З,И,И,И	4400	2500	6900

Оптимальный план задачи, приносящий максимальный доход 8 млн руб. – З,И,З,И.

Общее дерево вариантов в этой задаче приведено на следующем рисунке:



2.7.5. Задача оптимального распределения капитальных вложений на реконструкцию группы предприятий

Пусть существует n предприятий, на реконструкцию которых отведена сумма S. Для каждого предприятия известна зависимость $d_i(y)$ прироста его производительности от суммы, вложенной в его реконструкцию. Данная зависимость может быть нелинейной. Необходимо распределить сумму S таким образом, чтобы суммарный прирост производительности был максимальным. Будем предполагать, что распределяемые суммы – целые числа. Формализуем основные принципы динамического программирования.

1. Этапом k решения задачи будем считать анализ k-го предприятия.
2. Под альтернативами на этапе k будем понимать варианты распределния сумм между k-м предприятием и первыми $k-1$ предприятиями в совокупности.
3. Состоянием системы на первом этапе будет массив из S-элементов $x_1=\{ y_j\}$, где $y_j=j, j=1..S$. Под состоянием системы на этапе k будем понимать массив векторов $x_k=\{v_j\}$ $j=1..S$, где $v_j=(s_1,s_2,..s_k)$, s_i – сумма, отводимая на реконструкцию предприятия

i. Фактически v_j – это план оптимального распределения суммы j между первыми k предприятиями.

4. Числовая мера i-й альтернативы равна $d_k(s_j) + f^{i-s_j}(x_{k-1})$, где $f^{i-s_j}(x_{k-1})$ – прирост производительности при инвестировании суммы $i-s_j$ в первые k-1 предпрятия. Поскольку целью является максимизации прироста производительности, то в i-ю позицию состояния x_k будет попадать альтернатива с максимальным приростом:

$$f^i(x_k) = \max_{\substack{j=1..S \\ s_j<i}} (d_k(s_j) + f^{i-s_j}(x_{k-1})).$$

Рассмотрим работу алгоритма на примере.

<u>Пример 2.33</u>

Рассмотрим проект реконструкции 4-х энергоблоков атомной электростанции. Общая сумма капитальных вложений равна 70 млрд руб., при этом суммы, выделяемые на реконструкцию должны быть кратны 10 млрд руб. Значения функций $f_j(x_j)$ приведены в таблице:

y	0	10	20	30	40	50	60	70
$d_1(y)$	0	30	45	58	68	75	80	83
$d_2(y)$	0	35	55	71	83	93	98	101
$d_3(y)$	0	28	53	71	74	84	89	92
$d_4(y)$	0	60	81	90	97	102	104	105

ЭТАП 1. Формируем альтернативы – всего 7

№ альтернативы	Сумма на реконструкцию первого энергоблока, млн руб.	Прирост производительности	Состояние (v_j)
0	0	0	(0)
1	10	30	(100)
2	20	45	(200)
3	30	58	(300)
4	40	68	(400)
5	50	75	(500)
6	60	80	(600)
7	70	83	(700)

ЭТАП 2. Формируем альтернативы

№ альтернативы	Сумма на реконструкцию 1-го и 2-го энергоблоков	Прирост производительности	Состояние
0	0	0	(0,0)
1	10	max(0+30,**35+0**)=35	(0,10)
2	20	max(0+45,**35+30**,55+0)=65	(10,10)
3	30	max(0+58,35+45,**55+30**, 7+0)=85	(10,20)
4	40	max(0+68,35+58,55+45, **71+30**,83+0)=101	(10,30)
5	50	max(0+75,35+68,55+58, **71+45**,83+30,93+0)=116	(20,30)
6	60	max(0+80,35+75,55+68, **71+58**,83+45,93+30, 98+0)=129	(30,30)
7	70	max(0+83,35+80,55+75, 71+68,**83+58**,93+45, 98+30,101+0)=141	(30,40)

ЭТАП 3. Формируем альтернативы

№ альтернативы	Сумма на реконструкцию 1-го, 2-го и 3-го энергоблоков	Прирост производительности	Состояние
0	0	0	(0,0,0)
1	10	max(**0+35**,28+0)=35	(0,10,0)
2	20	max(**0+65**,28+35,53+0)=65	(10,10,0)
3	30	max(0+85,**28+65**,53+35, 71+0)=93	(10,10,10)
4	40	max(0+101,*28+85*,**53+65**, 71+35,74+0)=118	(10,10,20)
5	50	max(0+116,28+101,**53+85**, 71+65,74+35,84+0)=138	(10,20,20)
6	60	max(0+129,28+116,53+101, **71+85**,74+65,84+35, 89+0)=156	(10,20,30)
7	70	max(0+141,28+129,53+116, **71+101**,74+85,84+65, 89+35,92+0)=172	(10,30,30)

ЭТАП 4. Формируем альтернативы

№ альтернативы	Сумма на реконструкцию всех энергоблоков	Прирост производительности	Состояние
0	0	0	(0,0,0,0)
1	10	max(0+35,**60+0**)=60	(0,0,0,10)
2	20	max(0+65,**60+35**,81+0)=95	(0,10,0,10)
3	30	max(0+93,**60+65**,81+35, 90+0)=125	(10,10,0,10)
4	40	max(0+118,**60+93**,81+65, 90+35,97+0)=153	(10,10,10,10)
5	50	max(0+138,**60+118**,81+93, 90+65,97+35,102+0)=178	(10,10,20,10)
6	60	max(0+156,60+138,**81+118**, 90+93,97+65,102+35, 104+0)=199	(10,10,20,20)
7	70	max(0+172,60+156,**81+138**, 90+118,97+93,102+65, 104+35,105+0)=219	(10,20,20,20)

На последнем этапе альтернатива с максимальным значением целевой функции является оптимальным планом распределения средств. Сам план хранится в соответствующем этой альтернативе элементе v_j состояния. В нашем случае это v_7=(10,20,20,20), максимальный прирост производительности – 219.

Контрольные вопросы и задачи к разделу 2.7

1. Сравните алгоритм динамического программирования с циклической конструкцией любого языка программирования ЭВМ. Можно ли любую программу, содержащую цикл, относить к классу динамического программирования (если рассматривать цикл, как этап)?

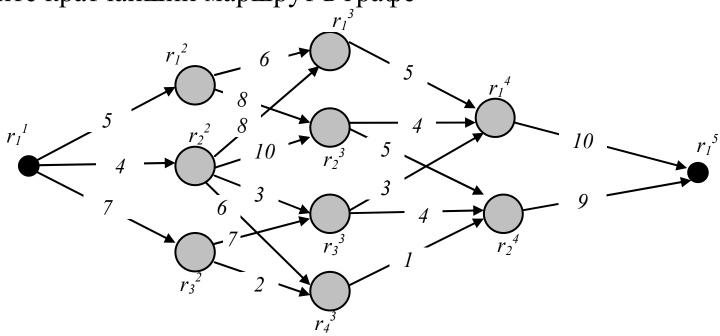
2. Сформулируйте алгоритм динамического программирования для случая существования цикла в графе, для случая наличия дуг между вершинами несмежных слоев.

3. Какие принципы динамического программирования гарантируют работоспособность алгоритма обратного хода динамического программирования?

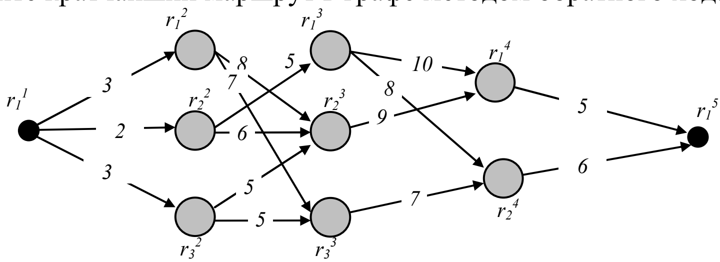
4. Может ли измениться оптимальное решение задачи о замене, если увеличить период планирования, оставив все исходные условия без изменения (речь идет о той части оптимального решения, которая хранит план замен за исходное количество лет)? Существует ли устойчивое решение в предельном случае, когда период планирования не ограничен?

5. За счет чего обеспечивается корректная работа алгоритма динамического программирования в задаче оптимизации распределения капитальных вложений в условиях нелинейной зависимости прироста производительности от суммы вложенных средств?

6. Найдите кратчайший маршрут в графе



7. Найдите кратчайший маршрут в графе методом обратного хода



8. Решите задачу о замене для четырехлетнего периода планирования. Предельный срок эксплуатации оборудования равен 5 годам, стоимость замены оборудования – 100 млн руб. Возраст оборудования на первом году планирования равен 3 годам. Остальные величины приведены в таблице.

Возраст	Доход, тыс руб.	Стоимость обсл., тыс руб.	Остаточная стоим., тыс руб.
0	27000	3000	
1	24000	3500	70000
2	22000	4000	60000
3	21000	5000	55000
4	20500	6500	50000
5			48000

9. Решите задачу о замене для пятилетнего периода планирования. Предельный срок эксплуатации оборудования равен 5 годам, стоимость замены оборудования – 25 млн руб. Возраст оборудования на первом году планирования равен 4 годам. Остальные величины приведены в таблице.

Возраст	Доход, тыс руб	Стоимость обсл., тыс руб.	Остаточная стоим., тыс руб.
0	2000	200	
1	1800	300	23000
2	1600	400	22500
3	1300	600	21000
4	1000	800	19000
5			17000

10. Решите задачу оптимального распеределения капитальных вложений на реконструкцию 5 предприятий. Общая сумма капитальных вложений равна 50 млн руб., при этом суммы, выделяемые на реконструкцию, должны быть кратны 10 млн руб. Значения функций $f_j(x_j)$ приведены в таблице:

y	0	10	20	30	40	50
$d_1(y)$	0	10	15	18	20	21
$d_2(y)$	0	18	25	30	33	34
$d_3(y)$	0	15	20	22	23	24
$d_4(y)$	0	18	26	28	32	34
$d_5(y)$	0	17	23	27	30	32

Ответы на задачи

Раздел 1.2

Задача 6

$x_1 = 14, x_2 = 18, Z = 278$.

Задача 7

Исходная задача:

$$x_1 + 2x_2 \to \max$$
$$x_1 + x_2 = 2$$
$$x_2 + x_3 - x_4 = 1$$
$$x_j >= 0 (j = 1..4)$$

.

Оптимальное решение:

$x_2 = 2, x_4 = 1, Z = 4$.

Базисная матрица

$$B^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}.$$

Задача 8

Решение оптимальное при $y \le 2; x \ge 2$. Целевая функция не ограничена сверху при $x \le 0$. Во всех остальных случаях решение следует продолжить.

Задача 9

Задача не имеет допустимых решений.

Задача 10

Оптимальное решение: $x_1 = 16, x_2 = 2, x_3 = 0, Z = -24$.

Раздел 1.3

Задача 6

Двойственная задача:

$$10y_1 + 6y_2 + 15y_3 \to \max$$
$$y_1 - y_2 + 4y_3 \le 1$$
$$y_1 + 3y_2 - y_3 = -2$$
$$2y_1 + 2y_2 - 3y_3 \le 4$$
$$y_1 + 5y_2 + 8y_3 \le -1$$
$$y_1 >= 0, y_2 - \textit{не ограничен в знаке}, y_4 \le 0$$

.

Задача 7

Оптимальное решение двойственной задачи $(0.6, 1.2, 0, 0); Z = 198$.

Задача 8

Исходная задача:

$$-x_1 - 18x_2 + x_3 + 2x_4 \to \max$$
$$-x_1 + 2x_2 + x_3 = -2$$
$$2x_1 - 8x_2 + x_4 = -1$$
$$x_j >= 0 (j = 1..4)$$

Оптимальное решение:

$$x_1 = \frac{9}{2}, x_2 = \frac{5}{4}, x_3 = x_4 = 0,$$
$$Z = -27$$

Обратная базисная матрица

$$B^{-1} = \begin{pmatrix} -2 & -1/2 \\ -1/2 & -1/4 \end{pmatrix}$$

Задача 9

а) Решение оптимальное при $x \geq 0, y \geq -10$.

б) Нет допустимых решений при $x < 0, y \geq 0$.

в) Решение следует продолжить при $x < 0, -10 \leq y < 0$.

г) Целевая функция не ограничена сверху при $x \geq 0, y < -10$.

д) Во всех остальных случаях в симплекс-таблице записано ни опорное решение, ни пседоплан, решение продолжить нельзя.

Задача 10

Оптимальное решение $x_1 = 18, x_2 = 16, Z = 258$.

Раздел 1.4

Задача 6

Оптимальное решение $x_1 = 15, x_2 = 16, Z = 295$.

Задача 7

Оптимальное решение прямой задачи $x = (0,0,1,2,0,0), Z = -500$.

Оптимальное решение двойственной задачи $y = (-60,-110), Z = -500$.

Задача 8

Оптимальное решение $x = (0,0,5/3,0), Z = 5$.

Задача 9

c_3	Оптимальное решение				$Z_0(c_3)$
	x_1	x_2	x_3	x_4	
$0 \leq c_3$	*5*	*0*	*0*	*10*	*20*
$0 \leq c_3 \leq 4/3$	*0*	*0*	*5*	*20*	*5 c_3+20*
$c_3 > 4/3$	Целевая функция не ограничена сверху				

Задача 10

a_1	Оптимальное решение				$Z_0(a_1)$
	x_1	x_2	x_3	x_4	
$0 \le a_1$	0	$-a_1/3$	$-2/3\, a_1 + 20$	0	$2/3\, a_1 + 20$
$0 \le a_1 \le 10$	a_1	0	0	$20-2a_1$	20
$a_1 \ge 10$	10	0	a_1-10	0	$50-3a_1$

Раздел 1.5

Задача 3

Оптимальное решение $x_1 = 9, x_2 = 5, Z = 66$.

Задача 4

Оптимальное решение $x_1 = 0, x_2 = 9/5, Z = 23/9$.

Раздел 1.6

Задача 3

Необходимые условия для существования экстремума:

$$3x_1^2 + x_2 + e^{x_1} + 2\lambda_1 x_1 + 2\lambda_2 x_1 x_2^2 = 0$$

$$2x_2 + x_1 + 2\lambda_1 x_2 + 2\lambda_1 x_2 x_3^2 + 2\lambda_2 x_1^2 x_2 + \lambda_2 e^{x_2} = 0$$

$$1 + 2\lambda_1 x_2^2 x_3 + 2\lambda_2 = 0$$

$$x_1^2 + x_2^2 + x_2^2 x_3^2 - 25 = 0$$

$$x_1^2 x_2^2 + e^{x_2} + 2x_3 - 5 = 0$$

Задача 4

Квадратичная форма отрицательно определенная.

Задача 5

Квадратичная форма неопределенная.

Раздел 2.1

Задача 5

$$60000x_{11} + 60000x_{12} + 65000x_{21} + 65000x_{22} + 80000x_{31} + 80000x_{32} - 2z_1 - 6000000y \to \max$$

$$1000x_{11} + 800x_{21} + 700x_{31} + 950x_{12} + 600x_{22} + 400x_{32} - z_1 = 0$$

$$2x_{11} + 2.3x_{21} + 2.7x_{31} + 2.1x_{12} + 2.2x_{22} + 2.5x_{32} - z_2 = 0$$

$$500y - z_2 = 0$$

$$y \le 10$$

$$x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, z_1, z_2, y \ge 0$$

$$y - \text{целые}$$

.

Задача 6

$$2y_1^1+5y_2^1+7y_3^1+3x_2-3y_2^2-4y_3^2\to \max$$

$$2y_1^1+5y_2^1+7y_3^1+x_2+3y_2^2+4y_3^2\le 10$$

$$x_2\le 5$$

$$2y_1^1+5y_2^1+7y_3^1+x_2-3y_2^2-4y_3^2\ge 2$$

$$y_1^1+y_2^1+y_3^1=1$$

$$y_1^2+y_2^2+y_3^2=1$$

$$x_2\ge 0;\, y_i^j\in\{0,1\}$$

Задача 7

$$\sin(1)y_1^1+\sin(3)y_2^1+\sin(5)y_3^1+y_1^2+8y_2^2+27y_3^2\to \max$$

$$y_1^1+3y_2^1+5y_3^1-y_1^2-2y_2^2-3y_3^2\le 3$$

$$y_1^1+3y_2^1+5y_3^1\le 8$$

$$y_1^1+y_2^1+y_3^1=1$$

$$y_1^2+y_2^2+y_3^2=1$$

$$y_i^j\in\{0,1\}$$

Задача 8

$$2x_{12}+5x_{13}+3x_{21}+2x_{23}+8x_{31}+x_{32}\to \max$$

$$x_{12}+x_{13}=1$$

$$x_{21}+x_{23}=1$$

$$x_{31}+x_{32}=1$$

$$x_{21}+x_{31}=1$$

$$x_{12}+x_{32}=1$$

$$x_{13}+x_{23}=1$$

$$u_2-u_3+2x_{23}\le 1$$

$$u_3-u_2+2x_{32}\le 1$$

$$x_{ij}\in\{0,1\};u_1,u_2-\textit{не ограничены в знаке}$$

Задача 9

Оптимальное решение задачи без требования целочисленности:

$x=(5.5,4.5), Z=10$.

Оптимальное решение задачи с целочисленными переменными:

$x=(5,3), Z=8$.

Раздел 2.2

Задача 6

При решении задачи одним методом возможны разные ответы, в зависимости от выбора вычеркиваемых строк и столбцов. Здесь приведен один из возможных ответов. Указаны только значения базисных переменных.

а)МСЗУ: $x_{11}=50, x_{12}=5, x_{22}=35, x_{23}=5, x_{33}=40, x_{34}=40; Z=2025$

МНС: $x_{12}=10, x_{13}=45, x_{24}=40, x_{31}=50, x_{32}=30, x_{34}=0; Z=1520$

МФ: $x_{12}=10, x_{13}=45, x_{22}=0, x_{24}=40, x_{31}=50, x_{32}=30; Z=1520$

МПЗ: $x_{12}=10, x_{13}=45, x_{22}=30, x_{24}=10, x_{31}=50, x_{34}=30; Z=1490$.

б) МСЗУ: $x_{11}=45, x_{21}=10, x_{22}=40, x_{32}=15, x_{33}=45, x_{34}=55; Z=2160$

МНС: $x_{12}=45, x_{23}=45, x_{24}=10, x_{31}=55, x_{32}=10, x_{34}=45; Z=1670$

МФ и МПЗ: $x_{12}=45, x_{24}=50, x_{31}=55, x_{32}=10, x_{33}=45, x_{34}=5; Z=1645$.

в) МСЗУ: $x_{11}=60, x_{12}=0, x_{22}=45, x_{32}=15, x_{33}=55, x_{34}=45; Z=2750$

МНС, МФ и МПЗ: $x_{12}=60, x_{21}=45, x_{31}=15, x_{32}=0, x_{33}=55, x_{34}=45; Z=2420$.

Задача 7

Оптимальное решение $x_{13}=5, x_{14}=50, x_{21}=40, x_{23}=20, x_{32}=40, x_{34}=20; Z=1630$.

Задача 8

Оптимальное решение $x_{11}=15, x_{14}=45, x_{21}=45, x_{31}=0, x_{32}=60, x_{33}=55; Z=1560$.

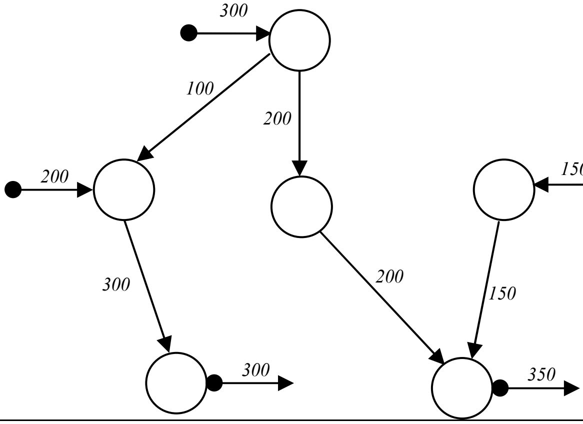
Задача 9

Закрытая транспортная модель

	1	1’	2	3	4	
1	11	11	11	M	7	10
2	9	9	10	10	M	30
3	16	M	15	15	10	105
	5	55	10	50	25	

Задача 10

Оптимизированный граф перевозок:



Раздел 2.3

Задача 6

Аналитическое представление выпуклой оболочки: $\begin{matrix} x_1 \le 5 \\ x_2 \le 4 \end{matrix}$.

Задача 7

Единственное правильное отсечение $x_1 + x_2 \le 8$.

Задача 8

Оптимальное решение $x_1 = 4, x_2 = 3, Z = 34$.

Задача 9

Оптимальное решение $x_1 = 7, x_2 = 4, Z = 11$.

Раздел 2.4

Задача 6

Оптимальное решение $x_1 = 3, x_2 = 4, Z = 11$.

Задача 7

Задача ветвится на две – одна не имеет допустимых решений (штраф сверху вычислить нельзя), вторая получается введением дополнительного ограничения $-x_1 - 0.5x_2 + x_5 = -0.5$.

Задача 8

Оптимальный маршрут: $1 \to 3 \to 4 \to 2 \to 1, Z = 7$.

Задача 8

Оптимальный маршрут: $1 \to 5 \to 3 \to 2 \to 4 \to 1, Z = 7$.

Раздел 2.5

Задача 6

Приближенное решение (совпадает с оптимальным) $x_1 = 1, x_2 = 2, Z = 4$.

Задача 7

Приближенное решение $x_1 = 0, x_2 = 2, Z = 82$.

Оптимальное решение $x_1 = 1, x_2 = 1, Z = 111$.

Задача 8

Приближенное решение задачи коммивояжера: $1 \to 2 \to 4 \to 3 \to 5 \to 1, Z = 18$.

Задача 9

Приближенное решение задачи коммивояжера: $1 \to 4 \to 2 \to 3 \to 1, Z = 6$.

Раздел 2.6

Задача 6

$x_1 + y_2 + 2x_3 + y_4 \to \max$

$x_1 + 2y_2 + x_3 \le 7$

$x_1' + y_2 + 2x_3 + 3y_4 \le 2$

$x_1 + 3y_2' + x_3 + 4y_4 \le 6$

$x_1, x_1', y_2, y_2', x_3, y_4 \in \{0,1\}$

Задача 7

Допустимое решение задачи $x_1 = 1, x_2 = 1, x_3 = 0; Z = 3$.

Задача 8

Оптимальное решение задачи $x_1 = 1, x_2 = 0, x_3 = 0; Z = 2$.

Задача 9

Оптимальное решение задачи $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1; Z = 4$.

Задача 10

$$\sum_{j=1}^{4} c_j x_j \to \min$$

$$\sum_{j=1}^{4} m_j x_j \le M$$

$$\sum_{j=1}^{4} x_j \ge 3$$

$x_4 \le x_2$

$x_1 + x_2 \ge 2x_5$

$x_1 + x_2 - 1 \le x_5$

$x_5 + x_3 \le 1$

Раздел 2.7

Задача 6

Кратчайший маршрут $r_1^1 \to r_3^2 \to r_4^3 \to r_2^4 \to r_1^5, Z = 19$.

Задача 7

Кратчайший маршрут $r_1^1 \to r_3^2 \to r_3^3 \to r_2^4 \to r_1^5, Z = 21$.

Задача 8

Оптимальный план замен – $\{З,И,И\}$, суммарный доход на начало 5-го года 72.5 млн руб.

Задача 9

Оптимальный план замен – $\{З,И,З,И\}$, суммарный доход на начало 6-го года 20.6 млн руб.

Задача 10

Оптимальный план распределения – (10,10,10,10,10), ожидаемый доход - 78

Список использованной литературы

1. Загребаев А.М., Крицына Н.А., Кулябичев Ю.П., Шумилов Ю.Ю. Методы математического программирования в задачах оптимизации сложных технических систем.– М.:МИФИ,2008.

2. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование.–М.:Наука, 1969.

3. Мостеллер Ф., Тьюки Дж. Анализ данных и регрессия.- М:Финансы и статистика, 1982.

4. Окунев В.С. Использование идей и методов решения многокритериальных задач для оптимизации состава сплава Na-K-Cs/Научная сессия МИФИ - 2001. Сборник научных трудов. - М., 2001.– Т.8: Молекулярно-селективные и нелинейные явления и процессы. Физико-технические проблемы нетрадиционной энергетики. Физико-технические проблемы ядерной энергетики. – С. 113-114.

5. Переславцев П.Э., Сахраи Д. Разработка оптимизационной модели ядерного центра. – Обнинск: Известия ВУЗов. Ядерная энергетика, № 2, 2002 г.

6. Рудик А.П. Оптимизация физических характеристик ядерных реакторов.–М: Атомиздат, 1979.

7. Салмин И.Д. Математические методы решения оптимизационных задач.–М.: МИФИ, 2004.

8. Салмин И.Д. Математическое программирование, ч.2.– М:МИФИ, 1979.

9.Чернавский С.Я. Системное прогнозирование ядерной энергетики.– М: Наука, 1980.

10. Dantzig G.B. Notes on linear programming. Rand Corporation,1953.

Список рекомендованной литературы

1. Акулич И.Л. Математическое программирование в примерах и задачах. Учебное пособие для студентов экономических специальностей ВУЗов.– М.: Высшая школа, 1986.
2. Зайченко Ю.П. Исследование операций. –Киев: Издательский дом «Слово», 2003.
3. Исследование операций. Т.1: Пер. с англ. /Под ред. Дж. Моудера, С. Элмаграби.– М.: Мир, 1981.
4. Карманов В.Г. Математическое программирование. –М.: Издательство физико-математической литературы, 2004.
5. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование.–М.:Наука, 1969.
6. Леонов А.А. Методы решения линейных экстремальных задач. Учебное пособие.– М.: МИФИ, 1984.
7. Максимов Ю.Я. Алгоритмы линейного и дискретного программирования.– М.:МИФИ, 1982.
8. Мину М. Математическое программирование. Теория и алгоритмы: Пер.с фр.– М.: Наука, 1990.
9. Пантелеев А.В., Летова Т.А. Методы оптимизации в примерах и задачах.– М: Высшая школа, 2002.
10. Салмин И.Д. Математические методы решения оптимизационных задач.– М.: МИФИ, 2004.
11.Салмин И.Д.Математическое программирование, Т.2.– М:МИФИ, 1979.
12. Таха Х. Введение в исследование операций/Пер. с англ. – М.: Вильямс, 2005.

Дмитрий Александрович Жолобов

Введение в математическое программирование

Учебное пособие

Редактор Е.Е. Шумакова
Оригинал-макет изготовлен Д.А.Жолобовым

Печать пособия выполнена в соответствии с оригиналом-макетом, представленным автором.

Подписано в печать 22.11.2008. Формат 60x84 1/16.
Печ. л. 23.5.Уч.-изд. л. 23.5. Тираж 150 экз.
Изд. № 1/43. Заказ №

Московский инженерно-физический институт
(государственный университет), 115409, Москва, Каширское ш., д.31.
Типография издательства «ТРОВАНТ»,
г. Троицк Московской области