УЧЕБНОЕ ПОСОБИЕ

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

О. А. Клачкова,
И. А. Кострикин,
Я. А. Рощина

75
Экономический
факультет
МГУ
имени
М.В. Ломоносова

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. Ломоносова
Экономический факультет

О. А. Клачкова, И. А. Кострикин, Я. А. Рощина

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Учебное пособие

Москва
2025

УДК 330.45
ББК 65
К47

Клачкова О. А., Кострикин И. А., Рощина Я. А.

К47 **Исследование операций:** учебное пособие. — М.: Экономический факультет МГУ имени М. В. Ломоносова, 2025. — 152 с. — URL: https://www.econ.msu.ru/elibrary/is/bef/#top

ISBN 978-5-907690-75-2

Учебное пособие содержит материалы первой части курса «Исследование операций», который изучают студенты бакалавриата экономического факультета МГУ. Каждая глава пособия включает основные теоретические положения, необходимые для решения задач, и набор задач для самостоятельного решения. Несколько типовых заданий приведено с подробным решением, снабженным методическими комментариями, «связывающими» его с теорией. Структура пособия позволяет читателю не только отработать навыки решения оптимизационных задач, но и систематизировать теоретические знания, лежащие в их основе.

УДК 330.45
ББК 65

ISBN 978-5-907690-75-2

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

При решении экономических задач результат зачастую может быть получен различными способами. Как правило, требуется найти *наилучший* способ, однако в зависимости от ситуации *наилучшими* могут являться разные решения. Целенаправленная деятельность, заключающаяся в получении наилучших результатов при соответствующих условиях, называется оптимизацией (от лат. optimus — наилучший)[1]. Задача поиска оптимального решения называется задачей оптимизации, а применяемые в процессе решения такой задачи методы — методами оптимизации, или методами оптимальных решений. В большинстве экономических задач понятие «*наилучший*» может быть выражено вполне определенными количественными критериями – минимум затрат или расходов, максимум прибыли и т.д. Для решения таких задач необходима постановка оптимизационной задачи нахождения экстремума целевой функции. Предлагаемое пособие посвящено изучению основных методов решения оптимизационных задач, возникающих при экономическом моделировании. Рассматриваемые методы служат для нахождения экстремальных значений целевой функции среди множества ее возможных значений, определяемых ограничениями. Именно наличие ограничений принципиально отличает рассматриваемые задачи от классических задач математического анализа по отысканию экстремума функции, методы решения которых при наличии ограничений оказываются непригодными.

Структура пособия соответствует курсу «Исследование операций», много лет преподаваемому авторами на экономическом факультете МГУ имени М. В. Ломоносова, в приложении А приведен образец программы данного курса. Пререквизитами для изучения предлагаемого материала являются базовые знания по математическому анализу и линейной алгебре. Пособие состоит из семи глав, посвященных первоначальным разделам курса. Каждая глава содержит основные теоретические положения, необходимые для решения задач соответствующего типа, и набор таких задач для самостоятельного решения. Несколько типовых

[1] Большой академический словарь русского языка. Т. 14. Опора — открыть. СПб.: Наука, 2010. С. 53. ISBN: 978-5-02-037358-7.

заданий приведено с подробным решением, снабженным методическими комментариями, «связывающими» его с теорией. Авторы не ставили целью дать исчерпывающий теоретический материал по рассматриваемым темам; в то же время изложение построено таким образом, чтобы читатель мог совместить получение практических навыков решения оптимизационных задач и систематизацию необходимых для этого теоретических знаний.

В первой главе рассматривается постановка задачи линейного программирования. Описаны основные экономические проблемы, при решении которых возникает необходимость в таком инструментарии. Результатом освоения материала главы является умение формализовать экономическую задачу, то есть описать ее с помощью одной из известных математических моделей, сведя к задаче линейного программирования.

Вторая глава дает представление о геометрии линейного программирования, основанной на понятии выпуклых множеств и на их основных свойствах. Результатом ее освоения является геометрическое представление о допустимом множестве решений задачи линейного программирования, а также понимание того, как геометрически устроен основной метод решения задач линейного программирования – симплексный метод.

Третья глава посвящена симплексному методу решения задачи линейного программирования, результатом изучения материала главы является умение решить любую такую задачу.

В четвертой главе рассматривается теория двойственности, а также экономическая интерпретация ее основных результатов. Результатом освоения материала является умение анализировать полученные количественные результаты и делать выводы по исходно возникшей экономической проблеме.

В пятой главе к задаче линейного программирования в общем виде добавляется условие на целочисленность переменных, то есть рассматриваются дискретные, или целочисленные, задачи линейного программирования. Результатом освоения главы станет умение решать линейные задачи с дополнительным условием на целочисленность с помощью двух принципиально различных методов: метода отсечений и метода упорядоченного перебора (метод ветвей и границ).

В шестой главе обсуждается транспортная задача и задача о назначениях, которые являются задачами линейного программирования в специальной форме. Результатом освоения главы является умение решать подобные задачи при помощи специальных методов, а также формализовать различные содержательные экономические постановки этих задач.

Наконец, седьмая глава посвящена формализации конкретных экономических задач в виде задачи линейного программирования, а также нахождению решений с помощью языка программирования R. Результатом освоения главы является умение решить любую задачу линейного программирования с использованием компьютера.

Представление о связи курса с другими дисциплинами, составляющими ядро образовательной программы экономического факультета, можно получить в (Количественные методы в экономических исследованиях, 2013).

ГЛАВА 1
Задача линейного программирования

В общем виде любая оптимизационная задача состоит в максимизации или минимизации целевой функции при некоторых условиях:

$$\begin{cases} f\left(x_1,\ldots,x_n\right) \to extr \\ g_i\left(x_1,\ldots,x_n\right) \le b_i, \;\; i=1,\ldots,k \\ g_i\left(x_1,\ldots,x_n\right) = b_i, \;\; i=k+1,\ldots,m \end{cases},$$

где $f, g_1,\ldots,g_m$ — заданные функции, а $b_1,\ldots,b_m$ — заданные действительные числа. Если функции $f, g_1,\ldots,g_m$ являются линейными, то соответствующая задача называется задачей линейного программирования (ЗЛП). Если хотя бы одна из указанных функций не является линейной, то соответствующая задача называется задачей нелинейного программирования. Термин «*программирование*» является отражением англоязычного термина «*programming*», хотя более точным его переводом было бы «планирование», поскольку в результате решения задачи линейного программирования находится план (программа) работы. Однако термины «линейное программирование» и «нелинейное программирование» прочно закрепились в русскоязычной литературе.

Из определения следует, что ЗЛП уместно ставить при математическом моделировании процессов, основанных на предпосылке линейности окружающего мира. Рассмотрим основные экономико-математические модели, приводящие к задачам линейного программирования.

Модель планирования производства (с ограниченным горизонтом планирования)

При планировании производства предприятие может ставить разные цели и, в зависимости от этого, использовать различные критерии оптимальности. Далее будет рассмотрена классическая постановка, в которой критерием оптимальности является максимизация прибыли.

Пусть в заданном периоде предприятие выпускает n видов продукции $(P_1,\ldots,P_n)$, используя для этого m видов ресурсов $(S_1,\ldots,S_m)$, запасы i-го из которых составляют b_i единиц, $i=1,\ldots,m$. Затраты ресурса S_i на выпуск единицы продукции вида P_j также известны и задаются коэффициентами a_{ij}. Выпуск единицы продукта j-го вида приносит прибыль c_j, $j=1,\ldots,n$. В рассматриваемом периоде планирования перечисленные параметры модели остаются постоянными. Результатом моделирования служит оптимальный план — необходимый объем производства каждого вида продукции P_j.

Обозначим через x_j количество единиц продукта P_j, производимое в заданном периоде, и запишем условие его неотрицательности для всех видов продуктов:

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0 \tag{1.1}$$

Количество i-го ресурса S_i, имеющееся в распоряжении у предприятия в планируемом периоде, составляет b_i единиц, поэтому можно выписать следующую систему ограничений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \\ \ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \end{cases} \tag{1.2}$$

Представим структуру модели в виде таблицы, содержащей всю информацию о модели (табл. 1.1).

Таблица 1.1

Балансовая структура модели планирования производства

Виды ресурсов	**Виды продукции**				**Ограничения на объемы ресурсов**
	P_1, x_1 ед.	**P_2, x_2 ед.**	**…**	**P_n, x_n ед.**	
S_1	$a_{11}x_1$	$a_{12}x_2$	…	$a_{1n}x_n$	b_1
S_2	$a_{21}x_1$	$a_{22}x_2$	…	$a_{2n}x_n$	b_2
…	…	…	…	…	…
S_m	$a_{m1}x_1$	$a_{m2}x_1$	…	$a_{mn}x_n$	b_m
Прибыль от реализации продукта	c_1x_1	c_2x_2	…	c_nx_n	—

Критерием оптимальности в модели является максимизация прибыли, задаваемой функцией $z = z(x_1, \ldots, x_n)$. Получаем следующую задачу условной оптимизации:

$$\begin{cases} z(x_1, \ldots, x_n) = z_1 x_1 + z_2 x_2 + \ldots + z_n x_n \to max \\ a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \le b_1 \\ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \le b_2 \\ \ldots\ldots\ldots \\ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \le b_m \\ x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0 \end{cases} \tag{1.3}$$

Использование знака суммирования позволяет записать задачу (1.3) более кратко:

$$\begin{cases} \sum_{j=1}^{n} z_j x_j \to max \\ \sum_{j=1}^{n} a_{ij} x_j \le b_i,\ i = 1, \ldots, m \\ x_j \ge 0,\ j = 1, \ldots, n \end{cases} \tag{1.3$'$}$$

Все ограничения производственной задачи использования ресурсов, а также целевая функция являются линейными, поэтому ее математическая постановка приводит к задаче линейного программирования на максимум и может быть решена методами, описанными в главе 3.

Модель планирования рациона (с ограниченным горизонтом планирования)

Модель планирования рациона позволяет из имеющихся в распоряжении продуктов составить минимальный по стоимости рацион питания с соблюдением нормативных значений по количеству питательных веществ. В качестве горизонта планирования, как правило, рассматривают дневной, недельный или двухнедельный периоды для людей, недельный, месячный или годовой периоды для животных (разница вызвана дополнительными требованиями к разнообразию рациона для людей). Пусть в заданном периоде в нашем распоряжении есть n видов продуктов $P_1, \ldots, P_n$, для каждого из которых известно значение их m потребительских характеристик $S_1, S_2, \ldots, S_m$ (например, калорийности, содержания различных веществ и т.д.), а также известна их стоимость за единицу c_j, $j = 1, \ldots, n$. Пусть также заданы предельно допустимые (как правило, минимальные) значения b_i i-й характеристики в рационе, $i = 1, 2, \ldots, m$. Наконец, пусть известны значения a_{ij} характеристики S_i для единицы продукта P_j, $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Обозначим через x_j количество единиц продукта P_j, включаемое в рацион в заданном периоде, и запишем условие его неотрицательности для всех видов продуктов:

$$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0 \tag{1.4}$$

Предельно допустимое (как правило, минимальное) значение i-й характеристики S_i в рационе в заданном периоде составляет b_i единиц, поэтому можно выписать следующую систему ограничений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geq b_2 \\ \ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq b_m \end{cases} \tag{1.5}$$

Представим структуру модели в виде таблицы, содержащей всю информацию о модели (табл. 1.2).

Таблица 1.2

Балансовая структура модели планирования рациона

Потребительские характеристики	**Виды продуктов питания**				**Минимальное допустимое значение характеристики**
	P_1, x_1 ед.	**P_2, x_2 ед.**	**…**	**P_n, x_n ед.**	
S_1	$a_{11}x_1$	$a_{12}x_2$	…	$a_{1n}x_n$	b_1
S_2	$a_{21}x_1$	$a_{22}x_2$	…	$a_{2n}x_n$	b_2
…	…	…	…	…	…
S_m	$a_{m1}x_1$	$a_{m2}x_2$	…	$a_{mn}x_n$	b_m
Стоимость продукта	c_1x_1	c_2x_2	…	c_nx_n	—

Критерием оптимальности в модели является минимизация суммарной стоимости $z=z(x_1,\dots,x_n)$ рациона питания за рассматриваемый период. Получаем задачу условной оптимизации:

$$\begin{cases} z(x_1,\dots,x_n)=z_1x_1+z_2x_2+\dots+z_nx_n\to min \\ a_{11}x_1+a_{12}x_2+\dots+a_{1n}x_n\ge b_1 \\ a_{21}x_1+a_{22}x_2+\dots+a_{2n}x_n\ge b_2 \\ \dots\dots\dots \\ a_{m1}x_1+a_{m2}x_2+\dots+a_{mn}x_n\ge b_m \\ x_1\ge 0, x_2\ge 0,\dots,x_n\ge 0 \end{cases} \tag{1.6}$$

В краткой записи задача (1.6) имеет вид:

$$\begin{cases} \sum_{j=1}^{n} z_j x_j \to min \\ \sum_{j=1}^{n} a_{ij}x_j \ge b_i,\ i=1,\dots,m \\ x_j\ge 0,\ j=1,\dots,n \end{cases} \tag{1.6′}$$

Все ограничения и целевая функция (1.6) являются линейными, таким образом, математическая постановка задачи планирования рациона приводит к задаче линейного программирования на минимум.

Модель планирования перевозок (транспортная задача)

Модель планирования перевозок применяется для нахождения оптимальной схемы перевозок продукции от m различных поставщиков $S_1,S_2,\dots,S_m$ к n различным потребителям $P_1,P_2,\dots,P_n$. Критерием оптимальности может служить минимизация стоимости или минимизация времени перевозки. Далее рассмотрим классическую постановку, в которой минимизируется стоимость и перевозится только один вид однородной продукции. Запасы поставщика S_i составляют a_i единиц, потребности потребителя P_j — b_j единиц продукции, перевозка единицы продукции от поставщика S_i к потребителю P_j стоит c_{ij}. Как и в двух предыдущих моделях, в заданном периоде параметры (запасы поставщиков, потребности потребителей, стоимость перевозки) постоянны. Изменение любого из них означает окончание допустимого периода планирования и приводит к переформулировке модели.

Обозначим через x_{ij} объем перевозки продукции от поставщика S_i к потребителю P_j и запишем условие его неотрицательности:

$$x_{ij} \geq 0, \ \ i = 1,\ldots,m, \ \ j = 1,\ldots,n \tag{1.7}$$

Запасы продукции i-го поставщика S_i составляют a_i единиц, поэтому суммарный объем поставок от него равен a_i. Аналогичные рассуждения верны для j-го потребителя, суммарный объем поставок которому равен его потребностям. Таким образом, можно выписать следующую систему ограничений:

$$\begin{cases} \sum\limits_{j=1}^{n} x_{ij} = a_i, \ i = 1,\ldots,m \\ \sum\limits_{i=1}^{m} x_{ij} = b_j, \ j = 1,\ldots,n \end{cases} \tag{1.8}$$

Представим структуру модели в виде таблицы, содержащей всю информацию о модели (табл. 1.3).

Таблица 1.3

Балансовая структура модели планирования перевозок

Поставщики продукции	**Потребители продукции**				**Запасы продукции у поставщика**
	P_1	P_2	…	P_n	
S_1	$c_{11}x_{11}$	$c_{12}x_{12}$	…	$c_{1n}x_{1n}$	a_1
S_2	$c_{21}x_{21}$	$c_{22}x_{22}$	…	$c_{2n}x_{2n}$	a_2
…	…	…	…	…	…
S_m	$c_{m1}x_{m1}$	$c_{m2}x_{m2}$	…	$c_{mn}x_{mn}$	a_m
Потребности в продукции потребителя	b_1	b_2	…	b_n	—

Критерием оптимальности в модели является минимизация суммарной стоимости $c = c(x_{11},\ldots,x_{mn})$ плана перевозок за рассматриваемый период. Получаем задачу условной оптимизации:

$$\begin{cases} \quad z = \sum_{i=1}^{m}\sum_{j=1}^{n} z_{ij} x_{ij} \to min \\ \quad \sum_{j=1}^{n} x_{ij} = a_i, \; i = 1,\ldots,m \\ \quad \sum_{i=1}^{m} x_{ij} = b_j, \; j = 1,\ldots,n \\ x_{ij} \ge 0, \;\; i = 1,\ldots,m, \;\; j = 1,\ldots,n \end{cases} \tag{1.9}$$

Такая задача разрешима тогда и только тогда, когда суммарная потребность потребителей совпадает с общим объемом запасов поставщиков, то есть когда выполнено равенство

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \tag{1.10}$$

При выполнении данного равенства задача является сбалансированной. В противном случае задача называется несбалансированной и является недопустимой.

Задача (1.9) является ЗЛП на минимум. Помимо решения общим методом, разработанным для ЗЛП, транспортную задачу можно решить существенно более простым методом благодаря особому виду матрицы ограничений. Этот метод основан на теории двойственности и изложен в главе 5.

Больше различных примеров задач можно посмотреть в (Акулич, 2011).

В общем случае (как и в трех рассмотренных выше примерах) любая ЗЛП состоит в максимизации или минимизации линейной целевой функции при линейно заданных ограничениях, которые могут быть равенствами или неравенствами. Кроме того, на ряд переменных может быть наложено условие неотрицательности (такие условия являются частным случаем линейных ограничений в виде неравенств). В зависимости от вида ограничений (только равенства, только неравенства или ограничения обоих видов), а также от наличия условий неотрицательности переменных выделяют три возможные формы записи ЗЛП.

Задача линейного программирования в общей форме

Запись задачи в общей форме дает, как следует из названия, возможность наиболее широкой постановки — могут использоваться ограничения как в виде равенств, так и в виде неравенств, а условие неотрицательности может быть наложено на любое число переменных (в том числе на все переменные или ни на одну из переменных). Такая постановка обладает наибольшей гибкостью и позволяет формализовать все возникающие на практике реальные ситуации, поддающиеся сведению к ЗЛП. Единственным условием, накладываемым на запись в общей форме, является соответствие типа экстремума и знака ограничений вида неравенства. Дадим строгое определение.

Определение. *Задачей линейного программирования в общей форме на максимум* называется задача, состоящая в максимизации функции (1.11) при условиях (1.12)—(1.13):

$$z = \sum_{j=1}^{n} c_j x_j \to max \tag{1.11}$$

$$\begin{aligned} &\sum\nolimits_{j=1}^{n} a_{ij} x_j \le b_i,\ i = 1,\ldots,k \\ &\sum\nolimits_{j=1}^{n} a_{ij} x_j = b_i,\ i = k+1,\ldots,m \end{aligned} \tag{1.12}$$

$$x_j \ge 0,\ j = 1,\ldots,l \tag{1.13}$$

Здесь и далее полагаются выполненными очевидные неравенства $k \le m, l \le n$.

Определение задачи линейного программирования в общей форме на минимум дается аналогично, при этом знак в ограничениях в виде неравенства меняется на противоположный.

Определение. *Задачей линейного программирования в общей форме на минимум* называется задача, состоящая в минимизации функции (1.14) при условиях (1.15)—(1.16):

$$z = \sum_{j=1}^{n} c_j x_j \to min \tag{1.14}$$

$$\begin{aligned} &\sum\nolimits_{j=1}^{n} a_{ij} x_j \ge b_i,\ i = 1,\ldots,k \\ &\sum\nolimits_{j=1}^{n} a_{ij} x_j = b_i,\ i = k+1,\ldots,m \end{aligned} \tag{1.15}$$

$$x_j \ge 0,\ j = 1,\ldots,l \tag{1.16}$$

Определение. Условия (1.13) ((1.16) в задаче на минимум) называются *общими ограничениями ЗЛП*; условия (1.12) ((1.15) в задаче на минимум) называются *специальными ограничениями ЗЛП*. Матрица $A = \left(a_{ij}\right)$ называется *матрицей специальных ограничений ЗЛП*.

Приведенная терминология общих и специальных ограничений используется и для записи ЗЛП в стандартной и канонической форме.

Задача линейного программирования в стандартной форме

Запись задачи линейного программирования в стандартной форме является частным случаем записи в общей форме, при котором все специальные ограничения имеют вид неравенств, а условие неотрицательности наложено на все переменные. Запись в такой форме удобно использовать для формального описания экономико-математических моделей при первичной постановке задачи. Например, в стандартной форме были первоначально записаны все три модели, рассмотренные выше (модель планирования производства, модель планирования рациона и модель планирования перевозок). Дадим строгое определение.

Определение. *Задачей линейного программирования в стандартной форме на максимум* называется частный случай задачи (1.11)—(1.13) в общей форме при $k=m$, $l=n$.

Определение. *Задачей линейного программирования в стандартной форме на минимум* называется частный случай задачи (1.14)—(1.16) в общей форме при $k=m$, $l=n$.

Задача линейного программирования в канонической форме

Задача линейного программирования в канонической форме также является частным случаем записи в общей форме, при этом все специальные ограничения имеют вид равенств, а условие неотрицательности вновь наложено на все переменные. Запись в такой форме удобно использовать для анализа свойств ЗЛП. Кроме того, запись именно в такой форме требуется для решения задачи симплекс-методом — универсальным методом решения любой ЗЛП, который будет рассмотрен далее в главе 3. Дадим строгое определение.

Определение. *Задачей линейного программирования в канонической (основной) форме на максимум* называется частный случай задачи (1.11)—(1.13) в общей форме при $k=0$, $l=n$.

Определение. *Задачей линейного программирования в канонической (основной) форме на минимум* называется частный случай задачи (1.14)—(1.16) в общей форме при $k=0$, $l=n$.

С помощью несложных операций можно переходить от записи ЗЛП в одной из трех рассмотренных форм к записи в любой другой форме. Рассмотрим три необходимых для этого вида преобразований:

1. Замена специальных ограничений в виде равенств на ограничения в виде неравенств (для перехода к стандартной форме записи). Пусть специальное ограничение имеет форму равенства $\sum_{j=1}^{n} a_{ij} x_j = b_i$. Тогда его можно заменить эквивалентной системой двух неравенств вида «меньше либо равно» (либо «больше либо равно», если задача на минимум):

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i; \quad \sum_{j=1}^{n} \left(-a_{ij}\right) \cdot x_j \le -b_i$$

2. Замена специальных ограничений в виде неравенств на ограничения в виде равенств (для перехода к канонической форме записи). Пусть специальное ограничение имеет форму неравенства $\sum_{j=1}^{n} a_{ij} x_j \le b_i$ (в случае задачи на максимум). Тогда его можно преобразовать в равенство, добавив к левой части дополнительную неотрицательную переменную:

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+1} = b_i,\ x_{n+1} \ge 0$$

Если специальное ограничение имеет форму неравенства $\sum_{j=1}^{n} a_{ij} x_j \ge b_i$ (в случае задачи на минимум), то дополнительную неотрицательную переменную нужно вычесть:

$$\sum_{j=1}^{n} a_{ij} x_j - x_{n+1} = b_i,\ x_{n+1} \ge 0$$

В обоих случаях общее число вводимых переменных равно числу неравенств, преобразуемых в равенства.

3. Замена переменных, на которые не наложено условие неотрицательности, на неотрицательные переменные (для перехода к стандартной или канонической форме записи). Пусть на переменную x_i не наложено условие неотрицательности. Тогда ее можно заменить разностью двух неотрицательных переменных:

$$x_i = x_i' - x_i'',\ x_i' \ge 0,\ x_i'' \ge 0$$

Кроме рассмотренных выше трех типов преобразований, позволяющих переходить от одной формы записи к другой, можно выделить также преобразование, приводящее к смене типа экстремума и переходу от задачи на максимум к задаче на минимум, и наоборот. Для этого достаточно ввести новую целевую функцию как произведение исходной целевой функции на минус единицу, а также домножить на минус единицу специальные ограничения в виде неравенств (при их наличии).

Обсудим теперь, что значит решить задачу линейного программирования. Для этого дадим еще ряд определений.

Определение. *Допустимым решением ЗЛП* (1.11)—(1.13) (задачи (1.14)—(1.16)) называется вектор $\left(x_1,\ldots,x_n\right)$, координаты которого удовлетворяют общим и специальным ограничениям этой задачи. Множество всех допустимых решений задачи называется ее *допустимым множеством*.

Определение. Если допустимое множество ЗЛП не пусто, то задача называется *допустимой*, в противном случае — *недопустимой*.

Определение. *Оптимальным решением ЗЛП на максимум* (*на минимум*) называется такое ее допустимое решение, при котором целевая функция достигает своего максимального (минимального) значения. Множество всех оптимальных решений задачи называется ее *оптимальным множеством*.

Определение. Допустимая ЗЛП на максимум (минимум) называется *неограниченной*, если на допустимом множестве ее целевая функция принимает сколь угодно большие значения (сколь угодно большие по модулю отрицательные значения). В противном случае задача называется *ограниченной*.

Таким образом, существует три вида задач линейного программирования (недопустимая, неограниченная и разрешимая), причем разрешимая задача может, в свою очередь, иметь единственное оптимальное решение или целое множество оптимальных решений (рис. 1.1).

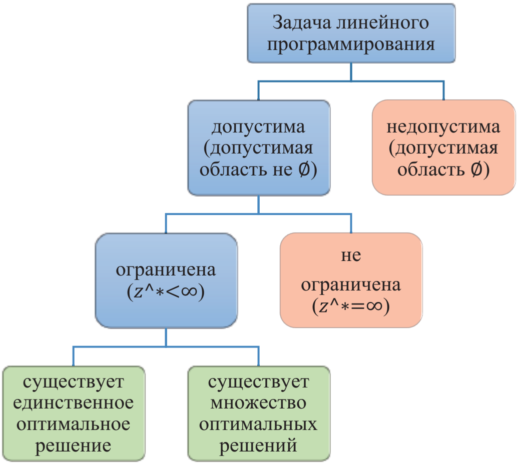


Рис. 1.1. Виды задач линейного программирования

Решить задачу линейного программирования — значит показать, к какому из указанных видов она относится, и в случае, если она разрешима, найти все ее оптимальные решения и оптимальное значение целевой функции. Условием разрешимости ЗЛП являются ее допустимость и ограниченность целевой функции на допустимом множестве. Если задача является недопустимой или неограниченной, то про нее говорят, что она неразрешима.

Примеры решения заданий

Задание 1.1

Дана задача линейного программирования:

$$z = 5x_1 - x_2 \to max$$

$$4x_1 + 5x_2 \le 41$$

$$a \cdot x_1 - x_2 \le -7$$

$$4x_1 - x_2 \le 11$$

$$x_1 + x_2 \le 9$$

$$x_2 \ge 0$$

Для $a = -2$ решите пункты (а) – (б):

(а) Изобразите допустимую область задачи. Четко укажите ее границы.

(б) Решите данную задачу линейного программирования.

(в) Найдите диапазон изменения a, при котором допустимая область задачи является неограниченной.
(г) Запишите исходную задачу в каноническом виде на максимум, укажите размерность матрицы специальных ограничений.

Решение задания 1.1:
(а) Изобразим допустимую область задачи: треугольник ABC, где $A=(-1,9)$, $B=(3,1)$, $C=(4,5)$. Нанесем также линии уровня целевой функции красным пунктиром.

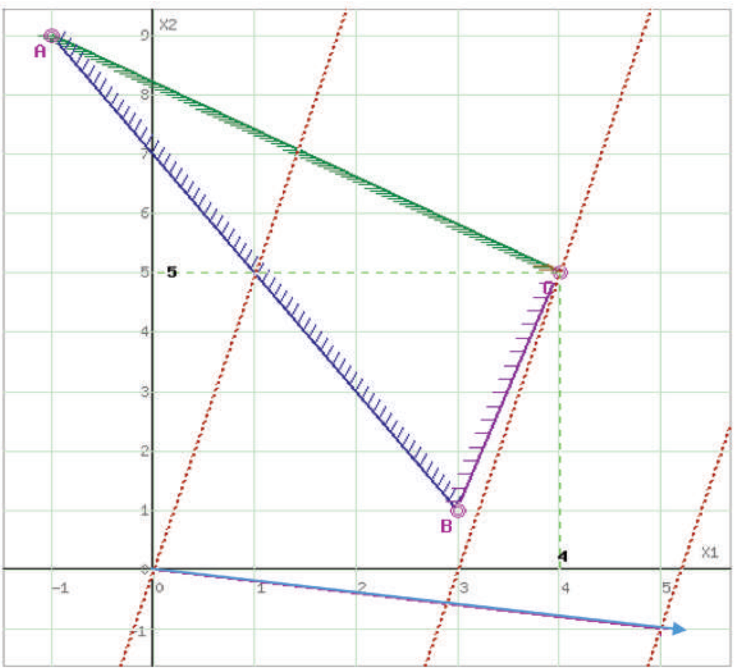


Рис. 1.2. Решение пункта (а) задания 1.1

(б) Решение задачи может быть найдено графически: точка $C=(4;5)$, $z_{max}=15$.
(в) При различных a меняется ограничение № 2 задачи.

Заметим, что при $a \geq 0$ область будет неограниченная, так как прямая, заданная соответствующим ограничением, будет иметь положительный наклон; допустимыми будут все точки, лежащие выше прямой, задаваемой данным ограничением.

Осталось найти границу на a снизу – она получится из условия параллельности ограничений № 2 и № 1. Это будет тогда, когда $\frac{a}{4} = -\frac{1}{5}$, то есть при $a = -\frac{4}{5}$.

Итого, $a \in [-4/5, +\infty)$.

(г) Перейдем к каноническому виду. Сделаем замену $x_1 = x_1' - x_1''$ и приведем неравенства к равенствам.

$$z = 5x_1' - 5x_1'' - x_2 \to max$$
$$4x_1' - 4x_1'' + 5x_2 + x_3 = 41$$
$$a \cdot x_1' - a \cdot x_1'' - x_2 + x_4 = -7$$
$$4x_1' - 4x_1'' - x_2 + x_5 = 11$$
$$x_1' - x_1'' + x_2 + x_6 = 9$$
$$x_1' \geq 0, x_1'' \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0$$

Матрица специальных ограничений задачи в каноническом виде имеет размерность 4×7, поскольку в задаче осталось 4 ограничения, но теперь 7 переменных.

Задания для самостоятельного решения

Задание 1.2

Дана задача линейного программирования (ЗЛП):

$$z = 3x_1 + cx_2 \to max$$
$$-x_1 + 2x_2 \leq 12$$
$$x_1 + ax_2 \leq 12$$
$$-x_1 - 2x_2 \leq -16$$
$$x_1 \geq 0, x_2 \geq 0$$

Даны шесть точек:

(А) (4 ; 8) (Б) (2 ; 7) (В) (5 ; 7) (Г) (9 ; 3) (Д) (8 ; 4) Е) (5 ; 6)

1. Пусть $a = 1$. Какие из указанных точек могут являться оптимальными решениями ЗЛП при различных значениях параметра c?

2. Пусть $c = 2$. Укажите какое-нибудь значение параметра a, при котором ЗЛП является неограниченной. Дайте графическую иллюстрацию.

3. Пусть $c = 2$. Укажите какое-нибудь значение параметра a, при котором оптимальное решение задачи неединственно. Выпишите все оптимальные решения в виде множества точек.

4. Запишите исходную задачу в каноническом виде на минимум, укажите размерность матрицы специальных ограничений.

Задание 1.3

Дана задача линейного программирования (ЗЛП):

$$z = cx_1 + 2x_2 \to max$$

$$2x_1 + x_2 \geq 8$$

$$-2x_1 + x_2 \geq -4$$

$$ax_1 + x_2 \leq 11$$

$$x_2 \geq 0$$

Даны шесть точек:

(А) (4 ; 0) (Б) (1 ; 8) (В) (2 ; 4) (Г) (3 ; 8) (Д) (–3 ; 14) (Е) (4 ; 4)

1. Пусть $a = 1$. Какие из указанных точек могут являться оптимальными решениями ЗЛП при различных значениях параметра c?

2. Пусть $c = 1$. Укажите какое-нибудь значение параметра a, при котором ЗЛП является неограниченной. Дайте графическую иллюстрацию.

3. Пусть $c = 1$. Укажите какое-нибудь значение параметра a, при котором оптимальное решение задачи неединственно. Выпишите все оптимальные решения в виде множества точек.

4. Запишите исходную задачу в каноническом виде на максимум, укажите размерность матрицы специальных ограничений.

Задание 1.4

Дана задача линейного программирования (ЗЛП):

$$z = cx_1 + x_2 \to max$$

$$ax_1 + 2x_2 \leq 10$$

$$3x_1 + bx_2 \leq 15$$

$$-x_1 + x_2 \geq 1$$

$$x_1 \geq 0, x_2 \geq 0$$

Даны шесть точек:

(А) (0 ; 1) (Б) (8 ; 9) (В) (5 ; 8) (Г) (4 ; 7) (Д) (3 ; 5) Е) (0 ; 5)

1. Пусть $a = -1$, $b = -1$. Какие из указанных точек могут являться оптимальными решениями ЗЛП при различных значениях параметра c?

2. Пусть $c = 2$, $b = -1$. Укажите какое-нибудь значение параметра a, при котором ЗЛП является неограниченной. Дайте графическую иллюстрацию.

3. Пусть $c = 2$, $a = -1$. Укажите какое-нибудь значение параметра a, при котором оптимальное решение задачи неединственно. Выпишите все оптимальные решения в виде множества точек.

4. Запишите исходную задачу в каноническом виде на минимум, укажите размерность матрицы специальных ограничений.

ГЛАВА 2
Геометрия линейного программирования

Значительную часть геометрических понятий и определений, вводимых в трехмерном пространстве, можно обобщить на n-мерный случай, при этом мысленное обращение к трехмерной постановке зачастую облегчает понимание многомерной постановки. Дадим два базовых определения – отрезка и прямой.

Определение. Отрезком с концами $\bar{x}', \bar{x}''\left(\bar{x}' \neq \bar{x}''\right)$ называется множество $\left\{\bar{x} = t \cdot \bar{x}' + (1-t) \cdot \bar{x}'' \mid t \in [0,1]\right\}$. Точки, отвечающие $t \in (0,1)$, называются внутренними точками отрезка.

Определение. Прямой в n-мерном пространстве называется линейное многообразие размерности 1.

Обычно используются три способа задания прямой.

1. В параметрическом виде: $\bar{x} = t \cdot \bar{a} + \bar{c}, t \in \mathbb{R}$, где $\bar{c}$ – одна из точек прямой, $\bar{a}$ – направляющий вектор прямой.
2. По двум точкам: $\bar{x} = t \cdot \bar{x}' + (1-t) \cdot \bar{x}'', t \in \mathbb{R}$. Это тот же самый параметрический вид, но немного по-другому оформленный.
3. Системой линейных уравнений $A\bar{x} = \bar{b}$, где $rank(A) = n-1$.

Вернемся к задаче линейного программирования (см. (1.11)–(1.13) или (1.14)–(1.16)). Приведем определения множеств, составляющих систему специальных ограничений задачи линейного программирования $A\bar{x} = \bar{b}\left(8;\, 8\, A\bar{x} \le \bar{b}\right)$.

Определение. Гиперплоскостью в $\mathbb{R}^n$ называется подмножество, заданное уравнением $a_1x_1 + \ldots + a_nx_n = b$.

Определение. Полупространством в $\mathbb{R}^n$ называется подмножество, заданное неравенством $a_1x_1 + \ldots + a_nx_n \le b$ или неравенством $a_1x_1 + \ldots + a_nx_n \ge b$.

Таким образом, система специальных ограничений ЗЛП представляет собой пересечение нескольких гиперплоскостей и полупространств.

Ключевыми свойствами множеств, использующимися при решении задач линейного программирования, являются замкнутость и выпуклость. Напомним их определения, вводимые в курсе математического анализа.

Определение. Множество называется замкнутым, если содержит все свои предельные точки.

Везде далее будут рассматриваться только замкнутые множества. Остановимся подробнее на свойстве выпуклости.

Определение. Множество называется выпуклым, если вместе с любыми своими двумя точками оно содержит отрезок, их соединяющий.

Теорема. Пересечение произвольного набора выпуклых множеств является выпуклым множеством.

Обратите внимание, что объединение выпуклых множеств не обязано являться выпуклым множеством.

Пользуясь определением выпуклого множества, несложно показать (сделайте это в качестве самостоятельного упражнения), что гиперплоскость и полупространство являются выпуклыми множествами.

Поскольку пересечение выпуклых множеств выпукло, а допустимое множество любой ЗЛП является пересечением гиперплоскостей и полупространств, то допустимое множество задачи линейного программирования является выпуклым множеством. Кроме того, выпукло и множество оптимальных решений, то есть верна следующая теорема:

Теорема. Множество допустимых и множество оптимальных решений задачи линейного программирования являются выпуклыми множествами.

Для понимания того, как устроен основной метод решения задач линейного программирования – симплексный метод, важно иметь геометрическое представление о допустимом множестве решений.

Определение. Точка множества называется крайней точкой множества, если она не является внутренней точкой никакого отрезка, состоящего из точек множества.

Иначе говоря, если $\bar{x} = t\cdot\bar{x}' + (1-t)\cdot\bar{x}'', t\in(0,1), \bar{x}'\neq\bar{x}''$ и $\bar{x}$ – крайняя точка выпуклого множества $X\subset\mathbb{R}^n$, то либо $\bar{x}'\notin X$, либо $\bar{x}''\notin X$.

Определение. Выпуклой линейной комбинацией k точек $x^{(1)},\dots,x^{(k)}\in\mathbb{R}^n$ называется их линейная комбинация $\alpha_1 x^{(1)}+\ldots+\alpha_k x^{(k)}$, коэффициенты которой удовлетворяют условиям $\alpha_1\geq 0,\dots,\alpha_k\geq 0, \alpha_1+\ldots+\alpha_k=1$.

Теорема. Множество всех выпуклых линейных комбинаций k точек является выпуклым множеством.

Кроме того, можно доказать, что выпуклая линейная комбинация выпуклых линейных комбинаций k точек является выпуклой линейной комбинацией этих k точек. В частности, отрезок, соединяющий две выпуклые линейные комбинации k точек, состоит из выпуклых линейных комбинаций этих k точек, поскольку представляет собой выпуклую линейную комбинацию двух точек, которые он соединяет.

Определение. Выпуклой оболочкой множества $X\subset\mathbb{R}^n$ называется наименьшее выпуклое множество, содержащее X.

Таким образом, выпуклой оболочкой множества $X \subset \mathbb{R}^n$ является пересечение всех выпуклых множеств в $\mathbb{R}^n$, содержащих множество X.

Теорема. Выпуклой оболочкой множества, состоящего из k точек $x^{(1)},\ldots,x^{(k)} \in \mathbb{R}^n$, является множество всех их выпуклых линейных комбинаций.

Определение. Выпуклым многогранным множеством в $\mathbb{R}^n$ называется пересечение конечного числа полупространств.

Отрезок AB, соединяющий две вершины A и B выпуклого многогранного множества M, называется ребром этого многогранного множества, если любой отрезок $CD \subseteq M$, имеющий общую внутреннюю точку с AB, содержится в AB. Две вершины A и B выпуклого многогранного множества M называются соседними, если отрезок AB является ребром этого многогранного множества.

Теорема. Выпуклое многогранное множество в $\mathbb{R}^n$ является выпуклым множеством.

Итак, допустимое множество задачи линейного программирования является выпуклым многогранным множеством. При этом каждое уравнение можно рассматривать как систему двух неравенств, иначе говоря, каждая гиперплоскость является пересечением двух полупространств.

Определение. Выпуклым многогранником в $\mathbb{R}^n$ называется непустое ограниченное многогранное множество. Крайние точки многогранника называются его вершинами. Каждый выпуклый многогранник имеет конечное число вершин.

Теорема (о представлении многогранника). Выпуклый многогранник является выпуклой оболочкой своих вершин.

Теорема (о представлении многогранного множества). Пусть дано неограниченное выпуклое многогранное множество с крайними точками $x^{(1)},\ldots,x^{(k)}$ и направляющими векторами неограниченных ребер $v^{(1)},\ldots,v^{(m)}$. Тогда любая точка множества представима в виде

$$x = \sum_{i=1}^{k} \alpha_i x^{(i)} + \sum_{j=1}^{m} \gamma_j v^{(j)},$$

$$\sum_{i=1}^{k} \alpha_i = 1,\ \alpha_i \geq 0, \forall i = 1,\ldots,k,\ \gamma_j \geq 0, \forall j = 1,\ldots,m$$

Таким образом, ограниченное допустимое множество задачи линейного программирования представимо в виде выпуклого многогранника и имеет конечное число крайних точек. Если же допустимое множество не ограничено, то оно представимо в виде суммы Минковского двух множеств (многогранника и конуса), то есть также имеет конечное число крайних точек. Это и есть геометрическое представление допустимого множества задачи линейного программирования.

Мы получили представление о том, что такое геометрически допустимое множество задачи линейного программирования. Для задачи в каноническом виде важно изучить задающую это множество систему линейных уравнений.

Определение. Решение системы линейных уравнений называется базисным, если столбцы матрицы ограничений, отвечающие переменным, принимающим ненулевые значения, линейно независимы.

Определение. Базисное решение системы специальных ограничений задачи линейного программирования в каноническом виде называется допустимым, если все его координаты неотрицательны.

Таким образом, допустимое решение системы линейных уравнений называется базисным, если столбцы матрицы ограничений, отвечающие переменным, принимающим *положительные* значения, линейно независимы.

Определение. Базисное решение называется вырожденным, если количество ненулевых координат меньше ранга матрицы системы специальных ограничений. В противном случае базисное решение называется невырожденным.

Замечание. Если базисное решение вырождено, то ему соответствует несколько различных наборов базисных переменных.

Теорема. Для задачи линейного программирования в канонической форме существует взаимно однозначное соответствие между крайними точками допустимого множества и допустимыми базисными решениями, то есть каждая крайняя точка допустимого множества определяется некоторым допустимым базисным решением, и наоборот – каждое допустимое базисное решение задает крайнюю точку допустимого множества.

Теорема. Если множество допустимых решений задачи линейного программирования в канонической форме не пусто, то существует базисное допустимое решение.

Теорема. Если задача линейного программирования в канонической форме имеет оптимальное решение, то она имеет и базисное оптимальное решение.

Теорема. Если допустимое множество задачи линейного программирования ограничено, то целевая функция принимает наименьшее значение по крайней мере в одной из его крайних точек.

Из приведенных выше теорем следует, что решение задачи линейного программирования – это, в некотором смысле, поиск при помощи перебора, в какой из крайних точек допустимого множества или на каком из базисных допустимых решений задачи линейного программирования целевая функция достигает своего оптимального значения. Основной метод решения задач линейного программирования — симплексный метод — представляет собой не что иное, как упорядоченный перебор вершин допустимого множества ЗЛП. Симплексный метод будет подробно рассмотрен в следующей главе. Сформулируем еще несколько теорем о разрешимости задачи линейного программирования.

Лемма. Если целевая функция ограничена снизу на допустимом множестве, то для любой точки допустимого множества x^0 найдется базисное допустимое решение x такое, что $cx \le cx^0$.

Теорема (критерий разрешимости задачи). Задача линейного программирования разрешима тогда и только тогда, когда допустимое множество не пусто и целевая функция ограничена на нем снизу.

Теорема. При решении задачи линейного программирования симплексным методом каждой итерации соответствует переход от вершины множества допустимых решений задачи к той же самой вершине или к соседней вершине допустимого множества.

Примеры решения заданий

Задание 2.1

Укажите, какие из указанных векторов являются допустимыми базисными решениями системы, а какие не являются. Дайте подробные объяснения.

$$\begin{aligned} x_1 - x_2 - x_3 &= 0 \\ 2x_1 - 2x_2 \qquad + x_4 + 2x_5 &= 4 \\ 4x_1 - 3x_2 - x_3 + x_4 + 3x_5 &= 6 \\ x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 &\ge 0,\ x_5 \ge 0 \end{aligned}$$

(А) $(0;0;0;0;2)$ (В) $(3;0;3;0;-1)$

(Б) $(1;0;1;0;1)$ (Г) $(2;0;2;0;0)$

Решение задания 2.1:

По определению допустимая точка x, удовлетворяющая системе общих $(x \ge 0)$ и специальных ограничений $(Ax = b)$, называется базисной, если набор столбцов матрицы A специальных ограничений, соответствующий ненулевым координатам x, является линейно независимым.

Все точки, кроме (В), являются допустимыми, а у точки (В) $x_5 < 0$, поэтому она является недопустимой, следовательно, точка (В) не подходит.

У точки (А) единственная ненулевая координата, ей соответствует набор из одного столбца; набор столбцов, состоящий из единственного столбцов, линейно независим, поэтому точка (А) подходит.

Точке (Б) соответствует следующий набор столбцов:

$$\left\{\begin{pmatrix}1\\2\\4\end{pmatrix},\begin{pmatrix}-1\\0\\-1\end{pmatrix},\begin{pmatrix}0\\2\\3\end{pmatrix}\right\}$$

Он является линейно зависимым, потому что сумма первого и второго столбцов равна третьему. Следовательно, точка (Б) не подходит.

Точке (Г) соответствует следующий набор столбцов:

$$\left\{\begin{pmatrix}1\\2\\4\end{pmatrix},\begin{pmatrix}-1\\0\\-1\end{pmatrix}\right\}$$

Он является линейно независимым, значит, точка (Г) подходит.

Таким образом, точки (А) и (Г) являются допустимыми базисными решениями.

Задание 2.2

Дана задача линейного программирования:

$$\begin{aligned}
&z = 5x_1 - x_2 \to max\\
&4x_1 + 5x_2 \le 41\\
&-2x_1 - x_2 \le -7\\
&4x_1 - x_2 \le 11\\
&x_1 + x_2 \le 9\\
&x_2 \ge 0
\end{aligned}$$

Представьте точки $(0,6)$ и $(2,6)$ в виде выпуклой линейной комбинации вершин допустимого множества задачи. Если это невозможно сделать, объясните почему.

Решение задания 2.2:

Точка $(0,6)$ не принадлежит допустимой области, поэтому не может быть представлена в виде выпуклой линейной комбинации вершин допустимого множества. Непосредственно в этом можно убедиться, решив систему:

$$\begin{cases} \begin{pmatrix} 0 \\ 6 \end{pmatrix} = \alpha_1 \begin{pmatrix} 4 \\ 5 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} -1 \\ 9 \end{pmatrix} \\ \alpha_1 + \alpha_2 + \alpha_3 = 1 \\ \alpha_i \geq 0, i = 1,2,3 \end{cases}$$

Из системы, без учета ограничений на знак, методами линейной алгебры найдется $\alpha_1 = -\dfrac{1}{6} < 0$, чего быть не должно.

Точка $(2,6)$ принадлежит допустимой области, поэтому может быть представлена в виде выпуклой линейной комбинации вершин допустимого множества. Непосредственно в этом можно убедиться, решив систему:

$$\begin{cases} \begin{pmatrix} 2 \\ 6 \end{pmatrix} = \alpha_1 \begin{pmatrix} 4 \\ 5 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} -1 \\ 9 \end{pmatrix} \\ \alpha_1 + \alpha_2 + \alpha_3 = 1 \\ \alpha_i \geq 0, i = 1,2,3 \end{cases}$$

Из системы методами линейной алгебры найдется $\alpha_1 = \dfrac{1}{2}, \alpha_2 = \dfrac{1}{8}, \alpha_3 = \dfrac{3}{8}$.

Все коэффициенты неотрицательные, следовательно, мы указали выпуклую линейную комбинацию для точки $(2,6)$.

Задания для самостоятельного решения

Задание 2.3. Докажите, что множество $\{\bar{x} = t \cdot \bar{a} + \bar{c} \mid t \in [\alpha, \beta]\}$ является отрезком.

Задание 2.4. Докажите, что отрезок и прямая являются выпуклым множеством.

Задание 2.5. Докажите, что гиперплоскость и полупространство являются выпуклым множеством.

Задание 2.6. Докажите, что допустимое и оптимальное множества задачи линейного программирования являются выпуклым множеством.

Задание 2.7. Докажите, что пересечение произвольного числа выпуклых множеств является выпуклым множеством. Приведите пример трех выпуклых множеств, объединение которых не является выпуклым множеством.

Задание 2.8. Докажите (без ссылок на какие-либо теоремы), что множество всех выпуклых линейных комбинаций трех точек $x^{(1)}, x^{(2)}, x^{(3)}$ является выпуклым множеством.

Задание 2.9. Докажите (используя только теорему о представлении многогранника, не используя геометрические соображения и теоремы Вейерштрасса), что наибольшее значение линейной функции $z = cx$ на множестве M всех выпуклых линейных комбинаций точек $x^{(1)},\dots,x^{(k)}$ достигается в одной из этих точек.

Задание 2.10. Докажите (без ссылок на какие-либо теоремы и не используя геометрических соображений), что выпуклая оболочка трех точек $x^{(1)}, x^{(2)}, x^{(3)}$ совпадает с множеством всех их выпуклых линейных комбинаций.

Задание 2.11. Гиперплоскость $ax = b$ делит пространство на два полупространства. Докажите (без ссылок на какие-либо теоремы), что множество всех выпуклых линейных комбинаций k точек $x^{(1)},\dots,x^{(k)}$ тогда и только тогда содержится в одном полупространстве, когда все эти точки содержатся в одном полупространстве.

Задание 2.12. Какие из указанных векторов являются допустимыми базисными решениями системы:

$$6x_1 - x_2 + 3x_3 + 2x_4 = 16$$

$$3x_1 + x_2 + x_3 + x_4 = 8$$

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0$$

(А) (0;0;0;8)	(Б) (0;1;3;4)	(В) (0;2;6;0)
(Г) (2;0;0;2)	(Д) (3;0;0;-1)	(Е) (3;2;0;0)

Задание 2.13

(а) Для задания 1.2 ответьте вопросы:

- Какие из указанных точек можно представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a = 1$?

- Какие из указанных точек можно неединственным образом представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a=1$?

(б) Для задания 1.3 ответьте на вопросы:

- Какие из указанных точек можно представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a=1$?
- Какие из указанных точек можно неединственным образом представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a=1$?

(в) Для задания 1.4 ответьте на вопросы:

- Какие из указанных точек можно представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a=-1,\ b=-1$?
- Какие из указанных точек можно неединственным образом представить в виде выпуклой линейной комбинации вершин множества, заданного системой ограничений ЗЛП при $a=-1,\ b=-1$?

ГЛАВА 3

Симплексный метод решения задач линейного программирования

В данной главе будет рассмотрен основной метод, применяемый для решения ЗЛП, – симплекс-метод[1]. Отметим, что формально он применим только для задач в каноническом виде с неотрицательной правой частью. Однако, как было разобрано в первом разделе, любая ЗЛП путем несложных преобразований может быть записана в каноническом виде.

Как мы обсуждали в предыдущей главе, наилучшее значение целевой функции, если оно есть, достигается в крайней точке (или вершине) допустимого множества, которая соответствует некоторому допустимому базисному решению. Общее число вершин может быть большим, и при этом прямой перебор их всех становится очень трудоемким даже при современном уровне развития вычислительной техники. Основной идеей предложенного американским математиком Джеймсом Данцигом симплекс-метода является упорядочивание перебора, чтобы при каждом переходе от одной вершины к другой значение целевой функции не ухудшалось.

Определение. *Симплексный метод (симплекс-метод)* — метод решения задач линейного программирования в канонической форме путем направленного перебора вершин допустимого множества.

Рассмотрим каноническую ЗЛП на минимум с неотрицательными правыми частями. Отметим, что мы делаем это без ограничения общности, так как любая ЗЛП может быть записана в канонической форме (см. главу 1), кроме того, если какой-то коэффициент в правой части отрицателен ($b_i < 0$), то можно сделать его положительным, домножив i-е равенство на минус единицу. Потребуем также, чтобы ранг матрицы специальных ограничений был не меньше, чем m (если это не так, то удалим из задачи «лишние» линейно зависимые ограничения). Кроме того, не-

[1] Существуют и другие методы решения задачи линейного программирования: двойственный симплекс-метод (обсуждается в главе 5), метод внутренней точки (interior point method), алгоритм Кармаркара.

много расширим классическую постановку и добавим в целевую функцию константу c_0. Назовем такую задачу (3.1) *основной* (для нее $m \le n, b_i \ge 0$):

$$\begin{cases} z = c_0 + c_1 x_1 + \ldots + c_n x_n \to min \\ a_{11} x_1 + \ldots + a_{1n} x_n = b_1 \\ \ldots \\ a_{m1} x_1 + \ldots + a_{mn} x_n = b_m \\ x_1 \ge 0, \ldots, x_n \ge 0 \end{cases} \tag{3.1}$$

Решение основной задачи (3.1) симплекс-методом осуществляется в два этапа:

- *на первом этапе* отыскивается какое-либо (исходное) допустимое базисное решение либо показывается, что задача недопустима;
- *на втором этапе* найденное исходное решение итерационно улучшается, в результате отыскивается оптимальное решение задачи либо показывается, что она не ограничена.

Можно считать, что нулевым этапом решения ЗЛП является ее приведение к основной, то есть к записанной в канонической форме на минимум с неотрицательной правой частью и матрицей специальных ограничений, имеющей полный ранг.

Рассмотрим подробнее матрицу A специальных ограничений основной задачи (3.1). Изначально она может содержать от нуля до m различных единичных столбцов, и соответствующие таким столбцам переменные являются базисными. Нашей целью является наличие в ней единичной подматрицы порядка m, то есть наличие ровно m единичных столбцов. В этом случае, приравнивая оставшиеся (если они есть) $n-m$ свободных переменных к нулю, а m базисных переменных — к соответствующим свободным членам, получаем исходное допустимое базисное решение. Если изначально матрица A содержит только r единичных столбцов, где $r<m$, то в $m-r$ уравнений вводятся $m-r$ *искусственных переменных* (по одной в каждое) таким образом, чтобы новая матрица специальных ограничений содержала единичную подматрицу порядка m. Рассмотрим для определенности случай, когда матрица A не содержит ни одного единичного столбца, то есть когда $r=0$. В этом случае искусственную переменную нужно вести в каждое уравнение системы (3.1): к левой части i-го уравнения добавляем $y_i \ge 0$, $i=1,\ldots,m$. Обозначим $u = \sum y_i \ge 0$ и заметим, что добавление искусственных переменных приводит к равносильной системе тогда и только тогда, когда $u=0$. Рассмотрим новую задачу с целевой функцией u (назовем такую задачу *вспомогательной*):

$$\begin{cases} u = y_1 + \ldots + y_m \to min \\ a_{11}x_1 + \ldots + a_{1n}x_n + y_1 = b_1 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n + y_m = b_m \\ c_0 + c_1x_1 + \ldots + c_nx_n = z \\ x_1 \geq 0, \ldots, x_n \geq 0, y_1 \geq 0, \ldots, y_m \geq 0 \end{cases} \tag{3.2}$$

Здесь выражение исходной целевой функции z через переменные x_i, записанное в предпоследней строке, формально говоря, не является частью задачи, а оставлено нами в технических целях. Поскольку матрица специальных ограничений задачи (3.2) (в которой $n+m$ столбцов и m строк) содержит ровно m единичных столбцов, то для нее существует исходное допустимое базисное решение, и, значит, она допустима. Возможны две ситуации: $u^* = 0$ и $u^* > 0$. В первом случае то решение, на котором достигается нулевой оптимум, выглядит как $\left(x_1', \ldots, x_n', 0, \ldots, 0\right)$, и при этом вектор $\left(x_1', \ldots, x_n'\right)$ является допустимым решением исходной задачи (3.1), таким образом, задача (3.1) допустима. Во втором случае не существует такого n-мерного вектора, который разрешал бы систему специальных ограничений исходной задачи, то есть задача (3.1) недопустима. Итак, исходная задача допустима тогда и только тогда, когда оптимальное значение целевой функции вспомогательной задачи равно нулю. Опираясь на этот факт, приведем пошаговый алгоритм решения основной задачи (3.1)

Первый этап. Поиск исходного допустимого базисного решения либо получение вывода о недопустимости задачи

Шаг 1. Выразим функцию u через свободные переменные и запишем z и u в следующем виде:

$$z - c_1x_1 - \ldots - c_nx_n = c_0 \tag{3.3}$$

$$u - d_1x_1 - \ldots - d_nx_n = d_0 \tag{3.4}$$

Выпишем исходную симплекс-таблицу для задачи (3.2) с учетом (3.3), (3.4):

Исходная симплекс-таблица

—	x_1	x_2	…	x_n	y_1	y_2	…	y_m	b
y_1	a_{11}	a_{12}	…	a_{1n}	1	0	…	0	b_1
y_2	a_{21}	a_{22}	…	a_{2n}	0	1	…	0	b_2
…	…	…	…	…	…	…	…	…	…
y_m	a_{m1}	a_{m2}	…	a_{mn}	0	0	…	1	b_m
z	$-c_1$	$-c_2$	…	$-c_n$	0	0	…	0	c_0
u	$-d_1$	$-d_2$	…	$-d_n$	0	0	…	0	d_0

На исходном допустимом базисном решении вспомогательной задачи $(0,\ldots,0,b_1,\ldots,b_m)$ значение целевой функции u равно d_0 (а значение z равно c_0). Это исходная вершина допустимого множества задачи. Переменные $x_1,\ldots,x_n$ являются свободными, переменные $y_1,\ldots,y_m$ – базисными. Напомним, что все переменные в задаче должны быть неотрицательны, и поскольку свободные переменные равны нулю, то в нашей власти, вводя их в базис, увеличивать их значение. При этом, конечно, поменяются значения базисных переменных (ведь они выражены через свободные), а значит, нужно следить, чтобы в результате они остались неотрицательными. Переходим к шагу 2.

Шаг 2. Рассмотрим элементы, расположенные в «свободных» столбцах строки u симплекс-таблицы (исходной или преобразованной на шаге 4).

2.1. Если среди них есть положительные, то, увеличивая значение соответствующей свободной переменной (если это возможно), можно уменьшить значение целевой функции u. Например, если $d_1 < 0$, то увеличение свободной переменной x_1 (изначально равной нулю) приведет к снижению u. Чтобы понять, можно ли увеличивать свободную переменную (в нашем примере x_1), нужно посмотреть, как это повлияет на базисные переменные, а именно проверить, что никакая из них не станет отрицательной. В любом случае нам придется изменить базис и ввести туда x_1 — переходим к шагу 3.

2.2. Если в «свободных» столбцах строки u нет положительных элементов, то уменьшать значение целевой функции больше нельзя – ведь любое изменение свободных переменных приведет лишь к ее росту. Значит, мы нашли оптимальное значение u – оно равно значению свободного члена, стоящему в последнем столбце строки u (изначально это элемент d_0). Если при этом найденное оптимальное значение u положительно, то исходная задача недопустима и ее решение закончено. Если же оно равно нулю, то равны нулю и все искусственные переменные $y_1,\ldots,y_m$

(поскольку целевая функция u представляет собой их сумму и они неотрицательны). Это значит, что либо переменные $y_1,\dots,y_m$ не входят в базис, либо мы имеем дело с вырожденным базисным решением, поскольку какая-то равная нулю переменная (пусть для определенности переменная y_1) находится в базисе. В этом случае в строке u обязательно найдется ноль, стоящий в свободном столбце (иначе все свободные переменные были бы равны нулю), и соответствующую этому столбцу переменную x_j можно ввести в базис вместо y_1. Таким образом, в любом случае решение вспомогательной задачи закончено, и в найденном оптимальном базисном решении базис образован исходными переменными, а все искусственные переменные (изначально образовывавшие базис) выведены из него. Значит, исходная задача допустима, и мы нашли ее допустимое базисное решение. Вычеркнув из таблицы строку u и столбцы, соответствующие (свободным!) искусственным переменным $y_1,\dots,y_m$, получаем исходную таблицу для основной задачи и переходим к шагу 5.

Шаг 3. Мы приходим на шаг 3, когда среди элементов, расположенных в «свободных» столбцах строки u, есть положительные. Выбираем любой такой положительный элемент (если их несколько, то при компьютерной реализации алгоритма берут тот, ввод которого в базис обеспечит наибольшее снижение целевой функции, а при ручной реализации – тот, выбор которого облегчает расчеты). Столбец $j_{\text{вед}}$, содержащий выбранный элемент, называется *ведущим*. Геометрически он соответствует ребру допустимого множества, вдоль которого мы будем двигаться, а содержательно – той свободной переменной $x_{j_{\text{вед}}}$, которую мы будем вводить в базис. Мы можем увеличивать переменную $x_{j_{\text{вед}}}$ до тех пор, пока все базисные переменные больше либо равны нулю. Поведение базисной переменной, стоящей в i-й строке, при увеличении $x_{j_{\text{вед}}}$ зависит от знака элемента, стоящего на пересечении i-й строки и столбца $j_{\text{вед}}$. Если этот знак положительный, то с ростом $x_{j_{\text{вед}}}$ значение базисной переменной x_i будет падать, если знак отрицательный – то расти, если же на пересечении i-й строки и столбца $j_{\text{вед}}$ стоит ноль, то рост $x_{j_{\text{вед}}}$ не повлияет на переменную x_i. Среди элементов ведущего столбца, расположенных в базисных строках, обязательно найдется хотя бы один положительный элемент (иначе целевую функцию можно было бы неограниченно уменьшать, а в нашей задаче она ограничена нулем). Для положительных элементов ведущего столбца, расположенных в первых m строках, подсчитываем частное $\dfrac{b_i}{a_{ij_{\text{вед}}}}$, $i = 1,\dots,m$. Выбираем в качестве ведущего элемент, для которого это частное минимально, соответствующая i-я строка становится ведущей: $\dfrac{b_{i_{\text{вед}}}}{a_{i_{\text{вед}} j_{\text{вед}}}} = \min\limits_{a_{ij_{\text{вед}}}>0}\left(\dfrac{b_i}{a_{ij_{\text{вед}}}}\right)$. Геометрически ведущий элемент соответствует расстоянию, которое нужно пройти по ребру для попадания в новую вершину допустимого множества. Отметим, что чем проще ведущий элемент, тем легче осуществлять расчеты вручную. Скажем, ведущий элемент, равный единице, «лучше», чем равный двойке, а равный пяти – «лучше», чем равный трем. Поскольку при фиксации ведущего столбца выбор ведущей строки осуществляется уже однозначно (при уникальном минимальном частном), то при ручных расчетах стоит внимательнее отнестись к выбору ведущего столбца (где у нас есть некоторая свобода) и делать этот выбор с учетом простоты ведущего элемента. Переходим к шагу 4.

Шаг 4. Ведущую строку (с номером $i_{\text{вед}}$) делим на ведущий элемент $a_{i_{\text{вед}} j_{\text{вед}}}$, из остальных строк (с номерами $i = 1, \ldots, m$, $i \neq i_{\text{вед}}$) вычитаем ведущую строку, умноженную на коэффициент $\dfrac{a_{ij_{\text{вед}}}}{a_{i_{\text{вед}} j_{\text{вед}}}}$. Ведущий столбец после таких преобразований становится единичным, переменная, соответствующая ведущей строке, выходит из базиса, а переменная, соответствующая ведущему столбцу, входит в базис. Преобразованную симплекс-таблицу записываем под текущей, переходим к шагу 2.

Второй этап. Поиск оптимального решения либо получение вывода о неограниченности задачи

Шаг 5. Шаг 5 во многом аналогичен шагу 2 первого этапа. Рассмотрим элементы, расположенные в «свободных» столбцах строки z симплекс-таблицы (полученной исходно на шаге 2 или преобразованной на шаге 7).

5.1. Если среди них есть положительные, то, увеличивая значение соответствующей свободной переменной (если это возможно), можно уменьшить значение целевой функции z. Чтобы понять, можно ли увеличивать свободную переменную, нужно посмотреть, как это повлияет на базисные переменные, а именно проверить, что никакая из них не станет отрицательной. В любом случае нам придется изменить базис – переходим к шагу 6.

5.2. Если в «свободных» столбцах строки z нет положительных элементов, то уменьшить значение целевой функции больше нельзя – ведь любое изменение свободных переменных приведет лишь к ее росту. Значит, мы нашли оптимальное значение z (оно равно значению свободного члена, стоящему в последнем столбце строки z) и базисное оптимальное решение (оно получается приравниванием базисных переменных свободным членам, а свободных переменных — нулю). Если при этом в свободных столбцах строки z есть нулевые элементы, то найденное решение неединственно. В этом случае для отыскания всех решений нужно посмотреть на каждый из свободных столбцов с нулевым элементом в строке целевой функции. Если в базисных строках какого-то из них нет положительных элементов, то мы можем бесконечно увеличивать соответствующую свободную переменную (не меняя при этом значение целевой функции), и это не выведет нас из допустимой области. В этом случае множество оптимальных решений не ограничено, и для его формального описания нужно параметризовать свободную переменную, соответствующую такому столбцу. Если же в базисных строках каждого из столбцов есть положительные элементы, то оптимальное множество ограничено и представляет собой выпуклую линейную комбинацию всех базисных оптимальных решений. Для их отыскания надо проделать шаг 6, выбрав в качестве ведущего каждый из таких столбцов.

Шаг 6. Шаг 5 во многом аналогичен шагу 3 первого этапа. Мы приходим на шаг 6, когда среди элементов, расположенных в «свободных» столбцах строки z, есть положительные. Выбираем любой такой положительный элемент (если их несколько, то при компьютерной реализации алгоритма берут любой максимальный, а при ручной реализации – тот, выбор которого облегчает расчеты). Столбец $j_{\text{вед}}$, содержащий выбранный элемент, вновь называется ведущим и соответствует той свободной переменной $x_{j_{\text{вед}}}$, которую мы будем вводить в базис. Мы можем увеличивать переменную $x_{j_{\text{вед}}}$ до тех пор, пока все базисные переменные больше либо равны нулю. Поведение базисной переменной, стоящей в i-й строке, при увеличении $x_{j_{\text{вед}}}$ зависит от знака элемента, стоящего на пересечении i-й строки и столбца $j_{\text{вед}}$. Если этот знак положительный, то с ростом $x_{j_{\text{вед}}}$ значение базисной переменной x_i будет падать, если знак отрицательный – то расти, если же на пересечении i-й строки и столбца $j_{\text{вед}}$ стоит ноль, то рост $x_{j_{\text{вед}}}$ не повлияет на переменной x_i. Если среди элементов ведущего столбца, расположенных в базисных строках, нет ни одного положительного, то целевую функцию можно неограниченно уменьшать. В этом случае задача не ограничена, ее решение закончено. Если же это не так, то для положительных элементов ведущего столбца, расположенных в первых m строках, подсчитываем частное $\dfrac{b_i}{a_{ij_{\text{вед}}}}$, $i = 1, \dots, m$. Выбираем в качестве ведущего элемент, для которого это частное минимально, соответствующая i-я строка становится ведущей: $\dfrac{b_{i_{\text{вед}}}}{a_{i_{\text{вед}}j_{\text{вед}}}} = \min\limits_{a_{ij_{\text{вед}}}>0}\left(\dfrac{b_i}{a_{ij_{\text{вед}}}}\right)$. Переходим к шагу 7.

Шаг 7. Шаг 7 во многом аналогичен шагу 4 первого этапа. Ведущую строку (с номером $i_{вед}$) делим на ведущий элемент $a_{i_{вед}j_{вед}}$, из остальных строк (с номерами $i = 1, \ldots, m$, $i \neq i_{вед}$) вычитаем ведущую строку, умноженную на коэффициент $\frac{a_{ij_{вед}}}{a_{i_{вед}j_{вед}}}$. Ведущий столбец после таких преобразований становится единичным, переменная, соответствующая ведущей строке, выходит из базиса, а переменная, соответствующая ведущему столбцу, входит в базис. Преобразованную симплекс-таблицу записываем под текущей, переходим к шагу 5.

Проиллюстрируем симплексный метод решения задач линейного программирования на примере задачи:

$$z = 4x_1 + x_2 \to max$$

$$x_1 + x_2 \leq 4 \quad (1)$$

$$-x_1 + x_2 \leq 2 \quad (2)$$

$$x_1 - x_2 \leq 2 \quad (3)$$

$$x_i \geq 0,\ i = 1,2$$

Для начала решим ее графически:

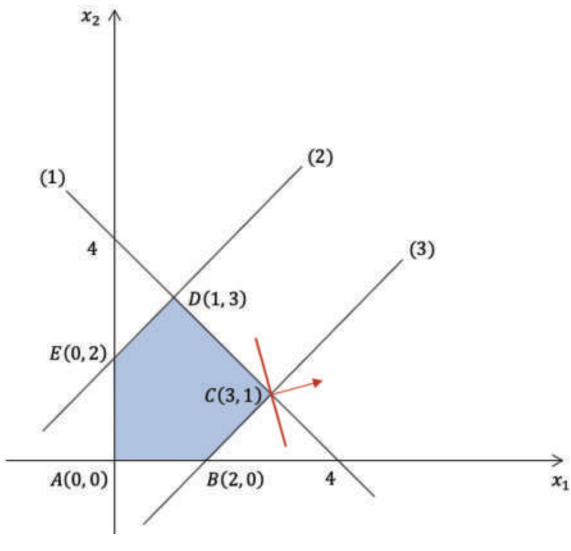


Рис. 3.1

Светло-синим цветом выделена допустимая область данной задачи линейного программирования. Точки A, B, C, D, E – крайние точки допустимой области. По теореме из предыдущей главы мы знаем, что точкам A, B, C, D, E соответствуют базисные допустимые решения задачи линейного программирования. Красным цветом на рис. 3.1 выделена линия уровня целевой функции и градиент – видно, что точка C является оптимальной.

Теперь решим эту задачу симплексным методом. Для этого, согласно описанному выше алгоритму, приведем ее к каноническому виду на минимум с неотрицательной правой частью:

$$\tilde{z} = -4x_1 - x_2 \rightarrow min$$

$$x_1 + x_2 + x_3 = 4$$

$$-x_1 + x_2 + x_4 = 2$$

$$x_1 - x_2 + x_5 = 2$$

$$x_i \geq 0, i = 1, \ldots, 5$$

В данной системе переменные x_3, x_4, x_5 являются базисными – соответствующие им столбцы матрицы специальных ограничений задачи линейного программирования являются линейно независимыми, так как образуют единичную подматрицу. Переменные x_1, x_2 являются свободными. Таким образом, мы можем пропустить этап 1 поиска исходного допустимого базисного решения, поскольку здесь оно задано сразу: данной системе соответствует допустимое базисное решение – точка $(0,0,4,2,2)$. На рис. 3.1 в двумерном пространстве ей соответствует точка A – крайняя точка допустимой области.

Исходному допустимому базисному решению соответствует симплексная таблица:

	x_1	x_2	x_3	x_4	x_5	b
x_3	1	1	1	0	0	4
x_4	–1	1	0	1	0	2
x_5	1	–1	0	0	1	2
$\tilde{z}$	4	1	0	0	0	0

Шаг 5: так как в строке целевой функции $\tilde{z}$ есть положительные элементы (и в столбце x_1, и в столбце x_2), то данная симплексная таблица не является финальной (значение целевой функции можно уменьшить).

Шаг 6: исходя из данной таблицы $\tilde{z} = -4x_1 - x_2 \to min$, следовательно, при введении в базис $x_1 \geq 0$ или $x_2 \geq 0$ целевая функция $\tilde{z}$ не увеличится. Если выбрать в качестве ведущего первый столбец (то есть если вводить в базис переменную x_1), то среди его положительных элементов отношение b_i / a_{i1} минимально для строки $x_5 \left(\frac{2}{1} < \frac{4}{1}\right)$, и ведущим станет его элемент, стоящий в строке x_5 (он равен единице). Если же выбрать в качестве ведущего второй столбец (то есть если вводить в базис переменную x_2), то среди его положительных элементов отношение b_i / a_{i2} минимально для строки $x_4 \left(\frac{2}{1} < \frac{4}{1}\right)$, и ведущим станет его элемент, стоящий в строке x_4 (он также равен единице). Значение ведущего элемента в данном случае не зависит от выбора столбца. Выберем в качестве ведущего первый столбец, в качестве ведущей – строку x_5.

Шаг 7: проведем итерацию симплексного метода и введем в базис переменную x_1 вместо переменной x_5, получим новую симплексную таблицу:

	x_1	x_2	x_3	x_4	x_5	b
x_3	0	2	1	0	–1	2
x_4	0	0	0	1	1	4
x_1	1	–1	0	0	1	2
$\tilde{z}$	0	5	0	0	–4	–8

Данной симплексной таблице соответствует допустимое базисное решение задачи – точка $(2,0,2,4,0)$, на рис. 3.1 в двумерном пространстве это точка B. Графически итерации соответствует переход по ребрам допустимой области от одной крайней точки к другой (то есть переход от одного допустимого базисного решения к другому), в данном случае переход по ребру AB в точку B, с улучшением целевой функции (ее значение уменьшилось на 8).

Шаг 5: так как в строке целевой функции $\tilde{z}$ остался положительный элемент в столбце x_2, то данная таблица не является финальной.

Шаг 6: выберем в качестве ведущего второй столбец. В его первых трех строках есть единственный положительный элемент (элемент 2 в строке x_3), он становится ведущим элементом.

Шаг 7: проведем итерацию симплексного метода и введем в базис переменную x_2 вместо переменной x_3, получим новую симплексную таблицу:

	x_1	x_2	x_3	x_4	x_5	b
x_2	0	1	1/2	0	−1/2	1
x_4	0	0	0	1	1	4
x_1	1	0	1/2	0	1/2	3
$\tilde{z}$	0	0	−5/2	0	−3/2	−13

Данной симплексной таблице соответствует допустимое базисное решение задачи – точка $(3,1,0,4,0)$, на рис. 3.1 в двумерном пространстве это точка C. Графически итерации соответствует переход по ребрам допустимой области от одной крайней точки к другой (то есть переход от одного допустимого базисного решения к другому), в данном случае переход по ребру BC в точку C, с улучшением целевой функции (ее значение уменьшилось на 5).

Шаг 5: таблица является финальной, поскольку все элементы в строке целевой функции являются неположительными, следовательно, найдено базисное оптимальное решение задачи, приведенной к каноническому виду (в пространстве $\mathbb{R}^5$):

$$\begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \\ x_5^* \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 0 \\ 4 \\ 0 \end{pmatrix}, \tilde{z}^* = -13$$

Данное решение единственно, так как все нули в строке целевой функции расположены в базисных столбцах. Окончательно получаем следующее решение исходной задачи (напомним, что она была поставлена в пространстве $\mathbb{R}^2$):

$$\begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, z^* = -\tilde{z}^* = 13$$

Отметим еще раз, что каждой симплекс-таблице соответствует базисное допустимое решение, например, финальной таблице разобранной выше задачи можно сопоставить следующий канонический вид задачи линейного программирования:

$$\begin{cases} \tilde{z} = -13 + \frac{5}{2}x_3 + \frac{3}{2}x_5 \to min \\ x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_5 = 1 \\ x_4 + x_5 = 4 \\ x_1 + \frac{1}{2}x_3 + \frac{1}{2}x_5 = 3 \\ x_i \geq 0, i = 1, \ldots, 5 \end{cases}$$

Задачу линейного программирования и симплексный метод можно представить в матричном виде. Начнем с задачи в каноническом виде:

$$z = c^T * x \to min$$

$$A * x = b$$

$$x \geq 0$$

В данном случае x – это n-мерный вектор-столбец неизвестных переменных, b – это m-мерный вектор-столбец правых частей ограничений, c^T – вектор-строка коэффициентов целевой функции, A – матрица специальных ограничений размера $m \times n$, пусть ее ранг равен k.

Пусть у нас известны k базисных переменных (соответствующих некоторому базисному решению, которое дополнительно является допустимым), тогда ограничения специального вида можно представить в следующем виде:

$$A_B * x_B + A_N * x_N = b$$

В этой записи x_B – это k-мерный вектор-столбец базисных переменных, A_B – это матрица, составленная из столбцов, соответствующих базисным переменным (заметим, что это линейно независимый набор столбцов), x_N – это $(n-k)$-мерный вектор-столбец свободных переменных, A_N – это матрица, составленная из столбцов, соответствующих базисным переменным. Далее можно найти значения базисных переменных:

$$x_B = A_B^{-1} * b - A_B^{-1} * A_N * x_N$$

Целевую функцию аналогичным образом можно записать через свободные переменные:

$$z = c_B^T * x_B + c_N^T * x_N =$$
$$= c_B^T * \left(A_B^{-1} * b - A_B^{-1} * A_N * x_N\right) + c_N^T * x_N =$$
$$= c_B^T * A_B^{-1} * b + \left(c_N^T - c_B^T * A_B^{-1} * A_N\right) * x_N$$

Если вспомнить признак оптимальности задачи линейного программирования в симплексном методе, становится понятно, что если $c_N^T - c_B^T * A_B^{-1} * A_N \geq 0$, то найденное базисное допустимое решение x_B является оптимальным, а оптимальное значение целевой функции при этом равно $c_B^T * A_B^{-1} * b$.

Примеры решения заданий

Задание 3.1. Приведите задачу к каноническому виду и решите ее симплексным методом. С обоснованием укажите количество базисных оптимальных решений.

$$z = x_1 + 8x_2 + 9x_3 + 9x_4 \to min$$
$$x_1 + 2x_2 + x_3 - x_4 \geq 24$$
$$-x_1 + x_2 + 3x_3 + x_4 = -16$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Решение задания 3.1:

Приведем задачу к каноническому виду на минимум с неотрицательной правой частью:

$$z = x_1 + 8x_2 + 9x_3 + 9x_4 \to min$$
$$x_1 + 2x_2 + x_3 - x_4 - x_5 = 24$$
$$x_1 - x_2 - 3x_3 - x_4 = 16$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Шаг 1: в данной системе в матрице специальных ограничений нет ни одного единичного столбца, поэтому введем искусственный базис и вспомогательную целевую функцию:

$$z = x_1 + 8x_2 + 9x_3 + 9x_4 \to min$$
$$x_1 + 2x_2 + x_3 - x_4 - x_5 + y_1 = 24$$

$$x_1 - x_2 - 3x_3 - x_4 + y_2 = 16$$

$$u = y_1 + y_2 \to min$$

$$x_1, x_2, x_3, x_4, x_5, y_1, y_2 \geq 0$$

Запишем симплексную таблицу:

	x_1	x_2	x_3	x_4	x_5	y_1	y_2	b
y_1	1	2	1	–1	–1	1	0	24
y_2	1	–1	–3	–1	0	0	1	16
z	–1	–8	–9	–9	0	0	0	0
u	2	1	–2	–2	–1	0	0	40

Шаг 2: в «свободных» столбцах строки u есть положительные элементы (в первом и втором столбцах), поэтому переходим к шагу 3.

Шаг 3: если мы выберем в качестве ведущего первый столбец, то будем вводить в базис переменную x_1 вместо той переменной, для которой минимально среди неотрицательных отношение $\frac{b_i}{a_{i1}}$, то есть вместо y_2. Ведущий элемент будет равен единице. Если же мы выберем в качестве ведущего второй столбец, то будем вводить в базис переменную x_2 вместо той переменной, для которой минимально среди неотрицательных отношение $\frac{b_i}{a_{i2}}$, то есть вместо y_1. Ведущий элемент будет равен двойке. С единицей в качестве ведущего элемента проводить расчеты легче, поэтому выбираем в качестве ведущего столбца первый столбец, а в качестве ведущей строки – строку y_2.

Шаг 4: проводим пересчет, первый столбец становится единичным, переменная y_2 выходит из базиса, переменная x_1 входит в базис. Выписываем преобразованную симплекс-таблицу и переходим к шагу 2:

	x_1	x_2	x_3	x_4	x_5	y_1	y_2	b
y_1	0	3	4	0	–1	1	–1	8
y_2	1	–1	–3	–1	0	0	1	16
z	0	–9	–12	–10	0	0	1	16
u	0	3	4	0	–1	0	–2	8

Шаг 2: в «свободных» столбцах строки u есть положительные элементы (во втором и третьем столбцах), поэтому переходим к шагу 3.

Шаг 3: если мы выберем в качестве ведущего второй столбец, то будем вводить в базис переменную x_2 вместо y_1. Ведущий элемент будет равен трем. Если же мы выберем в качестве ведущего третий столбец, то будем вводить в базис переменную x_3 вместо y_1. Ведущий элемент будет равен четырем. С четверкой в качестве ведущего элемента проводить расчеты легче, поэтому выбираем в качестве ведущего столбца третий столбец, а в качестве ведущей строки – строку y_1.

Шаг 4: проводим пересчет, третий столбец становится единичным, переменная y_1 выходит из базиса, переменная x_3 входит в базис. Выписываем преобразованную симплекс-таблицу и переходим к шагу 2:

	x_1	x_2	x_3	x_4	x_5	y_1	y_2	b
x_3	0	$\frac{3}{4}$	1	0	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	2
x_1	1	$\frac{5}{4}$	0	–1	$-\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	22
z	0	0	0	–10	–3	3	–3	40
u	0	0	0	0	0	–1	–1	0

Шаг 2: в «свободных» столбцах строки u нет положительных элементов, это значит, что решение вспомогательной задачи окончено. Поскольку оптимальное значение вспомогательной целевой функции равно нулю, то исходная задача допустима, и мы нашли ее допустимое базисное решение $(x_1, x_2, x_3, x_4, x_5) = (22;0;2;0;0)$. Вычеркнув из таблицы строку u и столбцы, соответствующие искусственным переменным, получаем исходную таблицу для основной задачи и переходим к шагу 5:

	x_1	x_2	x_3	x_4	x_5	b
x_3	0	$\frac{3}{4}$	1	0	$-\frac{1}{4}$	2
x_1	1	$\frac{5}{4}$	0	–1	$-\frac{3}{4}$	22
z	0	0	0	–10	–3	40

Шаг 5: в «свободных» столбцах строки z нет положительных элементов, это значит, что решение исходной задачи окончено. Найдено оптимальное значение целевой функции $z^* = 40$ и базисное оптимальное решение $(22;0;2;0;0)$. При этом в свободном столбце x_2 в строке целевой функции стоит ноль, а это значит, что найденное решение неединственно. Поскольку в базисных строках столбца x_2 есть положительные элементы, то оптимальное множество ограничено и представляет собой выпуклую линейную комбинацию всех базисных оптимальных решений. В данном случае их два, и для отыскания второго надо проделать шаг 6, выбрав в качестве ведущего столбца столбец x_2. Ведущей строкой при этом будет строка x_3. Таблица после пересчета:

	x_1	x_2	x_3	x_4	x_5	b
x_2	0	1	$\frac{4}{3}$	0	$-\frac{1}{3}$	$\frac{8}{3}$
x_1	1	0	$\frac{5}{3}$	-1	$-\frac{1}{3}$	$\frac{56}{3}$
z	0	0	0	-10	-3	40

Второе базисное оптимальное решение имеет вид $\left(\frac{56}{3};\frac{8}{3};0;0;0\right)$. Таким образом, ответ к задаче в каноническом виде имеет вид:

$$z^* = 40$$

$$x^* = a\begin{pmatrix}22\\0\\2\\0\\0\end{pmatrix} + (1-a)\begin{pmatrix}56/3\\8/3\\0\\0\\0\end{pmatrix}, a \in [0,1]$$

Отметим, что другим способом получения ответа сразу после нахождения первого оптимального базисного решения является не пересчет для поиска второго базисного решения, а параметризация свободной переменной x_2, выражение базисных переменных через параметр и отыскание границ его изменения:

$$\begin{cases} x_1 = 22 - \frac{5}{4}t \geq 0 \\ x_2 = t \geq 0 \\ x_3 = 2 - \frac{3}{4}t \geq 0 \\ x_4 = 0 \\ x_5 = 0 \\ t \in \left[0, \frac{8}{3}\right] \\ z^* = 40 \end{cases}$$

Несложно убедиться, что это разные записи одного и того же множества точек в $\mathbb{R}^5$. Окончательно получаем следующий ответ к первоначальной задаче, в которой не было переменной x_5 (напомним, что она была поставлена в пространстве $\mathbb{R}^4$):

$$z^* = 40$$

$$x^* = a\begin{pmatrix} 22 \\ 0 \\ 2 \\ 0 \end{pmatrix} + (1-a)\begin{pmatrix} 56/3 \\ 8/3 \\ 0 \\ 0 \end{pmatrix}, a \in [0,1] \text{ или } x^* = \begin{pmatrix} 22 \\ 0 \\ 2 \\ 0 \end{pmatrix} + t \cdot \begin{pmatrix} -5/4 \\ 1 \\ -3/4 \\ 0 \end{pmatrix}, t \in \left[0, \frac{8}{3}\right]$$

Итак, в задаче бесконечно много оптимальных решений (они образуют отрезок в $\mathbb{R}^4$) и при этом ровно два базисных оптимальных решения (они соответствуют концам отрезка и могут быть найдены подстановкой граничных значений параметра).

Задания для самостоятельного решения

Задание 3.2. Выразите в явном виде коэффициенты $d_0, d_1, \ldots, d_n$ в формуле (3.4) через коэффициенты исходной задачи.

Задание 3.3. В данной симплексной таблице последняя строка отвечает целевой функции, исследуемой на минимум, последний столбец – свободным членам. Переменная под номером i выводится из базиса и переменная под номером j вводится в базис. Укажите все возможные пары $(i; j)$.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_6	0	6	0	3	3	1	9
x_3	0	3	1	–8	1	0	4
x_1	1	–5	0	4	2	0	6
z	0	2	0	–3	4	0	1

Задание 3.4. В данной симплексной таблице последняя строка отвечает целевой функции, исследуемой на минимум, последний столбец – свободным членам. Эта таблица является финальной для решения ЗЛП, укажите все неверно заполненные клетки. Клетка определяется указанием номера строки (1–4) и столбца (1–7).

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_4	1	0	1	1	1	0	–1
x_2	3	1	–6	0	0	0	2
x_5	–4	0	5	0	1	1	5
z	–5	–2	3	0	0	0	–1

Задание 3.5. Приведите задачи к каноническому виду и решите симплексным методом:

3.5. $$\begin{cases} z = 11x_1 + x_2 + 2x_3 + 4x_4 \to min \\ x_1 - x_2 - 2x_3 + 2x_4 \geq 12 \\ x_1 + x_2 + x_3 - x_4 \geq 18 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

3.5’. $$\begin{cases} z = 11x_1 + x_2 + 2x_3 + 4x_4 \to min \\ x_1 + x_2 - 2x_3 + 2x_4 = 1 \\ x_1 - x_2 + x_3 - x_4 \geq 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

3.5’’. $$\begin{cases} z = 11x_1 - 12x_2 + 4x_3 + 3x_4 \to min \\ x_1 - x_2 - x_3 + x_4 \geq 6 \\ x_1 - 2x_2 + 2x_3 - x_4 \geq 12 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

Задание 3.6. Приведите задачи к каноническому виду и решите симплексным методом:

3.6. $$\begin{cases} z = 11x_1 - 12x_2 + 4x_3 + 3x_4 \to min \\ x_1 - x_2 - x_3 + x_4 \geq 6 \\ x_1 - 2x_2 + 2x_3 - x_4 \geq 12 \\ x_1 \geq 0, \qquad x_3 \geq 0, x_4 \geq 0 \end{cases}$$

3.6'. $$\begin{cases} z = 11x_1 + 12x_2 + 4x_3 - 3x_4 \to max \\ -x_1 - x_2 + x_3 + x_4 \geq -6 \\ x_1 + 2x_2 + 2x_3 + x_4 \leq 12 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

3.6''. $$\begin{cases} z = 11x_1 + 12x_2 + 4x_3 + 3x_4 \to max \\ x_1 + x_2 - x_3 - x_4 = 2 \\ x_1 + 2x_2 + 2x_3 + x_4 \leq 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

Задание 3.7. В задаче линейного программирования, заданной симплексной таблицей, укажите все базисные оптимальные решения. Целевая функция исследуется на минимум.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	1	1	–2	0	3	2
x_1	1	2	0	–2	0	1	5
x_5	0	–3	0	7	1	–3	5
z	0	2	0	–4	0	0	1

Задание 3.8. В задаче линейного программирования, заданной симплексной таблицей, укажите все оптимальные решения. Целевая функция исследуется на минимум.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_3	0	–1	1	–1	0	3	6
x_1	1	1	0	–2	0	–6	4
x_5	0	–2	0	–4	1	–3	7
z	0	–1	0	0	0	3	5

Задание 3.9. Найдите все оптимальные решения в задаче линейного программирования, заданной симплексной таблицей, рассмотрите два случая: $c = 0$ и $c = 2$. Целевая функция исследуется на минимум.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_6	0	7	0	3	3	1	9
x_3	0	2	1	–1	1	0	c
x_1	1	–2	0	4	2	0	6
z	0	4	0	–3	2	0	7

Задание 3.10. Найдите все оптимальные решения в задаче линейного программирования, заданной симплексной таблицей, рассмотрите два случая: $c = 0$ и $c = 2$. Целевая функция исследуется на минимум.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_4	–2	0	2	1	0	2	c
x_2	–10	1	4	0	0	6	18
x_5	–4	0	10	0	1	6	18
z	–4	0	4	0	0	4	5

Задание 3.11. Приведите задачу к каноническому виду и решите ее симплексным методом. С обоснованием укажите количество базисных оптимальных решений.

$$z = 5x_1 + 35x_2 - x_3 - 5x_4 \to max$$

$$x_1 + 4x_2 - 2x_3 - x_4 \le 3$$

$$x_1 + 7x_2 + x_3 + 2x_4 \le 4$$

$$x_1, x_2, x_3, x_4 \ge 0$$

Задание 3.12. Найдите все оптимальные решения в задаче линейного программирования, заданной симплексной таблицей. Целевая функция исследуется на минимум. С обоснованием укажите количество базисных оптимальных решений.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_1	1	0	0	1	2	–3	3
x_2	0	1	0	3	0	–5	6
x_3	0	0	1	1	–1	–1	0
z	0	0	0	4	–1	–6	0

Задание 3.13. Найдите все оптимальные решения в задаче линейного программирования, заданной симплексной таблицей. Целевая функция исследуется на минимум. С обоснованием укажите количество базисных оптимальных решений.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_6	1	–1	0	–1	0	1	2
x_3	2	–3	1	0	0	0	5
x_5	–1	0	0	1	1	0	1
z	2	–4	0	2	0	0	10

Задание 3.14. Приведите задачу к каноническому виду и решите ее симплексным методом (разберитесь с проблемой вырождения с помощью введения параметра ε_i).

$$\begin{aligned}
&z = x_1 + 4x_2 \to max\\
&x_1 + 2x_2 \le 6\\
&2x_1 + x_2 \le 6\\
&2x_1 - x_2 \le 6\\
&x_1 - x_2 \le 3\\
&x_1, x_2 \ge 0
\end{aligned}$$

Задание 3.15. Приведите задачу к каноническому виду и решите ее симплексным методом. С обоснованием укажите количество базисных оптимальных решений.

$$\begin{cases}
z = x_1 + x_2 + x_3 \to max\\
\qquad\quad x_2 + 3x_3 \ge 4\\
\;\; x_1 + \;\; x_2 - 4x_3 \ge 18\\
4x_1 + 2x_2 + \;\; x_3 \le 9\\
\qquad\quad x_i \ge 0
\end{cases}$$

Задание 3.16. Приведите задачу к каноническому виду и решите ее симплексным методом. С обоснованием укажите количество базисных оптимальных решений.

$$\begin{cases} z = -4x_1 - 2x_2 \to min \\ 3x_1 - \quad x_2 \le 3 \\ -2x_1 + \quad x_2 \le 2 \\ 4x_1 + 2x_2 \le 6 \\ x_1 \ge 0 \end{cases}$$

Задание 3.17. В задаче линейного программирования, заданной симплексной таблицей, найдите все оптимальные решения (целевая функция исследуется на минимум). Укажите количество базисных оптимальных решений. Представьте оптимальное множество задачи в виде суммы многогранника и конуса.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_4	0	3	1	1	–1	0	2
x_6	0	1	2	0	–1	1	6
x_1	1	2	-3	0	1	0	6
z	0	–5	0	0	0	0	7

Задание 3.18. В задаче линейного программирования, заданной симплексной таблицей, найдите все оптимальные решения (целевая функция исследуется на минимум). Укажите количество базисных оптимальных решений. Представьте оптимальное множество задачи в виде суммы многогранника и конуса.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_6	0	1	2	–3	0	1	3
x_1	1	–1	1	2	0	0	8
x_5	0	–1	3	1	1	0	2
z	0	0	–5	0	0	0	7

Задание 3.19. Дана симплекс-таблица для задачи линейного программирования в каноническом виде на минимум:

	x_1	x_2	x_3	x_4	b
x_3	a_{11}	a_{12}	1	0	b_1
x_4	a_{21}	a_{22}	0	1	b_2
z	c_1	c_2	0	0	0

Известно, что

$$b_1 > 0,\ b_2 > 0,\ c_1 > 0,\ c_2 < 0,\ a_{11} > 0,\ a_{12} > 0,\ a_{21} > 0,\ a_{22} < 0,\ \frac{b_1}{a_{11}} > \frac{b_2}{a_{21}}.$$

С точки зрения логики прямого симплексного метода полностью формально обоснуйте выбор ведущего элемента для перехода к новому базису. Требуется не описание алгоритма, а его обоснование.

ГЛАВА 4
Теория двойственности и анализ чувствительности

Вспомним задачу планирования производства (1.3′):

$$\begin{cases} \sum_{j=1}^{n} c_j x_j \to max \\ \sum_{j=1}^{n} a_{ij} x_j \le b_i, i = 1,\dots,m \\ x_j \ge 0, j = 1,\dots,n \end{cases}$$

Здесь:
n – количество выпускаемых продуктов;
m – количество используемых ресурсов;
a_{ij} – затраты i-го ресурса для выпуска единицы j-го продукта;
c_j – прибыль от единицы j-го продукта;
b_i – количество имеющегося i-го ресурса;
x_j – объем выпуска j-го продукта.

Как изменятся оптимальная прибыль или издержки при изменении экзогенных параметров модели, например b_i? Как получить уравнение спроса на ресурсы? Чтобы отвечать на такие вопросы, недостаточно уметь решать задачу линейного программирования. Надо изучить теорию двойственности, а для этого сначала научиться строить *двойственную задачу*. Для любой ЗЛП, записанной в общей форме, можно по определенным правилам построить другую задачу, называемую двойственной. Исходная задача при этом называется *прямой*. Прежде чем учиться строить двойственную задачу, «прорекламируем», зачем она нужна:

- чтобы проводить экономический, содержательный анализ решений ЗЛП (например, анализ двойственной задачи позволяет узнать, как изменится целевая функция, если изменить ограничение, – это называется анализом устойчивости);

- чтобы построить уравнение спроса на ресурс (зависимость необходимого в оптимуме количества ресурса от его цены);
- в некоторых случаях двойственную задачу намного проще решить, чем исходную задачу, а по решению одной легко находится решение другой.

Итак, теория двойственности изучает существующую глубокую связь между прямой и двойственной задачами, и мы начнем с того, что научимся строить такую двойственную задачу для ЗЛП в общей форме. Напомним, что любую ЗЛП несложно записать в общей форме, для этого достаточно согласовать знаки неравенств в ограничениях с типом экстремума: в задаче на максимум должны присутствовать только неравенства типа «меньше либо равно», в задаче на минимум – только неравенства типа «больше либо равно».

Определение (для задачи на максимум). Пусть дана задача (1) линейного программирования в общей форме на максимум. Тогда двойственной к ней задачей линейного программирования называется задача (2) на минимум.

$$\begin{cases} z=\sum_{j=1}^{n} c_j x_j \to max \\ \sum_{j=1}^{n} a_{ij} x_j \le b_i,\ i=1\ldots k \\ \sum_{j=1}^{n} a_{ij} x_j = b_i,\ i=k+1\ldots m \\ x_j \ge 0,\ j=1\ldots l \\ x_j - \text{произв.},\ j=l+1\ldots n \end{cases} (1) \ \Leftrightarrow \ \begin{cases} u=\sum_{i=1}^{m} b_i y_i \to min \\ y_i \ge 0,\ i=1\ldots k \\ y_i - \text{произв.},\ i=k+1\ldots m \\ \sum_{i=1}^{m} a_{ij} y_i \ge c_j,\ j=1\ldots l \\ \sum_{i=1}^{m} a_{ij} y_i = c_j,\ j=l+1\ldots n \end{cases} (2)$$

Определение (для задачи на минимум). Пусть дана задача (1) линейного программирования в общей форме на минимум. Тогда двойственной к ней задачей линейного программирования называется задача (2) на максимум.

$$\begin{cases} z=\sum_{j=1}^{n} c_j x_j \to min \\ \sum_{j=1}^{n} a_{ij} x_j \ge b_i,\ i=1\ldots k \\ \sum_{j=1}^{n} a_{ij} x_j = b_i,\ i=k+1\ldots m \\ x_j \ge 0,\ j=1\ldots l \\ x_j - \text{произв.},\ j=l+1\ldots n \end{cases} (1) \ \Leftrightarrow \ \begin{cases} u=\sum_{i=1}^{m} b_i y_i \to max \\ y_i \ge 0,\ i=1\ldots k \\ y_i - \text{произв.},\ i=k+1\ldots m \\ \sum_{i=1}^{m} a_{ij} y_i \le c_j,\ j=1\ldots l \\ \sum_{i=1}^{m} a_{ij} y_i = c_j,\ j=l+1\ldots n \end{cases} (2)$$

Те же самые определения можно записать в матричном виде: для задачи (1’) двойственной будет задача (2’):

$$\begin{cases} z = cx \to max \\ \quad Ax \le b \qquad (1') \\ \quad x \ge 0 \end{cases} \qquad \Leftrightarrow \qquad \begin{cases} u = yb \to min \\ \quad yA \ge c \qquad (2') \\ \quad y \ge 0 \end{cases}$$

В матричном виде верно, что x – вектор-столбец размера $n \times 1$, y – вектор-строка размера $1 \times m$, A – матрица $m \times n$, c – вектор-строка размера $1 \times n$, b – вектор-столбец размера $m \times 1$.

Сформулируем *правила построения двойственной задачи*.

- Тип экстремума меняется на противоположный (максимум на минимум, и наоборот). Знак неравенств в ограничениях всегда соответствует типу экстремума.
- Число переменных прямой задачи равно числу специальных ограничений двойственной задачи, и наоборот.
- Вектор c коэффициентов целевой функции прямой задачи становится вектором специальных ограничений двойственной задачи, и наоборот, вектор b специальных ограничений прямой задачи становится вектором коэффициентов целевой функции двойственной задачи.
- Матрица A специальных ограничений задачи транспонируется, то есть матрицей специальных ограничений двойственной задачи является матрица A^T.
- Если на переменную x_j прямой задачи наложено условие неотрицательности, то j-е специальное ограничение двойственной задачи является неравенством. Если же на переменную x_j не наложено условие неотрицательности, то j-е специальное ограничение является равенством.
- Если i-е специальное ограничение прямой задачи является неравенством, то на переменную y_i двойственной задачи накладывается условие неотрицательности. Если же i-е специальное ограничение является равенством, то на переменную y_i условие неотрицательности не накладывается – она может иметь любой знак.

Вернемся к задаче планирования производства (1.3′) и выпишем двойственную к ней задачу:

$$\begin{cases} z = \sum_{j=1}^{n} c_j x_j \to max \\ \sum_{j=1}^{n} a_{ij} x_j \le b_i,\ i = 1,\ldots,m \\ x_j \ge 0,\ j = 1,\ldots,n \end{cases} \qquad \begin{cases} u = \sum_{i=1}^{m} b_i y_i \to min \\ \sum_{i=1}^{m} a_{ij} y_i \ge c_j,\ j = 1,\ldots,n \\ y_i \ge 0,\ i = 1,\ldots,m \end{cases}$$

Сколько минимально должны стоить ресурсы (какова должна быть оценка их стоимости), чтобы нам стало выгодно их продавать, а не использовать для производства? То есть чтобы доход от их продажи был не меньше, чем доход от реализации продукции? Для ответа на этот вопрос посмотрим на двойственную задачу. Ее экономический смысл состоит в минимизации суммарной оценки ресурсов, и переменная y_i – это искомая оценка i-го ресурса (минимальная цена, при которой продать его станет выгоднее, чем использовать).

Отметим, что если исходно взять в качестве прямой задачи задачу (2) и выписать к ней двойственную задачу, то она совпадет с исходной задачей (1). Этот факт отражает более общую связь между прямой и двойственной задачами, выражаемую следующей теоремой.

Теорема. Задача, двойственная к двойственной задаче линейного программирования, совпадает с прямой.

Таким образом, все задачи линейного программирования разбиваются на пары взаимно двойственных задач, и их так и изучают – вместе, по парам. Изучение ЗЛП по парам – предмет теории двойственности, и центральными результатами этой теории служат три теоремы двойственности, устанавливающие связь между оптимальными решениями пары взаимно двойственных задач. Прежде чем изучать эти три теоремы, нам понадобится важная вспомогательная лемма.

Лемма (свойство взаимно двойственных задач). Пусть есть пара взаимно двойственных задач $z(x) \to min$ и $u(y) \to max$. Если x и y – их допустимые решения, то $z(x) \ge u(y)$.

Как следствие, если прямая задача является неограниченной, то двойственная задача – недопустимая (множество ее допустимых решений пустое). Обратное неверно: можно привести пример пары взаимно двойственных задач, каждая из которых недопустима (сделайте это сами в качестве упражнения).

Отметим, что для конкретных постановок задач линейного программирования данная лемма имеет и конкретную экономическую интерпретацию. Например, для задачи планирования производства (1.3′) утверждение леммы означает, что для любого допустимого плана производства и любого допустимого вектора оценок ресурсов стоимость изготовленного продукта не превосходит совокупной оценки ресурсов.

Переходим к формулировкам трех теорем двойственности. Первая теорема связывает оптимальные значения целевых функций взаимно двойственных задач и может быть сформулирована в «слабой» и «сильной» постановках. Несмотря на названия, эти постановки независимы в том смысле, что никакая из них не является прямым следствием другой.

Первая теорема двойственности, «слабая» постановка. Если $\tilde{x}$ и $\tilde{y}$ – допустимые решения прямой и двойственной ЗЛП и при этом $z(\tilde{x}) = u(\tilde{y})$, то $\tilde{x}$ и $\tilde{y}$ – оптимальные решения соответствующих задач.

Первая теорема двойственности, «сильная» постановка. Если прямая ЗЛП разрешима, то двойственная к ней тоже разрешима, и оптимальные значения их целевых функций совпадают.

Как следствие «сильной» постановки, если одна из пары взаимно двойственных задач неразрешима, то и вторая неразрешима.

Итак, в результате леммы и первой теоремы двойственности получаем, что для пары взаимно двойственных задач возможны следующие варианты:

- обе задачи недопустимы;
- одна задача не ограничена, другая задача недопустима;
- обе задачи разрешимы.

Основной теоремой теории двойственности является вторая теорема, называемая также теоремой о дополняющей нежесткости. Она, в частности, позволяет найти оптимальное решение одной из пары взаимно двойственных задач, если известно решение другой задачи. Также с ее помощью можно проверить, не решая саму задачу, является ли некоторый вектор оптимальным (если, например, по каким-то содержательным соображениям у нас есть такое подозрение). Дадим ее формулировку.

Вторая теорема двойственности. Пусть $\tilde{x}$ и $\tilde{y}$ – допустимые решения прямой и двойственной ЗЛП. Тогда $\tilde{x}$ и $\tilde{y}$ являются оптимальными решениями прямой и двойственной ЗЛП в том и только том случае, когда выполняются условия

$$\begin{cases} \tilde{y}_i \cdot \left(\sum_{j=1}^{n} a_{ij}\tilde{x}_j - b_i \right) = 0, \; i = 1 \ldots m \\ \left(c_j - \sum_{i=1}^{m} a_{ij}\tilde{y}_i \right) \cdot \tilde{x}_j = 0, \; j = 1 \ldots n \end{cases}$$

Эти условия называются «условиями дополняющей нежесткости». Поскольку произведение двух сомножителей равно нулю, то как минимум один из сомножителей должен быть нулевым. Следовательно, если про какой-то из сомножителей мы знаем, что он ненулевой, то это влечет равенство нулю другого сомножителя. А именно, если какое-либо неравенство системы ограничений одной задачи не обращается в строгое равенство в оптимальной точке, то соответствующая компонента оптимального решения двойственной задачи должна равняться нулю. И наоборот: если какая-либо компонента оптимального решения одной из задач не равна нулю, то соответствующее ограничение в двойственной задаче в оптимуме обращается в строгое равенство.

Вновь отметим, что для конкретных постановок теорема имеет и конкретную экономическую интерпретацию. Например, для задачи планирования производства (1.3’) утверждение теоремы означает, что:

- если при некотором (оптимальном) плане производства расход i-го ресурса строго меньше его запаса b_i (то есть ресурс не расходуется полностью, он есть в наличии с избытком), то и соответствующая оптимальная двойственная оценка этого ресурса равна нулю;
- если при некотором (оптимальном) решении двойственной задачи оценка i-го ресурса строго больше нуля, то в оптимальном плане производства расход соответствующего ресурса равен его запасу (то есть такой ресурс расходуется полностью).

Резюмируя, можно сказать, что двойственные оценки являются показателем дефицитности ресурсов. Дефицитный (полностью используемый при оптимальном плане производства) ресурс имеет положительную оценку, а избыточный (не полностью используемый) ресурс имеет нулевую оценку. Напомним, что под оценкой ресурса мы понимаем стоимость, по которой его выгодно продавать, а не использовать для производства.

Переходим к третьей теореме двойственности, позволяющей найти изменение оптимального значения целевой функции задачи линейного программирования, вызванное малым изменением правой части ограничения этой задачи.

Третья теорема двойственности. Пусть прямая и двойственная задачи имеют оптимальные решения x^* и y^* соответственно. Если при изменении в одном из ограничений прямой задачи b_i на $b_i' = b_i + \Delta b_i$ вектор y^* остается оптимальным решением двойственной задачи, то приращение оптимального значения целевой функции исходной задачи равно

$$\Delta z^* = \mathrm{y}_i^* \cdot \Delta b_i$$

Отметим, что теорема работает только для малых изменений правой части ограничения прямой задачи, то есть для таких, которые не влекут смену оптимальной точки двойственной задачи. Чтобы пояснить экономический смысл третьей теоремы двойственности применительно к задаче планирования производства, заметим, что при $\Delta b_i = 1$ приращение целевой функции равно $\Delta z^* = y_i^*$. Таким образом, оптимальное значение оценки ресурса равно изменению целевой функции, которое произойдет при увеличении количества этого ресурса на единицу. Поэтому оценки ресурсов зачастую называют скрытыми, теневыми или маргинальными оценками (ценами) ресурсов.

Доказательства теорем двойственности можно посмотреть в (Юдин, Гольштейн, 1963).

Примеры решения заданий

Задание 4.1. Каждое из указанных решений данной задачи линейного программирования проверьте на оптимальность, используя двойственную задачу и вторую теорему двойственности.

$$z = 4x_1 + x_2 - 2x_3 + 3x_4 \to min$$

$$2x_1 - x_2 + x_3 - 3x_4 \geq -12$$

$$x_1 - x_2 + 2x_3 + 2x_4 = 12$$

$$4x_1 + x_2 + 4x_3 - 2x_4 \geq 18$$

$$x_1 \geq 0, \quad x_3 \geq 0, x_4 \geq 0$$

1) Точка А: (6;–6;0;0) 2) Точка В: (2;2;6;0) 3) Точка С: (0;–2;5;0)

Решение задания 4.1:

Выпишем двойственную задачу к данной:

$$u = -12y_1 + 12y_2 + 18y_3 \to max$$

$$2y_1 + y_2 + 4y_3 \leq 4$$

$$-y_1 - y_2 + y_3 = 1$$

$$y_1 + 2y_2 + 4y_3 \leq -2$$

$$-3y_1 + 2y_2 - 2y_3 \leq 3$$

$$y_1 \geq 0, \quad y_3 \geq 0$$

1) Точка А: $(6;-6;0;0)$.

На основе второй теоремы двойственности: пусть x^0, y^0 – решения прямой и двойственной задач соответственно. Тогда верно следующее:

$$\begin{cases} x_j^0 > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i^0 = c_j \\ \sum_{i=1}^{m} a_{ij} y_i^0 > c_j \Rightarrow x_j^0 = 0 \end{cases}$$

$x_1^0 \neq 0 \Rightarrow$ первое ограничение двойственной задачи выполняется как равенство;

$x_2^0 \neq 0 \Rightarrow$ второе ограничение двойственной задачи выполняется как равенство;

x_3^0, $x_4^0 = 0$, поэтому про третье и четвертое ограничения двойственной задачи ничего утверждать нельзя.

Подставим точку А в ограничения прямой задачи.

Первое ограничение: $12 + 6 > -12 \Rightarrow y_1^0 = 0.$

Второе и третье ограничения прямой задачи выполнены как равенство, поэтому про соответствующие двойственные переменные y_2^0 и y_3^0 ничего утверждать нельзя.

Имеем систему:

$$\begin{aligned} y_2 + \ \ 4y_3 &= 4 \\ y_2 + \ \ \ \ y_3 &= 1 \\ 2y_2 + \ \ 4y_3 &\leq -2 \\ 2y_2 - \ \ 2y_3 &\leq 3 \\ y_3 &\geq 0 \end{aligned}$$

Из первых двух ограничений находим, что $y_3 = 1,\ y_2 = 0$. Проверяем оставшиеся ограничения в форме неравенств и выясняем, что найденные значения не удовлетворяют третьему ограничению.

Следовательно, точка А $(6;-6;0;0)$ не является решением прямой задачи.

2) Точка В: $(2;2;6;0)$.

На основе второй теоремы двойственности:

$x_1^0 \neq 0 \Rightarrow$ первое ограничение двойственной задачи выполняется как равенство;

$x_2^0 \neq 0 \Rightarrow$ второе ограничение двойственной задачи выполняется как равенство;

$x_3^0 \neq 0 \Rightarrow$ третье ограничение двойственной задачи выполняется как равенство;

$x_4^0 = 0$, поэтому про четвертое ограничение двойственной задачи ничего утверждать нельзя.

Подставим точку B в ограничения прямой задачи.

Первое ограничение: $4-2+6>-12 \Rightarrow y_1^0 = 0$.

Третье ограничение: $8+2+24>18 \Rightarrow y_3^0 = 0$.

Второе ограничение прямой задачи выполнено как равенство, поэтому про соответствующую двойственную переменную y_2^0 ничего утверждать нельзя.

Имеем систему:

$$\begin{aligned} y_2 &= 4 \\ y_2 &= 1 \\ 2y_2 &= -2 \\ 2y_2 &\le 3 \end{aligned}$$

Система является несовместной, следовательно, точка $B(2;2;6;0)$ не является решением прямой задачи.

3) Точка C: $(0;-2;5;0)$.

$x_2^0 \ne 0 \Rightarrow$ второе ограничение двойственной задачи выполняется как равенство;

$x_3^0 \ne 0 \Rightarrow$ третье ограничение двойственной задачи выполняется как равенство;

$x_1^0,\ x_4^0 = 0$, поэтому про первое и четвертое ограничения двойственной задачи ничего утверждать нельзя.

Подставим точку C в ограничения прямой задачи.

Первое ограничение: $-2-5>-12 \Rightarrow y_1^0 = 0$.

Второе и третье ограничения прямой задачи выполнены как равенство, поэтому про соответствующие двойственные переменные y_2^0 и y_3^0 ничего утверждать нельзя.

Имеем систему:

$$\begin{aligned} y_2 + 4y_3 &\le 4 \\ y_2 + y_3 &= 1 \\ 2y_2 + 4y_3 &= -2 \\ 2y_2 - 2y_3 &\le 3 \\ y_3 &\ge 0 \end{aligned}$$

Из первых двух ограничений находим, что $y_3 = 0,\ y_2 = -1$. Проверяем оставшиеся ограничения в форме неравенств и выясняем, что найденные значения удовлетворяют всем ограничениям.

Следовательно, точка $C(0;-2;5;0)$ является решением прямой задачи.

Задание 4.2. Дана задача линейного программирования:

$$z = 12x_1 + 12x_2 + 10x_3 - 16x_4 \to min$$
$$x_1 - x_2 + x_3 - x_4 \geq -3$$
$$x_1 + 2x_2 - x_3 - 2x_4 \geq 2$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0$$

(а) Решив графическим способом задачу, двойственную к данной, найдите решение исходной задачи.

(б) Используя двойственную задачу, выясните приращение оптимального значения функции z, если $\Delta b_2 = -6$.

(в) Используя двойственную задачу, выясните приращение оптимального значения функции z, если $\Delta b_2 = 6$.

(г) Используя двойственную задачу, выясните приращение параметра b_1, при котором приращение оптимального значения функции $\Delta z = -6$.

Решение задания 4.2:

(а) Выпишем задачу, двойственную к данной:

$$u = -3y_1 + 2y_2 \to max$$
$$y_1 + y_2 \leq 12$$
$$-y_1 + 2y_2 \leq 12$$
$$y_1 - y_2 \leq 10$$
$$-y_1 - 2y_2 \leq -16$$
$$y_1 \geq 0,\ y_2 \geq 0$$

Решение двойственной задачи может быть найдено графически: $(y_1; y_2) = (2;7)$, $u_{max} = 8$.

Исходя из первой теоремы двойственности, $z_{min} = 8$.

Найдем решение исходной задачи, используя вторую теорему двойственности.

$y_1^0 \neq 0 \Rightarrow$ первое ограничение прямой задачи выполняется как равенство;

$y_2^0 \neq 0 \Rightarrow$ второе ограничение прямой задачи выполняется как равенство.

Подставим найденную точку в ограничения двойственной задачи.

Первое ограничение: $2 + 7 < 12 \Rightarrow x_1^0 = 0$.

Третье ограничение: $2 - 7 < 10 \Rightarrow x_3^0 = 0$.

Второе и четвертое ограничения двойственной задачи выполнены как равенство, поэтому про соответствующие переменные прямой задачи x_2^0, x_4^0 ничего утверждать нельзя.

Имеем систему:

$$\begin{aligned} -x_2 - \quad x_4 &= -3 \\ 2x_2 - \ 2x_4 &= 2 \\ x_2 \geq 0,\ x_4 &\geq 0 \end{aligned}$$

Система имеет решение: $(x_2;\ x_4) = (2;1)$.

Тогда решение прямой задачи: $(x_1;\ x_2;\ x_3;\ x_4) = (0;\ 2;\ 0;1)$.

(б) При данном изменении правой части ограничения прямой задачи оптимальная точка двойственной задачи НЕ меняется, значит, для поиска приращения функции можно использовать третью теорему двойственности:

$$\Delta z = \Delta b_2 \cdot y_2^0 = -6 \cdot 7 = -42$$

(в) При данном изменении правой части ограничения прямой задачи оптимальная точка двойственной задачи меняется, значит, для поиска приращения функции нельзя использовать третью теорему двойственности.

Новое решение двойственной задачи: $(y_1;\ y_2) = (4;\ 8)$, $u_{max} = 52$.

Следовательно, $\Delta u = u_{\max}^1 - u_{\max}^0 = 52 - 8 = 44$.

Так как оптимальные значения прямой и двойственной задач совпадают (по первой теореме двойственности), то в данном случае $\Delta z = 44$.

(г) Если $\Delta z = -6$, то по первой теореме двойственности и $\Delta u = -6$, то есть новое значение целевой функции двойственной задачи равно 2.

Заметим, что вектор-градиент целевой функции двойственной задачи $grad(u) = (-3 + \Delta b_1; 2)$. Следовательно, оптимальными решениями двойственной задачи могут быть только $(y_1;\ y_2) = (2;7)$ и $(y_1;\ y_2) = (4;\ 8)$.

Проверим первый возможный вариант. Если оптимальной осталась точка $(y_1;\ y_2) = (2;7)$, то Δb_1 можно найти, используя третью теорему двойственности: $\Delta z = -6 = \Delta b_1 \cdot y_1^0 = \Delta b_1 \cdot 2 \cdot \Delta b_1 = -3$.

Проверим найденное $\Delta b_1 = -3$: действительно, при таком изменении Δb_1 оптимальным значением остается $(y_1;\ y_2) = (2;7)$. Следовательно, $\Delta b_1 = -3$ подходит.

Проверим второй возможный вариант. Если оптимальной точкой стала точка $(y_1;\ y_2) = (4;\ 8)$, то применять третью теорему двойственности нельзя, но найти Δb_1 можно из следующего уравнения:

$$u^1_{\max} = 2 = (-3 + \Delta b_1) \cdot y_1^0 + 2 \cdot y_2^0 = (-3 + \Delta b_1) \cdot 4 + 2 \cdot 8$$

$$\Delta b_1 = -\frac{1}{2}$$

Проверим найденное $\Delta b_1 = -\frac{1}{2}$: при таком изменении Δb_1 оптимальным значением остается $(y_1; y_2) = (2;7)$. Следовательно, $\Delta b_1 = -\frac{1}{2}$ не подходит.

Задание 4.3.

Дана задача линейного программирования:

$$z = 41x_1 - 7x_2 + 11x_3 + 9x_4 \rightarrow min$$

$$4x_1 + ax_2 + 4x_3 + x_4 = 5 (= b_1)$$

$$5x_1 - x_2 - x_3 + x_4 \geq -1$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

При $a = -2$ решите пункты (а) – (д):

(а) Изобразите допустимую область задачи, двойственной к данной.

(б) Решите графическим способом задачу, двойственную к данной.

(в) Решите исходную задачу с помощью решения двойственной задачи и второй теоремы двойственности.

(г) Не решая заново исходную и двойственную задачи, объясните, как изменится оптимальное значение целевой функции исходной задачи при увеличении b_1 на 2.

(д) Найдите диапазон изменения Δb_1, допускающий применение третьей теоремы двойственности. Дайте необходимые пояснения.

(е) Найдите диапазон изменения a, при котором допустимая область двойственной задачи является неограниченной.

Решение задания 4.3:

(а) Выпишем двойственную задачу к данной:

$$u = 5y_1 - y_2 \rightarrow max$$

$$4y_1 + 5y_2 \leq 41$$

$$-2y_1 - y_2 \leq -7$$

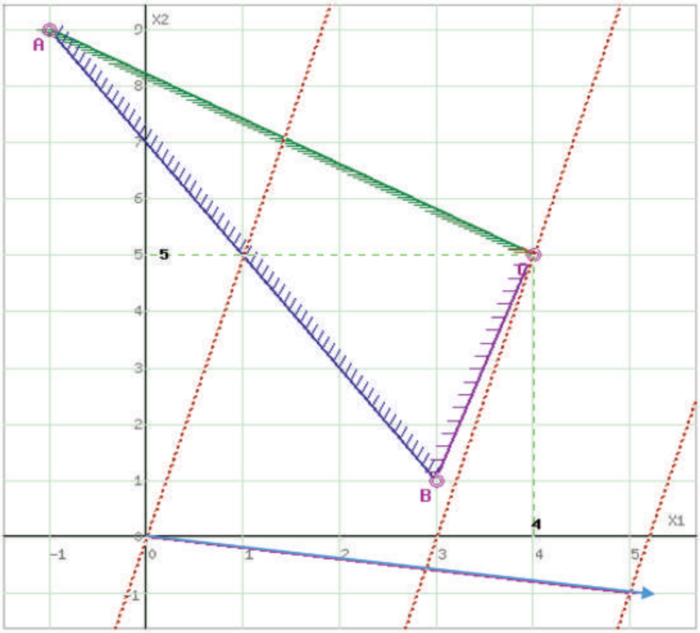
$$4y_1 - y_2 \leq 11$$

$$y_1 + y_2 \leq 9$$

$$y_2 \geq 0$$

Заметим, что на знак y_1 не наложено условие неотрицательности, поскольку первое ограничение исходной задачи является равенством.

Изобразим допустимую область двойственной задачи (треугольник $ABC, A=(-1,9), B=(3,1), C=(4,5)$), а также линии уровня целевой функции (красный пунктир):



(б) Решение двойственной задачи может быть найдено графически: точка $C=\left(y_1^0;\, y_2^0\right)=(4;5)$, $u_{max}=15$.

(в) Исходя из первой теоремы двойственности, $z_{min}=15$.

Найдем решение исходной задачи, используя вторую теорему двойственности.

$y_1^0 \neq 0 \Rightarrow$ первое ограничение прямой задачи выполняется как равенство;

$y_2^0 \neq 0 \Rightarrow$ второе ограничение прямой задачи выполняется как равенство.

Подставим найденную точку в ограничения двойственной задачи.

Второе ограничение: $-10-5<-7 \Rightarrow x_2^0=0$.

Первое, третье и четвертое ограничения двойственной задачи выполнены как равенство, поэтому про соответствующие переменные прямой задачи x_1^0, x_3^0, x_4^0 ничего утверждать нельзя.

Имеем систему:

$$4x_1^0 + 4x_3^0 + x_4^0 = 5$$
$$5x_1^0 - x_3^0 + x_4^0 = -1$$
$$x_1^0 \geq 0,\ x_3^0 \geq 0,\ x_4^0 \geq 0$$

Система имеет неединственное решение:

$$x^0 = \begin{pmatrix} -6+5t \\ 0 \\ t \\ 29-24t \end{pmatrix},\ t \in \left[\frac{6}{5}, \frac{29}{24}\right]$$

(г) При увеличении b_1 на 2 не меняется оптимальная точка в двойственной задаче, следовательно, можно применять третью теорему двойственности:

$$\Delta z = \Delta b_1 \cdot y_1^0 = 2 \cdot 4 = 8$$

(д) Условие третьей теоремы двойственности – неизменность оптимального решения двойственной задачи.

Если увеличивать Δb_1 до бесконечности, то линия уровня целевой функции двойственной задачи будет иметь положительный наклон, и максимум будет достигаться как раз в точке C.

Чтобы найти нижнюю границу Δb_1, нужно определить, при каком Δb_1 линия уровня станет параллельна ограничению № 3: это происходит, когда $4 = 5 + \Delta b_1$, то есть $\Delta b_1 = -1$.

Итого, $\Delta b_1 \in [-1, +\infty)$.

(е) При различных a меняется ограничение № 2 двойственной задачи.

Заметим, что при $a \geq 0$ область будет неограниченная, так как ограничение будет иметь положительный наклон, подходить будет все, что выше прямой.

Осталось найти границу на a снизу – она получится из условия параллельности ограничений № 2 и № 1. Это будет тогда, когда $\frac{a}{4} = \frac{-1}{5} \Rightarrow a = -\frac{4}{5}$.

Итого, $a \in [-4/5, +\infty)$.

Задание 4.4. Дана задача линейного программирования – двухпродуктовое производство с ограничениями на ресурсы.

$$z = x_1 + 4x_2 \to max$$
$$2x_1 + 2x_2 \le 60$$
$$x_1 + 2x_2 \le 40$$
$$x_1 \ge 0,\ x_2 \ge 0$$

Какой из предложенных вариантов является наиболее выгодным?

а) Купить 2 тонны 1-го ресурса по цене 1 млн ден. ед. за тонну.

б) Продать 1 тонну 1-го ресурса по цене 1 млн ден. ед. за тонну.

в) Купить 2 тонны 2-го ресурса по цене 1 млн ден. ед. за тонну.

г) Продать 2 тонны 2-го ресурса по цене 2 млн ден. ед. за тонну.

д) Продать 2 тонны 1-го ресурса по цене 2 млн ден. ед. за тонну.

Решение задания 4.4:

Чтобы ответить на вопрос условия, выпишем двойственную задачу, найдем ее решение и будем применять третью теорему двойственности.

Двойственная задача:

$$u = 60y_1 + 40y_2 \to \mathrm{m}in$$
$$2y_1 + y_2 \ge 1$$
$$2y_1 + 2y_2 \ge 4$$
$$y_1 \ge 0, y_2 \ge 0$$

Решением двойственной задачи является точка $(0,2)$ (в данном случае оптимальную точку можно было найти графически, например).

Найдем решение пунктов а) и в), остальные решаются аналогично.

а) Покупка двух тонн 1-го ресурса по цене 1 млн ден. ед. за тонну принесет издержки в размере 2 млн ден. ед. При этом $\Delta b_1 = 2$, новая целевая функция двойственной задачи, имеет вид $u = 62y_1 + 40y_2$ Однако точка оптимума двойственной задачи не изменится (точка $(0,2)$), а значит, можно применить третью теорему двойственности, чтобы найти прирост выручки z и сопоставить его с издержками:

$$\Delta z = y_1 * \Delta b_1 = 0 * 2 = 0$$

Итак, прирост выручки нулевой, а издержки равны 2 млн, следовательно, данный вариант является невыгодным.

в) Покупка двух тонн 2-го ресурса по цене 1 млн ден. ед. за тонну принесет издержки в размере 2 млн ден. ед. При этом $\Delta b_2 = 2$, новая целевая функция двойственной задачи имеет вид $u = 60y_1 + 42y_2$. Однако точка оптимума двойственной задачи не изменится (точка $(0,2)$), а значит, можно применить третью теорему двойственности, чтобы найти прирост выручки z и сопоставить его с издержками:

$$\Delta z = y_2 * \Delta b_2 = 2*2 = 4$$

Итак, прирост выручки равен 4 млн, издержки равны 2 млн, следовательно, данный вариант является выгодным (прибыль равна 2 млн).

Задания для самостоятельного решения

Задание 4.5. Используя определение, напишите задачу, двойственную к данной:

$$z = -2x_1 + x_2 + 5x_3 + 4x_4 \to max$$
$$x_1 + 3x_2 - x_3 + x_4 \geq 3$$
$$2x_1 + x_2 - 3x_3 + x_4 \leq 8$$
$$3x_1 + x_2 - 2x_3 - x_4 = -8$$
$$x_1 \geq 0, \qquad x_4 \geq 0$$

Задание 4.6.

Дана задача линейного программирования:

$$z = 14x_1 + 10x_2 + 8x_3 + 2x_4 \to min$$
$$x_1 + 2x_2 + x_3 + x_4 \geq b_1 = 2$$
$$2x_1 + x_2 + x_3 - x_4 = b_2 = -1$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Решив графическим способом задачу, двойственную к данной, найдите решение исходной задачи.

Задание 4.7.

Дана задача линейного программирования:

$$z = 14x_1 + 10x_2 + 8x_3 + 2x_4 \to min$$
$$x_1 + 2x_2 + x_3 + x_4 \geq b_1 = 2$$
$$2x_1 + x_2 + x_3 - x_4 = b_2 = -1$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

(а) Используя двойственную задачу, выясните приращение оптимального значения функции z, если

1) $\Delta b_1 = 2$ 2) $\Delta b_1 = -2$ 3) $\Delta b_2 = 2$ 4) $\Delta b_2 = -2$

(б) Используя двойственную задачу, выясните приращение параметра b_1, при котором приращение оптимального значения функции z равно:

1) 8 2) -8

(в) Используя двойственную задачу, выясните приращение параметра b_2, при котором приращение оптимального значения функции z равно:

1) 4 2) -4

Задание 4.8. Каждое из указанных решений данной задачи линейного программирования проверьте на оптимальность, используя двойственную задачу и вторую теорему двойственности.

$$z = 2x_1 - 2x_2 + x_3 - 9x_4 \to min$$
$$2x_1 - x_2 + x_3 + 3x_4 = 8$$
$$x_1 + x_2 + 2x_3 + 4x_4 \geq 16$$
$$3x_1 + x_2 + x_3 + 2x_4 \leq 12$$
$$x_1 \geq 0, \quad x_3 \geq 0, x_4 \geq 0$$

1) $(1;1;7;0)$ 2) $(0;0;8;0)$ 3) $(1;3;0;3)$ 4) $(0;4;0;4)$

Задание 4.9. Каждое из указанных решений данной задачи линейного программирования проверьте на оптимальность, используя двойственную задачу и вторую теорему двойственности.

$$z = x_1 + 2x_2 + 5x_3 - 3x_4 \to max$$
$$4x_1 + 2x_2 - 2x_3 + 3x_4 \geq 4$$
$$3x_1 + 4x_2 + x_3 \leq 8$$
$$4x_1 + 3x_2 + 2x_3 + 2x_4 = 24$$
$$x_1 \geq 0, \quad x_3 \geq 0, x_4 \geq 0$$

1) (20; –24; 4; 4) 2) (0; 2; 0; 9) 3) (6; –4; 6; 0)

Задание 4.10. Дана задача линейного программирования:

$$z = 4x_1 - 3x_2 + 2x_3 - 6x_4 \to max$$
$$2x_1 - x_2 + x_3 + 3x_4 = 4$$

$$
\begin{aligned}
&-x_1 + 2x_2 - x_3 + x_4 \geq 4\\
&3x_1 + x_2 + x_3 + 2x_4 \leq 6\\
&x_1 \geq 0, x_3 \geq 0, x_4 \geq 0
\end{aligned}
$$

а) Решите графическим способом задачу, двойственную к данной.

б) Решите исходную задачу с помощью решения двойственной задачи и второй теоремы двойственности.

Задание 4.11. Дана задача линейного программирования:

$$
\begin{aligned}
&z = 4x_1 - 3x_2 + 2x_3 - 6x_4 \to min\\
&2x_1 - x_2 + x_3 + 3x_4 = 8\\
&-x_1 + 2x_2 + x_3 + x_4 \geq 8\\
&3x_1 + x_2 + x_3 + 2x_4 \leq 12\\
&x_1 \geq 0, x_3 \geq 0, x_4 \geq 0
\end{aligned}
$$

а) Решите графическим способом задачу, двойственную к данной.

б) Решите исходную задачу с помощью решения двойственной задачи и второй теоремы двойственности.

Задание 4.12. Дана задача линейного программирования:

$$
\begin{cases}
z = -x_1 + 7a \cdot x_2 + 2a \cdot x_3 \to max\\
-3x_1 + 2x_2 - x_3 \leq 0\\
2x_1 - x_2 + 2x_3 \leq 5\\
x_1 + x_2 + x_3 = 2\\
x_2 \geq 0
\end{cases}
$$

Известно, что $x^* = (-1, 0, 3)$ – оптимальное решение. Найдите возможные значения параметра a.

ГЛАВА 5
Целочисленное программирование

Дискретное программирование – область математики, изучающая экстремальные задачи на не более чем счетных (чаще всего конечных) множествах. Такие задачи часто возникают при моделировании реальных ситуаций и имеют ряд особенностей по сравнению со стандартными непрерывными задачами. Общая постановка имеет вид

$$\begin{cases} f(x) \to extr \\ x \in D \subseteq \mathbb{R}^n, \ D - \text{не более чем счетное} \end{cases}$$

Рассмотрим наиболее частую ситуацию, когда оптимизационная задача решается на множестве конечного объема $|D| = N < \infty$. При небольших N и n можно использовать метод полного перебора. Но даже если переменная x_i бинарная (и $N = 2^n$), то с ростом n объем вычислительной работы резко возрастает, и уже при $n = 20$ (совсем небольшое для реальной задачи число переменных) нужно перебрать более миллиона точек. На современном этапе развития вычислительной техники перебор 2^{30} вариантов занимает секунды, 2^{40} – часы (и не у всех есть такие ресурсы), а перебор 2^{50} вариантов и вовсе невозможен. Ситуация очевидным образом усложняется, если число принимаемых переменной x_i значений больше двух. Таким образом, при больших значениях N и n необходимы специальные методы решения.

Сделаем важное практическое замечание. До недавнего времени считалось, что для снижения вычислительной сложности задачи нужно, по возможности:

- минимизировать число целочисленных переменных (например, аппроксимировать целочисленные переменные непрерывными там, где это возможно, скажем, переходя к дробным значениям рабочего времени – полчеловека не может работать одну смену, но человек может работать половину смены);
- уменьшить интервалы допустимых значений целочисленных переменных (например, отбросив редко встречающиеся варианты);
- избегать нелинейных ограничений, аппроксимируя их линейными.

С одной стороны, эти советы продолжают работать и не теряют своей актуальности. С другой стороны, развитие технологий и рост производительности современных компьютеров повышают эффективность существующих алгоритмов и снижают необходимость следовать этим советам. Возможно, ряд ранее полученных результатов, в том числе в экономических задачах, можно будет в скором времени пересмотреть, сформулировав математическую постановку более точно, не избегая целочисленности и нелинейности.

Прежде чем перейти к описанию специальных методов решения дискретных задач, приведем их основные отличия от классических непрерывных (в первую очередь линейных и выпуклых) задач.

1. *Нерегулярность*. Область допустимых значений невыпукла (и вообще несвязна), что делает неприменимым богатый математический аппарат, разработанный для таких областей. В частности, оптимум не обязательно достигается в крайних точках, а может быть внутренней точкой допустимой области (без учета целочисленности).

2. *Сложность понятия окрестности*. Если x^1 и $x^2 \in D \subseteq \mathbb{R}^n$ близки, то это не значит, что $f\left(x^1\right)$ и $f\left(x^2\right)$ также близки. Близость точек надо оценивать по значениям функции в них.

3. *Сложность поиска допустимого решения*. Понять, является ли разрешимой система ограничений, задающая область D (даже, например, система линейных равенств и неравенств в целых числах), — трудоемкая вычислительная задача сама по себе, зачастую не легче исходной.

4. *Недопустимость округления*. Исторически первые попытки решения задач дискретного программирования были основаны на идее сведения к непрерывной задаче и округлении до ближайшей допустимой точки. Но такая идея часто не работает. Чтобы показать это, рассмотрим такой пример:

$$\begin{gathered}
z = 9x_1 + 7x_2 \to max \\
2x_1 + x_2 \le 9 \\
5x_1 + 4x_2 \le 29 \\
x_1, x_2 \ge 0 \\
x_1, x_2 \in Z
\end{gathered}$$

Игнорируя условие целочисленности, получим $x_1^* = \begin{pmatrix} 2\frac{1}{3} \\ 4\frac{1}{3} \end{pmatrix}, z_1^* = \dfrac{154}{3}$.

Оптимальное решение с учетом целочисленности: $x_2^* = \begin{pmatrix} 1 \\ 6 \end{pmatrix}, z_2^* = 51 = \frac{153}{3}$.

Никаким округлением первого, нецелочисленного решения невозможно получить второе оптимальное целочисленное решение.

5. Кроме того, переменная x_i может не просто выражать число неделимых предметов, но и отражать смысловую бинарность (пол, наличие/отсутствие спроса и т.д.). В этом случае округление содержательно некорректно.

Основные примеры задач

1. ЗЛП, которые формально к дискретным не относятся, но при соответствующих исходных данных обязательно имеют целочисленный оптимум. Примерами служат транспортная задача и ее модификации (ТЗ с ограничениями, задача о назначениях, задача о потоках в сетях). В силу их особенностей и распространенности для них разработаны специальные методы, часть которых описана далее в главе 6, а часть будет рассмотрена в будущих версиях пособия.

2. ЗЛП, в которых переменные представляют собой физически неделимые величины (так называемые задачи с неделимостями). Примерами служат задача распределения ресурсов при планировании выпуска неделимых видов продукции, задача распределения капиталовложений при планировании использования неделимых факторов производства, задача планирования человеческих ресурсов.

3. Задачи, где элементами множества D являются упорядоченные или неупорядоченные выборки из нескольких объектов (комбинаторные задачи). Примерами служат задача о коммивояжере (TSP, travelling salesman problem – поиск оптимального маршрута, проходящего через все города), задача составления оптимального расписания, задача о рюкзаке (knapsack problem – из заданного множества объектов с характеристиками «стоимость» и «вес» выбрать подмножество с максимальной общей стоимостью, соблюдая ограничение на суммарный вес).

Приведем математическую постановку задачи о рюкзаке. Пусть есть n различных товаров, m_k – количество товаров типа $k, k = 1 \ldots n$, которые кладем в рюкзак. У каждого товара есть вес a_k и ценность c_k (числовые параметры в расчете на единицу товара). Оптимизационная задача максимизации ценности рюкзака при ограничении на его суммарный вес (w) имеет вид:

$$z = c_1 m_1 + \ldots + c_n m_n \to max$$
$$a_1 m_1 + \ldots + a_n m_n \le w$$
$$m_k = \{0, 1, 2, 3, \ldots\}$$

4. Задачи об упаковке в контейнеры (BPP, bin packing problem) – требуется упаковать объекты заданной формы в как можно меньшее число контейнеров заданной формы. Существуют различные варианты этой задачи (одномерная/двумерная/трехмерная упаковка, упаковка по весу/стоимости), применяемые в разных областях. Так, одномерная задача упаковки возникла для решения задач раскроя и перевозки материалов. Двумерная задача хорошо моделирует задачу оптимизации размещения объектов (например, автомобилей в вагонах). Трехмерная задача подходит для оптимизации размещения трехмерных объектов на складах. Отметим, что речь может идти не только о логистических задачах, но и, например, о размещении файлов на дисках или о формировании программы учебного курса из отдельных модулей.

Общая постановка и основные методы решения ЗЦЛП

Далее будем рассматривать задачу целочисленного линейного программирования (ЗЦЛП) – одну из наиболее часто встречающихся в приложениях и наиболее изученных задач дискретного программирования. Общая постановка: к задаче линейного программирования в стандартной или канонической форме добавляется условие целочисленности некоторых (возможно, всех) переменных $x_j \in \mathbb{Z}, j = 1, \ldots, n_1, n_1 \le n$. Если $n_1 < n$, то задача называется частично целочисленной, если $n_1 = n$ — полностью целочисленной. Далее будем рассматривать полностью целочисленную задачу:

$$\begin{cases} z = (c, x) = \sum_{j=1}^{n} c_j x_j \to extr \\ x \in D \subseteq \mathbb{R}^n \\ x_j \in Z,\ j = 1 \ldots n \end{cases} \tag{5.1}$$

где $D \subseteq \mathbb{R}^n$ – выпуклый многогранник, задаваемый системой m линейных равенств и неравенств и n условий неотрицательности:

$$Ax = b(\le, \ge), x \ge 0$$

Обозначим через D^c множество целых точек из D: $D^c = \left\{x \in D, x_j \in \mathbb{Z}, j = 1 \ldots n\right\}$. Тогда система (1) эквивалентна системе

$$\begin{cases} z = (c, x) = \sum_{j=1}^{n} c_j x_j \to extr \\ x \in D^c \subseteq \mathbb{R}^n \end{cases} \tag{5.2}$$

Мы умеем решать любую задачу вида

$$\begin{cases} c, x \to extr \\ x \in D' \end{cases}, \tag{5.3}$$

где D' – выпуклый многогранник.

Существуют три основные идеи того, как можно поступить с искомой задачей (5.2), и им соответствуют три группы методов дискретной оптимизации:

№	Идея	Методы	Плюсы и минусы
1	Найти такое D', чтобы решение (5.2) сводилось к решению (5.3)	Отсечения (замена решения исходной задачи решением последовательности непрерывных задач)	Существуют задачи, в которых число отсечений быстро растет с ростом размерности и коэффициентов. Подходят для решения чисто целочисленных задач, которые составляют небольшую часть реальных прикладных задач. Плохо работают для задач с разреженными матрицами ограничений.
2	Перебрать все возможные точки из D и выбрать ту, на которой решение оптимально	Комбинаторные (упорядоченный перебор допустимых решений)	Плохо работают при больших n. Меньше, чем методы отсечений, подвержены ошибкам округления. Очевидна конечность алгоритма.
3	Решить искомую задачу (5.1), отбросив (полностью или частично) условия целочисленности, и потом каким-то разумным способом округлить решение	Приближенные (эвристические)	Нужна аккуратность в понимании окрестности и близости точек.

Никаких других глобальных идей пока не придумано. Принципиальную возможность сведения (5.2) к задаче типа (5.3) демонстрирует следующая теорема.

Теорема (о возможности сведения ЗЦЛП к ЗЛП). Множества оптимальных решений задач (5.2) и (5.4) совпадают (или, что то же самое, совпадают множества оптимальных решений задач (5.1) и (5.4)):

$$\underset{(5.2)}{\begin{cases} z=(c,x)\to extr \\ x\in D^c \end{cases}} \quad \Leftrightarrow \quad \underset{(5.4)}{\begin{cases} z=(c,x)\to extr \\ x\in Y=co\left(D^c\right) \end{cases}}$$

Здесь $co\left(D^c\right)$ – выпуклая оболочка, D^c – наименьшее выпуклое множество, содержащее D^c. Теорема показывает принципиальную возможность сведения (5.2) к задаче типа (5.4), но она не дает алгоритм поиска Y. В настоящее время эффективный алгоритм построения выпуклой линейной оболочки неизвестен, вместо этого для реализации первой идеи используют методы отсечения, основанные на поэтапном построении Y.

На практике для решения ЗЦЛП нередко используются методы третьей группы, позволяющие получить хорошее допустимое решение задачи за приемлемое время. В качестве примера можно привести «жадный» алгоритм решения задачи о рюкзаке.

Мы подробнее рассмотрим методы первой группы (на примере метода Гомори) и методы второй группы (на примере метода ветвей и границ).

Метод отсечений (на примере метода Гомори)

Опишем общую идею метода отсечений. Сначала решается исходная задача без учета условия целочисленности. Если полученное решение является целочисленным, то задача решена, иначе к ее ограничениям добавляется новое *линейное* ограничение (оно называется отсечением), обладающее двумя свойствами:

1) *полученное нецелочисленное оптимальное решение ему не удовлетворяет;*
2) *любое целочисленное допустимое решение ему удовлетворяет.*

Решается задача с этим дополнительно введенным ограничением, и процесс повторяется до получения целочисленного решения (или демонстрации того, что оно не существует). Идея метода приводит к трем проблемам:

- поиск универсального правила построения отсечений;
- доказательство конечности алгоритма;
- чрезмерное «разрастание» размерности задачи при добавлении ограничений.

В 1950-х гг. американским математиком Ральфом Гомори был разработан универсальный и вычислительно реализуемый алгоритм, решающий первые две проблемы. Алгоритм представляет собой (теперь наиболее известную) модификацию симплексного метода для решения ЗЦЛП, называемую методом Гомори. Метод основан на следующей теореме.

Теорема (лемма Гомори). Пусть для полностью целочисленной задачи (5.1) нашлось нецелочисленное базисное оптимальное решение x^0, i-я координата которого не является целой. Пусть ему соответствует следующая симплексная таблица:

	x_1	x_2	...	x_n	b
x_{k_1}	α_{11}	α_{12}	...	α_{1n}	β_1
...	...	...	...	...	...
x_{k_i}	α_{mi}	α_{mi}	...	α_{mi}	β_i
...	...	...	...	...	...
x_{k_m}	α_{m1}	α_{m2}	...	α_{mn}	β_m
z	γ_1	γ_2	...	γ_n	β_0

В данной таблице $\beta_i \notin \mathbb{Z}, \gamma_j \leq 0 \,\forall\, j = 1 \ldots n$.

Обозначим через $N = \{1, \ldots, n\} \setminus \{k_1, \ldots, k_m\}$ множество индексов свободных переменных. Тогда неравенство

$$-\{\beta_i\} + \sum_{j \in N} \{\alpha_{ij}\} x_j \geq 0$$

является правильным отсечением.

Схема алгоритма решения ЗЦЛП методом Гомори

Шаг 0. Решить задачу без учета целочисленности симплексным методом, получить оптимальное решение x^0 или показать, что задача недопустима (или не ограничена), и тогда исходная задача тоже недопустима (не ограничена).

Шаг 1. Если $x_j^0 \in \mathbb{Z}, j = 1,\dots,n$, то x^0 является решением исходной задачи, иначе выбрать $i: x_i^0 \notin \mathbb{Z}$.

Шаг 2. Добавить к последней симплексной таблице (соответствующей x^0) ограничение, построенное по лемме Гомори.

Шаг 3. Решить полученную задачу двойственным симплекс-методом. Если условия ЗЛП несовместны, то исходная задача недопустима, иначе возвращаемся на шаг 1.

Задание 5.1

Дана задача целочисленного линейного программирования:

$$\begin{cases} z = x_1 + x_2 \to max \\ \quad 4x_1 + 4x_2 \le 9 \\ \quad x_i \ge 0, x_i \in Z \end{cases}$$

Решите эту задачу, используя алгоритм метода отсечений Гомори.

Решение задания 5.1

Приведем задачу к каноническому виду на минимум:

$$\begin{cases} \tilde{z} = -z = -x_1 - x_2 \to min \\ \quad 4x_1 + 4x_2 + x_3 = 9 \\ \quad x_i \ge 0,\ x_i \in Z,\ i = 1,2 \end{cases}$$

Решив данную задачу без учета целочисленности симплексным методом, получим следующую финальную симплексную таблицу:

	x_1	x_2	x_3	b
x_1	1	1	1/4	9/4
$\tilde{z}$	0	0	–1/4	–9/4

Оптимальное решение $x^0 = \begin{pmatrix} \frac{9}{4} \\ 0 \\ 0 \end{pmatrix}$ не является целочисленным, поэтому выбираем $i = 1$ и строим отсечение по лемме Гомори. Выпишем уравнение, соответствующее первой строке таблицы, так, чтобы левая часть являлась целочисленной, а правая часть строго не превосходила единицу (так как $\frac{1}{4} < 1$, а $x_3 \ge 0$):

$$x_1 + x_2 - 2 = \frac{1}{4} - \frac{1}{4}x_3 \ (<1)$$

Исходя из этого: $\frac{1}{4} - \frac{1}{4}x_3 \leq 0$. Заменив неравенство «$<1$» на «$\leq 0$», мы «отсекли» часть точек (нам удалось это сделать, пользуясь целочисленностью). Полученное ограничение в точности соответствует отсечению Гомори, построенному по формулировке леммы:

$$-\frac{1}{4} + \frac{1}{4}x_3 \geq 0$$

Введем дополнительную переменную $x_4 \geq 0$, $x_4 \in \mathbb{Z}$ (по построению), чтобы преобразовать ограничение к равенству:

$$-\frac{1}{4} + \frac{1}{4}x_3 - x_4 = 0$$

Добавим построенное дополнительное ограничение в финальную симплексную таблицу.

	x_1	x_2	x_3	x_4	b
x_1	1	1	1/4	0	9/4
x_4	0	0	–1/4	1	–1/4
$\tilde{z}$	0	0	–1/4	0	–9/4

Таблица соответствует базисной недопустимой точке $\begin{pmatrix} \frac{9}{4} \\ 0 \\ 0 \\ -\frac{1}{4} \end{pmatrix}$, в которой выполнен критерий оптимальности. Проведем итерацию двойственного симплекс-метода, который последовательно «приближает» точку к допустимой области, сохраняя условия оптимальности. Выбираем ведущей строку x_4 с недопустимостью (в ней правая часть является отрицательной). Выбираем ведущий элемент, исходя из соображений сохранения критерия оптимальности целевой функции: среди отрицательных a_{ij} выбираем такой, для которого минимальным является соотношение $\frac{c_j}{a_{ij}}$. В нашем случае единственным отрицательным элементом ведущей строки является $-1/4$. После пересчета получаем таблицу:

	x_1	x_2	x_3	x_4	b
x_1	1	1	0	1	2
x_3	0	0	1	–4	1
$\tilde{z}$	0	0	0	–1	–2

Этой таблице соответствует целочисленное базисное решение $x^* = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}$, $z^* = 2$.

Поскольку в строке целевой функции коэффициент при свободной переменной x_2 равен нулю, то решение задачи неединственное:

$$x^* = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} + t \cdot \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

В отличие от непрерывных задач, параметр t принимает только целочисленные значения, в данном случае — $0,1,2$. Таким образом, окончательно получаем, что исходная задача (напомним, что она была поставлена в $\mathbb{R}^2$) имеет три оптимальных решения $\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}$, при этом $z^* = 2$.

Комбинаторные методы (на примере метода ветвей и границ)

Альтернативным подходом к решению ЗЦЛП являются комбинаторные методы, основанные на упорядоченном переборе точек с целочисленными координатами. Отметим, что применение таких методов начинается (так же как и в методах отсечений) с решения исходной задачи без учета целочисленности. При этом такая задача может оказаться недопустимой (или неограниченной), и тогда исходная задача тоже недопустима (не ограничена). В противном случае (когда задача без учета целочисленности разрешима) исходная задача либо также разрешима, либо недопустима (если среди допустимых точек нет целочисленных), но заведомо ограничена. Поэтому можно говорить об упорядоченном переборе *конечного числа* точек с целочисленными координатами.

В 1960 г. американскими математиками Алисой Лэнд и Элисон Дойг был разработан комбинаторный метод решения ЗЦЛП, основанный на двух процедурах: разбиении допустимой области на более мелкие подобласти (ветвлении) и нахождении нижних (верхних при задаче на максимум) оценок целевой функции на этих подобластях (границ). Метод был назван методом ветвей и границ (по-английски BB, B&B или BnB, *branch and bound*). В основе метода лежит следующая идея: если нижняя граница целевой функции на некоторой подобласти больше, чем верхняя граница на какой-либо другой подобласти (верхняя граница меньше нижней для задачи на максимум), то последнюю можно исключить из рассмотрения.

Для формального описания алгоритма метода введем ряд определений.

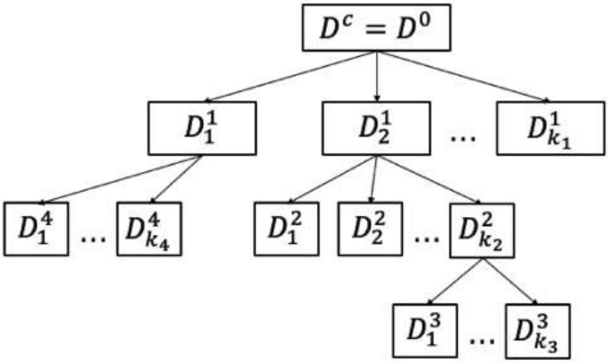
Определение. Рассмотрим задачу (5.2): $\begin{cases} z = (c,x) \to extr \\ x \in D^c \end{cases}$, где через D^c обозначено конечное множество целых точек из выпуклого многогранника D. *Ветвлением* называется разбиение задачи на конечное число подзадач:

$$\begin{cases} z = (c,x) \to extr \\ x \in D_i^c \end{cases}$$

таких, что

$$D^c = \bigcup_i D_i^c, \ \bigcap_i D_i^c = \varnothing$$

Таким образом, ветвление задачи базируется на ветвлении множества допустимых точек. Такое ветвление можно проводить рекурсивно, деля подмножества на более мелкие части. Процесс ветвления исходного множества D^c на подмножества удобно представлять в виде дерева (*дерево ветвей и границ*), узлами которого являются построенные подобласти:



Здесь верхний индекс отвечает за номер ветвления, число k_i обозначает число множеств, на которое делится допустимое множество на -м ветвлении.

Определение. Пусть при очередном ветвлении была получена задача оптимизации на множестве $x \in D'$. Число $l(D')$ $(u(D')$ в задаче на максимум) называется нижней (верхней) границей целевой функции $z(x)$ на D', если $\forall x \in D'$ $z(x) \geq l(D')$ $\left(\forall x \in D'\ u(D') \geq z(x)\right)$.

Для поиска границы на области D'можно использовать следующий способ. Решаем задачу $\begin{cases} z = (c,x) \to extr \\ x \in D' \end{cases}$ без учета целочисленности и находим оптимальное решение x^0. Если оно оказывается целочисленным, то $l(D') = z(x^0)$ $(u(D') = z(x^0)$в задаче на максимум). В противном случае $l(D') = \left[z(x^0)\right] + 1$ $(u(D') = \left[z(x^0)\right]$в задаче на максимум).

Определение. Наилучшее из известных допустимых решений называется *рекордным*, при этом значение целевой функции на нем называется *рекордом* и обозначается z^0.

Ключевой идеей метода ветвей и границ является нахождение искомого оптимума между наилучшей границей и рекордом. Так, обозначим через $l^0 = \min\left(D_i^j\right)$ самую нижнюю из нижних границ по всем вершинам дерева ветвей и границ (аналогично обозначим через $u^0 = \max\left(D_i^j\right)$ самую верхнюю из верхних границ в задаче на максимум). Тогда $l^0 \leq z_{min} \leq z^0$ $\left(z^0 \leq z_{max} \leq u^0\right)$. Задача, очевидно, решена, если граница совпала с рекордом ($l^0 = z^0$; или $z^0 = u^0$ в задаче на максимум соответственно).

Схема алгоритма решения ЗЦЛП методом ветвей и границ

Инициализация алгоритма. Пока не известно ни одного допустимого решения, поэтому рекорд равен $z^0 = +\infty$ ($z^0 = -\infty$ в задаче на максимум). Текущий список вершин дерева ветвей и границ состоит только из самого множества $D^c = D^0$.

Шаг 0. Найти нижнюю границу $l\left(D^c\right)$ множества D^c (верхнюю границу $u\left(D^c\right)$). Если при этом удалось найти допустимое решение $x' \in D^c$ исходной задачи, то:

- если $z\left(x'\right) = l\left(D^c\right)\left(z\left(x'\right) = u\left(D^c\right)\right)$, то x' – оптимальное решение исходной задачи;
- иначе обновить сведения о текущем рекорде $z^0 = z\left(x'\right)$.

Шаг $k, k > 0$.

- Если список вершин пуст, то прекратить работу, при этом если рекорд конечен, то рекордное решение является оптимальным, иначе задача недопустима.
- Выбрать для ветвления то множество из списка, которое имеет минимальную нижнюю границу (максимальную верхнюю границу). Осуществить ветвление.
- Модифицировать список вершин. Для новых подмножеств (появившихся при ветвлении) найти нижнюю (верхнюю) границу.
- Если при поиске границы стали известны допустимые решения, то корректировать сведения о рекорде. Если при этом рекорд равен минимальной из нижних границ (максимальной из верхних границ), то прекратить работу, рекордное решение является оптимальным.
- Проверить выполнение правила отсева: если для какого-то подмножества D' из списка выполнено $l\left(D'\right) > z_0$ ($u\left(D'\right) < z_0$), то это подмножество можно удалить из списка (в нем точно не содержится оптимум).

Задание 5.2

Дана задача целочисленного линейного программирования:

$$z = 2x_1 + 5x_2 \to max$$

$$\left.\begin{array}{c} 3x_1 + 3x_2 \geq 11 \\ 6x_1 - 3x_2 \leq 1 \\ 0 \leq x_1 \leq 4 \\ 0 \leq x_2 \leq 4 \\ x_1, x_2 - \text{целые} \end{array}\right\} D^c \subseteq R^2$$

Решите эту задачу, используя метод ветвей и границ.

Решение задания 5.2

Рекорд равен $z^0 = -\infty$. Текущий список вершин дерева состоит из множества $D^c = D^0$. Решим задачу без учета целочисленности (симплексным методом или графически): $x^* = \begin{pmatrix} \frac{13}{6} \\ 4 \end{pmatrix}$, $z^* = \frac{73}{3}$. Это решение не является допустимым (иначе мы бы решили исходную задачу). Коэффициенты целевой функции целые, условие целочисленности наложено на все переменные, поэтому в качестве верхней границы функции z на множестве D^c можно взять $u\left(D^c\right) = \left[\frac{73}{3}\right] = 24$. Рекорд остается равным $-\infty$.

Выполним ветвление по первой (единственной дробной) компоненте полученного на предыдущем шаге решения задачи без учета целочисленности: $D^c = D^0 = D_1^1 \cup D_2^1$, $D_1^1 \cap D_2^1 = \varnothing$, где множество D_1^1 получено добавлением к ограничениям D^0 ограничения $x_1 \le 2$, а множество D_2^1 – добавлением ограничения $x_1 \ge 3$. Множество D_2^1 пусто, соответствующая задача удаляется из рассмотрения. В результате получаем одну оценочную подзадачу:

$$z \to max$$
$$3x_1 + 3x_2 \ge 11$$
$$6x_1 - 3x_2 \le 1$$
$$0 \le x_1 \le 2$$
$$0 \le x_2 \le 4$$

Ее решение имеет вид $x^* = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$, $z^* = 24$, оно допустимое и поэтому обновляет текущий рекорд: $z^0 = z^* = 24$, $u\left(D_1^1\right) = z^* = 24$. Список допустимых вершин пуст, рекорд конечен, поэтому рекордное решение является оптимальным.

Ответ: $x^* = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$, $z^* = 24$.

Задания для самостоятельного решения

Задание 5.3. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = x_1 \to max$$
$$2x_1 \leq 3$$
$$x_1 \geq 0,\ x_1 \in \mathbb{Z}$$

Указания к решению задания 5.3.
Отсечение № 1: $\frac{1}{2}x_2 \geq \frac{1}{2}$

Задание 5.4. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = x_1 \to max$$
$$2x_1 \leq 1$$
$$3x_1 \geq 1$$
$$x_1 \geq 0,\ x_1 \in \mathbb{Z}$$

Указания к решению задания 5.4: в допустимой области нет целочисленных точек.

Задание 5.5. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = x_1 + x_2 \to max$$
$$4x_1 + 4x_2 \leq 9$$
$$x_i \geq 0,\ i = 1,2,\ x_i \in \mathbb{Z}$$

Указания к решению задания 5.5.
Отсечение № 1: $\frac{1}{4}x_3 \geq \frac{1}{4}$ или $x_1 + x_2 \leq 2$

Задание 5.6. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = x_1 + 2x_2 \to max$$
$$4x_1 + 2x_2 \leq 8$$
$$-4x_1 + 4x_2 \leq 1$$
$$x_i \geq 0,\ i = 1,2,\ x_i \in \mathbb{Z}$$

Указания к решению задания 5.6.

Отсечение № 1: $\frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2}$ или $x_2 \leq 1$.

Отсечение № 2: $\frac{1}{4}x_3 + \frac{1}{2}x_5 \geq \frac{1}{2}$ или $x_1 + x_2 \leq 2$.

Задание 5.7. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = 21x_1 + 11x_2 \to max$$
$$7x_1 + 4x_2 \leq 13$$
$$x_i \geq 0, \ \ i = 1,2, \ x_i \in \mathbb{Z}$$

Указания к решению задания 5.7.

Отсечение № 1: $\frac{4}{7}x_2 + \frac{1}{7}x_4 \geq \frac{6}{7}$ или $x_1 \leq 1$.

Отсечение № 2: $\frac{1}{4}x_3 + \frac{1}{4}x_4 \geq \frac{1}{2}$ или $2x_1 + x_2 \leq 3$.

Задание 5.8. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = 2x_1 + 3x_2 \to max$$
$$x_1 + 3x_2 \leq 9$$
$$4x_1 + x_2 \leq 13$$
$$x_i \geq 0, \ \ i = 1,2, \ x_i \in \mathbb{Z}$$

Указания к решению задания 5.8.

Отсечение № 1: $\frac{10}{11}x_3 + \frac{3}{11}x_4 \geq \frac{8}{11}$ или $2x_1 + 3x_2 \leq 11$.

Отсечение № 2: $\frac{1}{3}x_3 + \frac{1}{3}x_5 \geq \frac{2}{3}$ или $x_1 + 2x_2 \leq 6$.

Отсечение № 3: $\frac{2}{7}x_4 + \frac{6}{7}x_6 \geq \frac{6}{7}$ или $x_1 + x_2 \leq 4$.

Задание 5.9. Решить ЗЦЛП методом Гомори. Построить допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$
\begin{gathered}
z = 4x_1 + x_2 \to max \\
2x_1 + 2x_2 \ge 13 \\
x_1 - 2x_2 \ge -4 \\
10x_1 - 4x_2 \le 20 \\
x_1, x_2 \ge 0 \\
x_1, x_2 \in Z
\end{gathered}
$$

Указания к решению задания 5.9.
Оптимум без учета целочисленности имеет вид $x^* = \left(\frac{7}{2}, \frac{15}{4}\right)$, $z^* = 17\frac{3}{4}$.
Задача недопустима.

Задание 5.10. Решить ЗЦЛП методом ветвей и границ.

$$
z = 2x_1 + 5x_2 \to min
$$

$$
\left.\begin{gathered}
3x_1 + 3x_2 \ge 11 \\
6x_1 - 3x_2 \le 1 \\
0 \le x_1 \le 4 \\
0 \le x_2 \le 4 \\
x_1, x_2 - \text{целые}
\end{gathered}\right\} D^c \subseteq R^2
$$

Решение задания 5.10

Рекорд равен $z^0 = +\infty$. Текущий список вершин дерева состоит из множества $D^c = D^0$. Решим задачу без учета целочисленности (симплексным методом или графически): $x^* = \begin{pmatrix} \frac{4}{3} \\ \frac{7}{3} \end{pmatrix}$, $z^* = \frac{43}{3}$. Это решение не является допустимым (иначе мы бы решили исходную задачу). Коэффициенты целевой функции целые, условие целочисленности наложено на все переменные, поэтому в качестве нижней границы функции z на множестве D^c можно взять $l\left(D^c\right) = \left[\frac{43}{3}\right] + 1 = 15$. Рекорд остается равным $+\infty$.

Выполним ветвление по, например, первой дробной компоненте полученного на предыдущем шаге решения. Тогда $D^c = D^0 = D_1^1 \cup D_2^1$, $D_1^1 \cap D_2^1 = \varnothing$, где множество D_1^1 получено добавлением к ограничениям D^0 ограничения $x_1 \le 1$, а множество D_2^1 – добавлением ограничения $x_1 \ge 2$. В результате получаем две оценочные подзадачи:

$$z \to min$$
$$3x_1 + 3x_2 \geq 11$$
$$6x_1 - 3x_2 \leq 1$$
$$0 \leq x_1 \leq 1$$
$$0 \leq x_2 \leq 4$$

и

$$z \to min$$
$$3x_1 + 3x_2 \geq 11$$
$$6x_1 - 3x_2 \leq 1$$
$$2 \leq x_1 \leq 4$$
$$0 \leq x_2 \leq 4$$

Решение первой из них имеет вид $x^* = \begin{pmatrix} 1 \\ 8/3 \end{pmatrix}$, $z^* = \frac{46}{3}$, оно недопустимое, соответственно в качестве нижней границы функции z на множестве D_1^1 можно взять $l\left(D_1^1\right) = \left[\frac{46}{3}\right] + 1 = 16$. Ни одного допустимого решения по-прежнему не известно, поэтому текущий рекорд остается равным $z^0 = +\infty$.

Решение второй имеет вид $x^* = \begin{pmatrix} 2 \\ 11/3 \end{pmatrix}$, $z^* = \frac{67}{3}$, оно также недопустимое, $l\left(D_2^1\right) = \left[\frac{67}{3}\right] + 1 = 23$, текущий рекорд равен $z^0 = +\infty$.

Выберем для ветвления множество с минимальной нижней границей – D_1^1. Выполним ветвление по второй (единственной дробной) компоненте полученного на первом шаге решения соответствующей оценочной подзадачи. Тогда $D_1^1 = D_1^2 \cup D_2^2$, $D_1^2 \cap D_2^2 = \varnothing$, где множество D_1^2 получено добавлением к ограничениям D_1^1 ограничения $x_2 \leq 2$, а множество D_2^2 – добавлением ограничения $x_2 \geq 3$. Множество D_1^2 пусто, соответствующая задача удаляется из рассмотрения. В результате получаем одну оценочную подзадачу:

$$z \to min$$
$$3x_1 + 3x_2 \geq 11$$
$$6x_1 - 3x_2 \leq 1$$
$$0 \leq x_1 \leq 1$$
$$3 \leq x_2 \leq 4$$

Ее решение имеет вид $x^* = \begin{pmatrix} 2/3 \\ 3 \end{pmatrix}$, $z^* = \frac{49}{3}$, оно недопустимое, $l\left(D_2^2\right) = \left[\frac{49}{3}\right] + 1 = 17$, текущий рекорд остается равным $z^0 = +\infty$.

Выберем для ветвления множество с минимальной нижней границей – D_2^2. Выполним ветвление по первой (единственной дробной) компоненте решения соответствующей оценочной подзадачи. Тогда $D_2^2 = D_1^3 \cup D_2^3$, $D_1^3 \cap D_2^3 = \varnothing$, где множество D_1^3 получено добавлением к ограничениям D_2^2 ограничения $x_1 = 0$, а множество D_2^2 – добавлением ограничения $x_1 = 1$. В результате получаем две оценочные подзадачи:

$$\begin{gathered} z \to min \\ 3x_1 + 3x_2 \geq 11 \\ 6x_1 - 3x_2 \leq 1 \\ x_1 = 0 \\ 3 \leq x_2 \leq 4 \end{gathered}$$

и

$$\begin{gathered} z \to min \\ 3x_1 + 3x_2 \geq 11 \\ 6x_1 - 3x_2 \leq 1 \\ x_1 = 1 \\ 3 \leq x_2 \leq 4 \end{gathered}$$

Решение первой из них имеет вид $x^* = \begin{pmatrix} 0 \\ 11/3 \end{pmatrix}$, $z^* = \frac{55}{3}$, оно недопустимое, $l\left(D_1^3\right) = \left[\frac{55}{3}\right] + 1 = 19$. Текущий рекорд остается равным $z^0 = +\infty$.

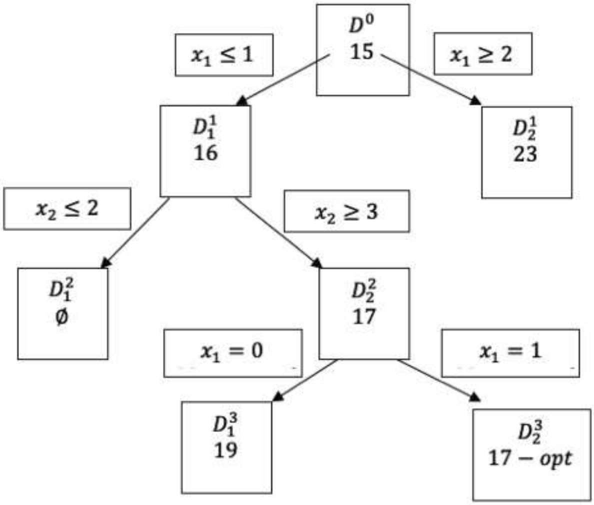
Решение второй имеет вид $x^* = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$, $z^* = 17$, оно допустимое и обновляет текущий рекорд: $z^0 = z^* = 17$, $l\left(D_2^3\right) = z^* = 17$.

Нижние границы на множествах D_1^3 и D_2^1 больше, чем текущий рекорд, поэтому они вместе с соответствующими подзадачами удаляются из рассмотрения.

Список допустимых вершин пуст, рекорд конечен, поэтому рекордное решение является оптимальным.

Ответ: $x^* = (1, 3)$, $z^* = 17$.

Дерево поиска:



Задание 5.11. Решить ЗЦЛП методом ветвей и границ.

$$z = 7x_1 + 10x_2 \to max$$

$$\left.\begin{array}{r} -x_1 + 3x_2 \leq 6 \\ 7x_1 + x_2 \leq 35 \\ 0 \leq x_1 \leq 5 \\ 0 \leq x_2 \leq 5 \\ x_1, x_2 - \text{ целые} \end{array}\right\} D^c \subseteq R^2$$

Задание 5.12. Решите ЗЦЛП методом Гомори и методом ветвей и границ. Обратите внимание на возможную неединственность. Постройте допустимую область, нецелочисленный и целочисленный оптимум и отсечения.

$$z = x_1 + 2x_2 \to max$$

$$\left.\begin{array}{r} 5x_1 + 7x_2 \leq 21 \\ -x_1 + 3x_2 \leq 8 \\ x_1 \geq 0 \\ x_1 \geq 0 \\ x_1, x_2 - \text{ целые} \end{array}\right\} D^c \subseteq R^2$$

Указания к решению задания 5.12.
Ответ: $x^* = (1, 2)$, $z^* = 5$.
Нецелочисленный оптимум имеет вид $x^* = \left(\frac{7}{22}, \frac{61}{22}\right)$, $z^* = \frac{129}{22}$

Первое отсечение по x_2 имеет вид: $\frac{1}{22}x_3 + \frac{5}{22}x_4 \geq \frac{17}{22}$

Оптимум после первого отсечения $x^* = \left(\frac{7}{5}, 2\right)$, $z^* = \frac{27}{5}$

Второе отсечение по x_1 имеет вид: $\frac{1}{5}x_3 + \frac{3}{5}x_5 \geq \frac{2}{5}$

ГЛАВА 6
Транспортная задача

Подробная постановка модели планирования перевозок (транспортной задачи) была дана в первой главе, в данной главе рассмотрим метод решения этой задачи. Транспортная задача является задачей линейного программирования, однако благодаря особому виду матрицы ограничений допускает решение существенно более простым, чем симплекс-метод, способом.

Постановку транспортной задачи удобно производить в виде таблицы:

	b_1	…	b_j	…	b_n
a_1					
…					
a_i			c_{ij} x_{ij}		
…					
a_m					

Здесь:

m – число производителей;

n – число потребителей;

a_i – количество произведенного i-м производителем товара (запас товара у i-го производителя);

b_j – количество потребленного j-м потребителем товара (потребность j-го потребителя в товаре);

c_{ij} – стоимость перевозки (выраженная в денежных единицах или в единицах времени) единицы товара от i-го производителя к j-му потребителю.

Целью решения задачи является определить оптимальный (минимальный по стоимости или времени) план перевозок, то есть количество товара x_{ij}, перевозимого от i-го производителя к j-му потребителю, где $i = 1,\ldots,m$, $j = 1,\ldots,n$. Таким образом, в математической постановке задачи участвуют $m \cdot n$ переменных x_{ij}. При этом нужно учесть ограничения, связанные с запасами производителей и с потребностями потребителей. Таким образом, в математической постановке задачи будет $m+n$ ограничений. Выпишем еще раз транспортную задачу в виде задачи линейного программирования (задача (1.9) из первой главы):

$$\begin{cases} z=\sum_i \sum_j c_{ij} x_{ij} \to min \\ \sum_j x_{ij} = a_i,\ i=1,\ldots,m \\ \sum_i x_{ij} = b_j,\ j=1,\ldots,n \\ \qquad x_{ij} \geq 0 \end{cases}$$

План и стоимость перевозок можно представлять двумя способами:

- как вектор-столбец $x=\left(x_{11},x_{12},\ldots,x_{1n},x_{21},x_{22},\ldots,x_{2n},\ldots,x_{m1},x_{m2},\ldots,x_{mn}\right)^T$ и вектор-строку $c=\left(c_{11},c_{12},\ldots,c_{1n},c_{21},c_{22},\ldots,c_{2n},\ldots,c_{m1},c_{m2},\ldots,c_{mn}\right)$ соответственно, такое представление используется в классической векторно-матричной постановке ЗЛП, которая приведена в главе 1, уравнения (1.7)–(1.8);
- как $m \times n$ матрицы $X=\left\{x_{ij}\right\}$ и $C=\left\{c_{ij}\right\}$ соответственно, такое представление удобно при постановке задачи и при содержательном анализе планов перевозок.

При использовании второго способа представления матрицы издержек ее бывает удобно преобразовать, используя следующие две теоремы.

Теорема. Точка оптимума целевой функции в транспортной задаче не меняется при прибавлении константы к строке или к столбцу матрицы $C=\left\{c_{ij}\right\}$.

Теорема. Точка оптимума целевой функции в транспортной задаче не меняется при умножении матрицы $C=\left\{c_{ij}\right\}$ на константу.

Напомним, что задача называется сбалансированной, если суммарные запасы в точности равны суммарным потребностям: $\sum_i a_i = \sum_j b_j$ (соотношение (1.10) из первой главы). Верна следующая теорема:

Теорема. Транспортная задача (1.9) разрешима тогда и только тогда, когда она сбалансирована, то есть когда $\sum_i a_i = \sum_j b_j$.

Если нарушается условие баланса, то задача становится неразрешимой, но с помощью технической «хитрости» можно перейти к сбалансированной задаче. Пусть суммарные запасы производителей больше суммарных потребностей: $\sum_i a_i > \sum_j b_j$. Тогда исходная задача (1.9) недопустима, но при этом будет иметь решение задача, в которой ограничение для производителей записано как неравенство:

$$\begin{cases} z = \sum_i \sum_j c_{ij} x_{ij} \to min \\ \sum_j x_{ij} \le a_i,\ i = 1,\ldots,m \\ \sum_i x_{ij} = b_j,\ j = 1,\ldots,n \\ x_{ij} \ge 0 \end{cases}$$

Содержательно такая модификация постановки означает, что мы допускаем наличие излишка (не отправленного ни одному из потребителей) у какого-то производителя (а может быть, и у нескольких производителей). Такую задачу можно свести к сбалансированной с помощью добавления дополнительного, фиктивного $(n+1)$-го потребителя с потребностью

$$b_{n+1} = \sum_i a_i - \sum_j b_j > 0$$

При этом положим, что стоимость перевозок $(n+1)$-му потребителю от каждого производителя равна нулю $\left(c_{i,n+1} = 0\right)$. Модифицированная задача будет иметь вид:

$$\begin{cases} z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to min \\ \sum_{j=1}^{n} x_{ij} + \boldsymbol{x_{in+1}} = a_i,\ i = 1,\ldots,m \\ \sum_{i=1}^{m} x_{ij} = b_j,\ j = 1,\ldots,n,n+1 \\ x_{ij} \ge 0 \end{cases}$$

Пусть теперь суммарные запасы производителей меньше суммарных потребностей: $\sum_i a_i < \sum_j b_j$. В этом случае исходная задача также неразрешима, и мы вынуждены допустить, что по крайней мере один из потребителей недополучит желаемое количество товара. В этом случае следует аналогичным образом перейти к допустимой задаче с помощью добавления фиктивного производителя с запасом

$$a_{m+1} = \sum_j b_j - \sum_i a_i > 0$$

Транспортная задача, как и любая задача линейного программирования, может быть решена симплекс-методом, однако благодаря специфическому строению матрицы специальных ограничений для нее существует более простой алгоритм решения. Он базируется на свойствах матрицы ограничений, сформулированных в следующих теоремах.

Теорема. Ранг матрицы специальных ограничений транспортной задачи равен $m+n-1$.

Теорема. Любые $m+n-1$ строк матрицы специальных ограничений транспортной задачи линейно независимы.

В общем случае матрица специальных ограничений транспортной задачи имеет вид:

$$\left(\begin{array}{ccccccccccccc}
1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \ddots & 1 & 1 & \cdots & 1\\
\hline
1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots & 1 & 0 & \cdots & 0\\
0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 1
\end{array}\right)
\begin{array}{l}
\text{ограничения}\\
\text{по строкам } (a_i)\\
\\
\\
\text{ограничения}\\
\text{по столбцам } (b_j)\\
\\
\\
\end{array}$$

Чтобы структура матрицы стала более понятной, выпишем ее для случая трех поставщиков $(m=3)$ и четырех потребителей $(n=4)$:

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

В качестве упражнения полезно попрактиковаться в выписывании этой матрицы для различных m и n (см. задание 6.3 в конце главы).

Перейдем к описанию метода решения транспортной задачи. Как и в случае решения симплекс-методом, ***на первом шаге*** необходимо отыскать исходное допустимое базисное решение (или, что то же самое, найти первоначальное заполнение таблицы транспортной задачи). Рассмотрим два основных способа сделать это — метод северо-западного угла и метод минимального элемента.

Рассмотрим сначала **метод северо-западного угла** первоначального заполнения таблицы транспортной задачи (поиска допустимого базисного решения). Построение исходного допустимого базисного решения начнем с левого верхнего угла таблицы (северо-западного угла). Пусть от первого производителя к первому потребителю перевозится максимально возможное число товара, то есть $x_{11}=\min(a_1,b_1)$. Тогда или запасы первого производителя исчерпаны (если $a_1 \le b_1$ и $x_{11}=a_1$), или потребность первого потребителя удовлетворена (если $a_1 \ge b_1$ и $x_{11}=b_1$), и можно исключить из рассмотрения первую строку или первый столбец соответственно, заполнив ее (его) оставшиеся элементы нулями и сократив запасы первого производителя (или потребности первого потребителя) на величину $x_{11}=\min(a_1,b_1)$. Таким образом, мы приходим к необходимости заполнить таблицу уже меньшего размера (($m-1)\times n$ или $m\times(n-1)$ или даже $(m-1)\times(n-1)$)). К ней можно снова применить ту же операцию, заполнив ее северо-западный угол. Проделав эту операцию $n+m-1$ раз (или меньше, если строка и столбец выбывали из рассмотрения одновременно), получим искомое исходное допустимое базисное решение. Проиллюстрируем данный метод на примере, заполнив таблицу для задачи с тремя потребителями и тремя поставщиками.

	$b_1=10$	$b_2=15$	$b_3=25$
$a_1=11$	$c_{11}=1$ $x_{11}=10$	$c_{12}=2$ $x_{12}=1$	$c_{13}=1$
$a_2=23$	$c_{21}=2$	$c_{22}=1$ $x_{22}=14$	$c_{32}=1$ $x_{23}=9$
$a_3=16$	$c_{31}=3$	$c_{32}=2$	$c_{33}=2$ $x_{33}=16$

Число заполненных клеток совпало с рангом матрицы специальных ограничений (в нашем случае он равен $m+n-1=3+3-1=5$). Это клетки, соответствующие базисным переменным, сами такие клетки также называются базисными. Клетки, соответствующие свободным переменным, называются свободными клетками.

Полученное исходное допустимое решение с координатами $x_{11}=10, x_{12}=1, x_{13}=0, x_{21}=0, x_{22}=14, x_{23}=9, x_{31}=0, x_{32}=0, x_{33}=16$ является базисным. Чтобы убедиться в этом, выпишем матрицу специальных ограничений и убедимся, что ее столбцы, соответствующие ненулевым переменным полученного решения, линейно независимы (в матрице ниже они выделены **жирным шрифтом**):

$$\begin{pmatrix} \mathbf{1} & \mathbf{1} & 1 & 0 & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 & 1 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & 1 & 0 & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & 0 & 1 & \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{0} & 0 & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 & 0 & \mathbf{0} & \mathbf{1} & 0 & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Теперь рассмотрим **метод минимального элемента** первоначального заполнения таблицы транспортной задачи (поиска допустимого базисного решения). Построение исходного допустимого базисного решения в этом методе начинается с любой такой клетки (i_0, j_0), для которой значение стоимости перевозки минимально (отсюда и название метода):

$$c_{i_0, j_0} = \min_{\substack{i=1,\dots,m \\ j=1,\dots,n}} c_{ij}$$

Пусть от i_0-го производителя к j_0-му потребителю перевозится максимально возможное число товара, то есть $x_{i_0, j_0} = \min\left(a_{i_0}, b_{j_0}\right)$. Тогда или запасы i_0-го производителя исчерпаны (если $a_{i_0} \le b_{j_0}$ и $x_{i_0, j_0} = a_{i_0}$), или потребность j_0-го потребителя удовлетворена (если $a_{i_0} \ge b_{j_0}$ и $x_{i_0, j_0} = b_{j_0}$), и можно исключить из рассмотрения i_0-ю строку или j_0-й столбец соответственно, заполнив ее (его) оставшиеся элементы нулями и сократив запасы i_0-го производителя (или потребности j_0-го потребителя) на величину $x_{i_0, j_0} = \min\left(a_{i_0}, b_{j_0}\right)$. Таким образом, как и в методе «северо-западного угла», мы приходим к необходимости заполнить таблицу уже меньшего размера, и к ней можно снова применить ту же операцию, найдя и заполнив клетку с минимальной стоимостью перевозок. Проделав эту операцию $m+n-1$ раз (или меньше, если строка и столбец выбывали из рассмотрения одновременно), получим искомое исходное допустимое базисное решение. Стоит отметить, что если есть несколько клеток с одинаковой минимальной стоимостью перевозок, то можно выбирать любую из них.

Проиллюстрируем данный метод на том же самом примере, заполнив таблицу для задачи с тремя потребителями и тремя поставщиками.

	$b_1 = 10$	$b_2 = 15$	$b_3 = 25$
$a_1 = 11$	$c_{11} = 1$; $x_{11} = 10$	$c_{12} = 2$	$c_{13} = 1$; $x_{13} = 1$
$a_2 = 23$	$c_{21} = 2$	$c_{22} = 1$; $x_{22} = 15$	$c_{32} = 1$; $x_{23} = 8$
$a_3 = 16$	$c_{31} = 3$	$c_{32} = 2$	$c_{33} = 2$; $x_{33} = 16$

Число заполненных (базисных) клеток снова совпало с рангом матрицы специальных ограничений, равным пяти.

Полученное исходное допустимое решение с координатами $x_{11} = 10, x_{12} = 0, x_{13} = 1, x_{21} = 0, x_{22} = 15, x_{23} = 8, x_{31} = 0, x_{32} = 0, x_{33} = 16$ является базисным. Чтобы убедиться в этом, вновь выпишем матрицу специальных ограничений и убедимся, что ее столбцы, соответствующие ненулевым переменным данного решения, линейно независимы (в матрице ниже они выделены **жирным шрифтом**):

$$\begin{pmatrix} \mathbf{1} & 1 & \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 1 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} & 1 & 1 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} & 1 & 0 & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} & 0 & \mathbf{1} & \mathbf{0} & 0 & 1 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{1} & 0 & \mathbf{0} & \mathbf{1} & 0 & 0 & \mathbf{1} \end{pmatrix}$$

Итак, используя методы северо-западного угла и минимального элемента, мы получили два разных исходных допустимых базисных решения. Чтобы понять, какое из них лучше, сравним значения целевой функции: ее значение на базисном решении, полученном методом северо-западного угла, равно 67, а на базисном решении, полученном методом минимального элемента, равно 66. Поскольку мы решаем задачу на минимум, то второй метод дает лучший результат: стоимость плана перевозок меньше, следовательно, он ближе к оптимальному. Вообще, если при составлении исходного базисного решения учитывать стоимость перевозки единицы груза, то, как правило, план будет значительно ближе к оптимальному.

В примерах выше при получении исходного базисного решения число заполненных клеток совпало с рангом матрицы специальных ограничений (равным $n+m-1$). Однако их число может оказаться меньше, чем ранг, если на каком-то шаге получения базисного решения произойдет одновременное обнуление элементов и строки, и столбца (если $\min(a_i, b_j) = a_i = b_j$). В этом случае полученное исходное допустимое базисное решение является вырожденным, и часть незаполненных нулевых клеток также надо сделать базисными. Чтобы отличить базисную клетку с нулевой перевозкой от свободной клетки, в свободных клетках принято проставлять прочерки, тогда как в базисных клетках с нулевой перевозкой проставляют ноль в явном виде. Таким образом, число заполненных клеток (клеток без прочерков) снова будет равно $m+n-1$. Иначе говоря, следует добавить недостающее число нулевых перевозок. Как правило, они добавляются в клетки с наименьшей стоимостью перевозки, эти клетки также становятся базисными.

Следующими шагами после нахождения исходного допустимого базисного решения являются его проверка на оптимальность (***шаги 2 и 3***) и, если оно не оптимально, его улучшение (***шаг 5***). Наконец, ***шаг 4*** посвящен получению всех оптимальных решений (если оптимальное решение не единственно) и выписыванию ответа.

Проверка на оптимальность и улучшение решения базируются на второй теореме двойственности, поэтому сначала необходимо выписать двойственную задачу. Двойственные переменные, соответствующие ограничениям на запас производителей, обозначим через $u_1,\ldots,u_m$. Двойственные переменные, соответствующие потребностям потребителя, обозначим через $v_1,\ldots,v_n$.

Транспортная и двойственная к ней задачи:

$$\begin{cases} z = \sum\limits_i \sum\limits_j c_{ij} x_{ij} \to min \\ \sum\limits_i x_{ij} = b_j,\ j = 1,\ldots,n \\ \sum\limits_j x_{ij} = a_i,\ i = 1,\ldots,m \\ \qquad x_{ij} \geq 0 \end{cases} \Leftrightarrow \begin{cases} w = \sum\limits_i a_i u_i + \sum\limits_j b_j v_j \to max \\ \qquad u_i + v_j \leq c_{ij} \\ j = 1,\ldots,n;\ i = 1,\ldots,m \end{cases}$$

Число переменных в двойственной задаче равно $m+n$, число специальных ограничений равно $m \cdot n$, общие ограничения отсутствуют. Двойственные переменные u_i и v_j называют потенциалами, а сам метод решения транспортной задачи – методом потенциалов.

Сформулируем вторую теорему двойственности применительно к данной паре задач.

Вторая теорема двойственности для транспортной задачи. Допустимое базисное решение транспортной задачи x^* является оптимальным тогда и только тогда, когда существует вектор $u_1^*,\ldots,u_m^*,v_1^*,\ldots,v_n^*$ такой, что

$$\begin{cases} u_i^* + v_j^* = c_{ij} \quad \forall ij : x_{ij} > 0 \\ u_i^* + v_j^* \le c_{ij} \quad \forall ij \end{cases}.$$

Эта система условий означает, что в оптимальном решении для всех ненулевых клеток выполнено равенство $u_i^* + v_j^* = c_{ij}$. Если решение невырожденное, то таких клеток $m+n-1$, таким образом, получаем систему из $m+n-1$ уравнений для $m+n$ неизвестных. Приписывая любой из переменных произвольное значение (обычно полагают $u_1 = 0$), можем определить потенциалы. Если решение вырожденное, то таких клеток $n+m-1-k$, таким образом, получаем систему из меньше чем $m+n-1$ уравнений для $m+n$ неизвестных. Приписывая любым $k+1$ переменным произвольное значение, можем определить потенциалы. Однако на практике для единообразия работы алгоритма поступают альтернативным способом: приписывают любой из переменных произвольное значение (обычно полагают $u_1 = 0$), кроме того, дополняют систему еще k уравнениями, соответствующими нулевым базисным клеткам, после чего находят потенциалы.

Суммируя вышесказанное, опишем **пошаговый алгоритм метода потенциалов решения транспортной задачи.**

Шаг 1. Находим исходное допустимое базисное решение методом «северо-западного угла» или методом минимального элемента.

Шаг 2. Для базисных клеток составляем систему линейных уравнений $u_i^* + v_j^* = c_{ij}$. Полагая, например, $u_1 = 0$, находим ее частное решение.

Шаг 3. Для свободных клеток вычисляем ***невязки*** $\delta_{ij} = u_i + v_j - c_{ij}$. Дальнейшие действия основываются на следующей идее (в свою очередь, вытекающей из доказательства второй теоремы двойственности):

Если для некоторой свободной клетки ij транспортной задачи невязка δ_{ij} положительна, то введение такой клетки в базис приведет к неувеличению целевой функции. Если же для некоторой клетки невязка положительна, то введение такой клетки в базис приведет к неуменьшению целевой функции.

Таким образом, если невязки для всех свободных клеток оказываются неположительными, то найденное решение оптимально, и мы переходим на шаг 4. В противном случае выбираем клетку с максимальной положительной невязкой, эта клетка становится ведущей, и переходим на шаг 5.

Шаг 4. Полученное базисное решение является оптимальным. Если при этом среди невязок есть нулевые, то оно является неединственным, и для отыскания всех базисных решений следует проделать шаг 5, выбрав в качестве ведущей клетки каждую клетку с нулевой невязкой. Выпуклая линейная комбинация всех найденных оптимальных базисных решений даст общее оптимальное решение транспортной задачи.

Шаг 5. В ведущую клетку вводим некоторую неотрицательную перевозку объема $q \geq 0$. Для сохранения баланса по строкам и столбцам строим ***цикл пересчета***.

Определение. Цикл пересчета – ломаная линия, вершины которой (кроме исходной) расположены в базисных (заполненных) клетках таблицы, а звенья – вдоль строк и столбцов, причем в каждой вершине встречается ровно два звена, одно из которых находится в строке, а другое – в столбце.

Дальнейшие действия основываются на следующей теореме:

Теорема. Для любой свободной клетки существует единственный начинающийся с нее цикл пересчета.

Начинаем строить цикл с ведущей клетки, вершины цикла помечаем поочередно знаками «+» и «–» (у ведущей клетки при этом знак «+»). Знак «+» означает, что объем перевозки в данной клетке будет увеличен на q единиц, знак «–» означает, что объем перевозки в данной клетке сократится на q единиц. Среди клеток, помеченным знаком «–», находим клетку с минимальным значением объема перевозки (обозначим его через c_{min}). Тогда при любом значении $q \in [0, c_{min}]$ новый план перевозок будет допустимым (баланс по строкам и столбцам будет сохранен, каждая переменная останется неотрицательной). Чтобы новый план перевозок был не только допустимым, но еще и базисным, какая-то из клеток должна выйти из базиса, для этого выберем $q = c_{min}$ и произведем пересчет перевозок. Найденная выше клетка с минимальным значением объема перевозки (или одна из таких, если их несколько) выходит из базиса, ведущая клетка входит в базис. В результате получаем новое допустимое базисное решение, переходим на шаг 2.

Ограничения на перевозки в транспортной задаче

При постановке транспортных задач на практике зачастую возникают ситуации, когда перевозки по какому-либо маршруту (от i-го производителя к j-му потребителю) ограничены дополнительными условиями. Содержательно ограничения могут быть вызваны самыми разными причинами: дополнительными контрактами на поставки от фиксированного поставщика, ограниченной грузоподъемностью части транспортных средств, ремонтом дорог и т.д., однако математически все они могут быть разделены на три типа:

1) ограничения сверху, когда от i-го производителя к j-му потребителю нельзя перевозить больше, чем k единиц товара;
2) ограничения снизу, когда i-го производителя к j-му потребителю нельзя перевозить меньше чем k единиц товара;
3) двусторонние ограничения, когда i-го производителя к j-му потребителю должно быть перевезено фиксированное количество k единиц товара.

Отметим, что ограничения различных типов могут, конечно, накладываться одновременно на разные маршруты, скажем, от первого поставщика нельзя возить более трех единиц товара второму потребителю и одновременно с этим от второго поставщика необходимо возить ровно пять единиц товара второму потребителю. Кроме того, если положить $k = 0$ в первом типе ограничений, то получим ситуацию запрета на перевозки по определенному маршруту. Научимся решать задачи с ограничениями каждого из типов.

Ограничения сверху

Пусть от i-го производителя к j-му потребителю нельзя перевозить больше чем k единиц товара. При этом мы, конечно, предполагаем, что потребности i-го потребителя $b_j > k$, иначе наличие данного ограничения не повлияет на решение задачи. Итак, пусть $b_j = k + l, l > 0$. Разделим j-й столбец таблицы на два столбца с ограничениями $b_{j1} = k$ и $b_{j2} = l$ (то есть рассмотрим вместо j-го потребителя двух потребителей с потребностями k и l). Стоимости перевозок в столбце $j1$, а также во всех строках столбца $j2$, кроме i-й строки, сделаем равными тем, которые находились в исходном столбце j. Таким образом, стоимость перевозок от каждого из производителей к потребителю $j1$ совпадает с исходной стоимостью перевозки к потребителю j, стоимость перевозок от каждого из производителей, кроме i-го, к потребителю $j2$ совпадает с исходной стоимостью перевозки к потребителю j.

Наконец, положим стоимость перевозки $c_{i,j2} = M$, где M – достаточно большое число. Таким образом, за счет высокой стоимости перевозки гипотетический потребитель $j2$ не получит товар от производителя i, а значит, исходный «объединенный» потребитель $j = j1 + j2$ получит от i-го производителя не более чем k единиц товара, что и являлось нашим ограничением. Проиллюстрируем данный метод на конкретном примере. Рассмотрим задачу с двумя производителями и двумя потребителями, заданную таблицей:

	$\boldsymbol{b_1 = 5}$	$\boldsymbol{b_2 = 12}$
$\boldsymbol{a_1 = 8}$	$c_{11} = 1$	$c_{12} = 2$
$\boldsymbol{a_2 = 9}$	$c_{21} = 4$	$c_{22} = 3$

Пусть в задаче существует ограничение первого типа (ограничение сверху), состоящее в том, что от первого поставщика второму потребителю можно перевезти не более 5 единиц товара. Разделим второй столбец на два столбца с ограничениями $b_{21} = 5$ и $b_{22} = 7$. Стоимости перевозок в столбце 21, а также во всех строках столбца 22, кроме первой строки, сделаем равными тем, которые находились в исходном втором столбце. Положим стоимость перевозки $c_{1,22} = M$, где M – достаточно большое число:

	$\boldsymbol{b_1 = 5}$	$\boldsymbol{b_{21} = 5}$	$\boldsymbol{b_{22} = 12 - 5 = 7}$
$\boldsymbol{a_1 = 8}$	$c_{11} = 1$	$c_{1,21} = 2$	$\boldsymbol{c_{1,22} = M}$
$\boldsymbol{a_2 = 9}$	$c_{21} = 4$	$c_{2,21} = 3$	$c_{2,22} = 3$

Далее полученную задачу можно решить стандартным методом потенциалов.

Прежде чем переходить к ограничению второго типа, рассмотрим частный случай ограничения сверху при $k = 0$, соответствующий запрету на перевозки. Пусть в предыдущей задаче с двумя производителями и двумя потребителями запрещено перевозить товар от первого поставщика второму потребителю. Тогда делить второй столбец не нужно, достаточно положить стоимость перевозки $c_{12} = M$, где M – достаточно большое число:

	$b_1 = 5$	$b_2 = 12$
$a_1 = 8$	$c_{11} = 1$	$c_{12} = M$
$a_2 = 9$	$c_{21} = 4$	$c_{22} = 3$

Отметим, что в обоих примерах, рассмотренных выше, после формализации задачи с ограничениями она решается стандартным методом потенциалов. Для клетки с «запретительной» стоимостью перевозки M невязка всегда будет отрицательна, поэтому такая клетка не войдет в базис и перевозка по ней будет нулевой. В случае $k > 0$ после решения задачи с разделенным j-м столбцом необходимо вернуться к исходной постановке, полагая перевозки в j-м столбце равными сумме перевозок в $j1$ и $j2$ столбцах.

Ограничения снизу

Пусть от i-го производителя к j-му потребителю нельзя перевозить меньше чем k единиц товара. При этом мы предполагаем, что запасы i-го производителя $a_i \geq k$ и потребности j-го потребителя $b_j \geq k$, иначе наличие данного ограничения приведет к неразрешимости задачи. Итак, пусть $a_i = k + l_a, b_j = k + l_b, l_a, l_b \geq 0$. Зафиксируем перевозку объема k единиц от i-го производителя к j-му потребителю, тогда запасы i-го производителя уменьшатся до l_a единиц, потребности j-го потребителя уменьшатся до l_b единиц. Решим полученную транспортную задачу с обновленными ограничениями, после чего добавим к ее оптимальному плану перевозок перевозку k единиц товара от i-го производителя к j-му потребителю. Проиллюстрируем данный метод на предыдущем примере – снова рассмотрим задачу с двумя производителями и двумя потребителями, заданную таблицей:

	$b_1 = 5$	$b_2 = 12$
$a_1 = 8$	$c_{11} = 1$	$c_{12} = 2$
$a_2 = 9$	$c_{21} = 4$	$c_{22} = 3$

Пусть в задаче существует ограничение второго типа (ограничение снизу), состоящее в том, что от первого поставщика второму потребителю можно перевезти не менее 5 единиц товара. Зафиксируем эту перевозку объема 5 единиц и получим транспортную задачу с обновленными ограничениями:

	$b_1 = 5$	$b_2 - 5 = 7$
$a_1 - 5 = 3$	$c_{11} = 1$	$c_{12} = 2$
$a_2 = 9$	$c_{21} = 4$	$c_{22} = 3$

Далее полученную задачу можно решить стандартным методом потенциалов. Для получения ответа к исходной задаче к полученному оптимальному плану надо добавить 5 единиц товара от первого поставщика второму потребителю.

Двусторонние ограничения

Пусть от i-го производителя к j-му потребителю должно быть перевезено фиксированное количество k единиц товара. При этом мы, как и в предыдущем случае, предполагаем, что запасы i-го производителя $a_i \geq k$ и потребности j-го потребителя $b_j \geq k$, иначе наличие данного ограничения приведет к неразрешимости задачи. Итак, пусть снова $a_i = k + l_a$, $b_j = k + l_b, l_a, l_b \geq 0$. Нам нужно снова зафиксировать перевозку объема k единиц от i-го производителя к j-му потребителю, тогда запасы i-го производителя уменьшатся до l_a единиц, потребности j-го потребителя уменьшатся до l_b единиц. Но теперь, в отличие от предыдущего случая, нужно запретить дальнейшие перевозки по этому маршруту, то есть положим стоимость перевозки $c_{i,j} = M$, где M – достаточно большое число. Решая полученную транспортную задачу с обновленными ограничениями, получим план перевозок, в котором за счет высокой стоимости M потребитель j не получит товар от производителя i, после чего добавим к этому плану перевозку k единиц товара от i-го производителя к j-му потребителю. Проиллюстрируем данный метод на предыдущем примере – снова рассмотрим задачу с двумя производителями и двумя потребителями, заданную таблицей:

	$b_1 = 5$	$b_2 = 12$
$a_1 = 8$	$c_{11} = 1$	$c_{12} = 2$
$a_2 = 9$	$c_{21} = 4$	$c_{22} = 3$

Пусть в задаче существует ограничение третьего типа (двустороннее ограничение), состоящее в том, что от первого поставщика второму потребителю необходимо перевезти ровно 5 единиц товара. Зафиксируем эту перевозку объема 5 единиц, положим $c_{12} = M$, где M – достаточно большое число, и получим транспортную задачу с обновленными ограничениями:

	$b_1 = 5$	$b_2 - 5 = 7$
$a_1 - 5 = 3$	$c_{11} = 1$	$c_{12} = M$
$a_2 = 9$	$c_{21} = 4$	$c_{22} = 3$

Далее полученную задачу можно решить стандартным методом потенциалов. Для получения ответа к исходной задаче к полученному оптимальному плану надо добавить 5 единиц товара от первого поставщика второму потребителю.

Задача о назначениях

Как мы помним, транспортная задача – важный частный случай задачи линейного программирования. В свою очередь, важным (распространенным на практике) частным случаем транспортной задачи является задача о назначениях. Опишем ее содержательную постановку. Пусть имеется n работ и n возможных исполнителей этих работ. Любой исполнитель должен быть назначен на выполнение ровно одной любой работы, при этом характеристики выполнения работ исполнителями различаются. Под характеристиками понимают или затраты (временные или денежные) на исполнение (и тогда задача о назначениях ставится на минимум), или эффективность выполнения (и тогда задача о назначениях ставится на максимум). Итак, задача о назначениях полностью задается матрицей затрат (или эффективностей) $C=\left(c_{ij}\right)_{i,j=1,\dots,n}$, где c_{ij} – затраты (эффективность) i-го исполнителя при выполнении i-й работы. Нужно распределить работы так, чтобы выполнить работы с минимальными затратами (с максимальной эффективностью).

Задачу о назначениях можно рассматривать как транспортную задачу, в которой исполнители являются «поставщиками услуг», а работы – «потребителями услуг», тогда (при задаче на минимум) затраты c_{ij} имеют смысл «стоимости доставки услуг».

Такая постановка задачи (когда число исполнителей совпадает с числом работ) называется *линейной задачей о назначениях*. Мы далее рассмотрим именно такую классическую постановку, хотя, вообще говоря, задачу можно ставить более широко, рассматривая неравное число исполнителей и задач. В этом случае при превышении числа исполнителей над числом задач для решения прибегают к уже знакомому нам приему – вводят нужное число фиктивных работ с нулевыми затратами (эффективностью) выполнения. Итоговое распределение какому-то из исполнителей такой фиктивной задачи означает, что он не будет работать (не получит задачи для выполнения). Аналогично при превышении числа задач над числом исполнителей вводят нужное число фиктивных исполнителей, и назначение какой-то задаче такого исполнителя означает, что она не будет выполняться. Существуют и более сложные подходы к решению, когда работа может быть назначена нескольким исполнителям (исполнитель может быть назначен на выполнение нескольких работ).

Перейдем к математической постановке линейной задачи о назначениях.

Введем переменные $x_{ij}=\begin{cases}1, \text{ если } i\text{-й работник назначен на } j\text{-ю работу} \\ 0, \text{ иначе}\end{cases}$.

Условие выполнения i-м кандидатом ровно одной работы имеет вид: $\sum_{j=1}^{n} x_{ij}=1$. Условие назначения на выполнение j-й работы ровно одного кандидата имеет вид: $\sum_{i=1}^{n} x_{ij}=1$. Итак, математическая постановка задачи имеет вид:

$$z=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}\rightarrow\min(\max)$$

$$\sum_{j=1}^{n} x_{ij}=1, i=1,\dots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1,\ldots,n$$

$$x_{ij} \geq 0, i, j = 1,\ldots,n$$

В такой постановке решение может быть не целым, но при этом всегда существует целочисленное оптимальное решение (это следует из свойств матрицы специальных ограничений). Так как задача о назначениях является частным случаем транспортной задачи, то для ее решения можно использовать метод потенциалов. Кроме того, разработаны специальные методы решения, учитывающие специфику задачи. Основным таким методом является венгерский метод. Подробнее про него можно прочитать в книге Х. Таха «Введение в исследование операций (в 2 книгах)» (Таха, 1985).

Примеры решения заданий

Задание 6.1

(а) Решите транспортную задачу, заполнив исходную таблицу методом северо-западного угла. С обоснованием укажите количество базисных оптимальных решений.

	8	10	11
4	1	2	1
3	4	3	8
13	1	7	6
9	3	6	4

(б) Выпишите линейную комбинацию столбцов матрицы специальных ограничений, соответствующих клеткам первого цикла пересчета в вашем решении.

(в) Пусть от третьего поставщика первому потребителю запрещены перевозки. Формализуйте в виде таблицы задачу, учитывающую данное условие, и решите ее.

Решение задания 6.1

(а) Первоначальное заполнение, потенциалы (при $\varepsilon_1 = 0$) и первый цикл:

	8	10	11	
4	1 4	2 –	1 –	$u_1 = 0$
3	4 $3-\varepsilon_1$	3 $+\varepsilon_1$	8 –	$u_2 = 3$
13	1 $1+\varepsilon_1$	7 $10-\varepsilon_1$	6 2	$u_3 = 0$
9	3 –	6 –	4 9	$u_4 = -2$
	$v_1 = 1$	$v_2 = 7$	$v_3 = 6$	

Клетка 22 выбрана ведущей, так как для нее $\delta_{22} = u_2 + v_2 - c_2 = 7 > 0$, наибольшая невязка из существующих.

Для пересчета берем $\varepsilon_1 = 3$. Получаем новое допустимое базисное решение (при $\varepsilon_2 = 0$) и строим второй цикл:

	8	10	11
4	1 $4-\varepsilon_2$	2 –	1 $+\varepsilon_2$
3	4 –	3 3	8 –
13	1 $4+\varepsilon_2$	7 7	6 $2-\varepsilon_2$
9	3 –	6 –	4 9

Для пересчета берем $\varepsilon_2 = 2$. Получаем новое допустимое базисное решение (при $\varepsilon_3 = 0$) и строим третий цикл:

	8	10	11
4	1 $2-\varepsilon_3$	2 $+\varepsilon_3$	1 2
3	4 –	3 3	8 –
13	1 $6+\varepsilon_3$	7 $7-\varepsilon_3$	6 –
9	3 –	6 –	4 9

Для пересчета берем $\varepsilon_3 = 2$. Получаем новое допустимое базисное решение (при $t = 0$), оно является оптимальным. Строим цикл с параметром, чтобы выписать неединственное решение:

	8	10	11
4	1 –	2 $2+t$	1 $2-t$
3	4 –	3 3	8 –
13	1 8	7 $5-t$	6 $+t$
9	3 –	6 –	4 9

Таблица выше – ответ, изменения параметра $t \in [0, 2]$, $z_{min} = 94$. Существует два базисных оптимальных решения, которые можно получить при подстановке крайних точек отрезка изменений параметра.

(б) В цикле участвуют переменные x_{21}, x_{22}, x_{31}, x_{32}. Соответствующая циклу линейная комбинация столбцов матрицы специальных ограничений:

$$-1\cdot\begin{pmatrix}0\\1\\0\\0\\1\\0\\0\end{pmatrix}+1\cdot\begin{pmatrix}0\\1\\0\\0\\0\\1\\0\end{pmatrix}+1\cdot\begin{pmatrix}0\\0\\1\\0\\1\\0\\0\end{pmatrix}-1\cdot\begin{pmatrix}0\\0\\1\\0\\0\\1\\0\end{pmatrix}=\overline{0}$$

(в) Формализация запрета на перевозки от третьего поставщика первому потребителю заключается в установлении запретительной стоимости перевозки. Положим $c_{31}=M$, где M – конечное, но очень большое число, такое, что $\forall c\ M-c>0,\ c-M<0$.

Первоначальное заполнение (при $\varepsilon_1=0$) и первый цикл:

	8	10	11
	1	2	1
4	4	–	–
	4	3	8
3	3	–	–
	M	7	6
13	$1-\varepsilon_1$	10	$2+\varepsilon_1$
	3	6	4
9	$+\varepsilon_1$	–	$9-\varepsilon_1$

Для пересчета берем $\varepsilon_1=1$. Получаем новое допустимое базисное решение (при $\varepsilon_2=0$) и строим второй цикл:

	8	10	11
4	4 (1)	– (2)	– (1)
3	$3-\varepsilon_2$ (4)	$+\varepsilon_2$ (3)	– (8)
13	– (M)	$10-\varepsilon_2$ (7)	$3+\varepsilon_2$ (6)
9	$1+\varepsilon_2$ (3)	– (6)	$8-\varepsilon_2$ (4)

Для пересчета берем $\varepsilon_2 = 3$. Получаем новое допустимое базисное решение (при $\varepsilon_3 = 0$) и строим третий цикл:

	8	10	11
4	$4-\varepsilon_3$ (1)	$+\varepsilon_3$ (2)	– (1)
3	– (4)	3 (3)	– (8)
13	– (M)	$7-\varepsilon_3$ (7)	$6+\varepsilon_3$ (6)
9	$4+\varepsilon_3$ (3)	– (6)	$5-\varepsilon_3$ (4)

Для пересчета берем $\varepsilon_3 = 4$. Получаем новое допустимое базисное решение (при $t = 0$), оно является оптимальным. Строим цикл с параметром, чтобы выписать неединственное решение:

	8	10	11
4	1 –	2 $4-t$	1 $+t$
3	4 –	3 3	8 –
13	M –	7 $3+t$	6 $10-t$
9	3 8	6 –	4 1

Таблица выше – ответ, изменения параметра $t \in [0,4]$, $z_{min} = 126$.

Задание 6.2

Дана матрица эффективностей выполнения трех работ Ирой, Олей и Асей:

$$\begin{array}{r c} & \begin{array}{ccc} \text{Ира} & \text{Оля} & \text{Ася} \end{array} \\ \begin{array}{r} \text{работа } 1 \\ \text{работа } 2 \\ \text{работа } 3 \end{array} & \begin{pmatrix} 5 & 3 & 8 \\ 2 & 2 & 4 \\ 6 & 1 & 9 \end{pmatrix} \end{array}$$

(а) Поставьте данную задачу о назначениях как ЗЛП.

(б) Выпишите двойственную задачу.

(в) Пусть Оля выполняет работу 1, Ира – работу 2, Ася – работу 3. Проверьте данное решение на оптимальность, используя вторую теорему двойственности.

(г) Решите исходную задачу о назначениях.

Решение задания 6.2

(а) Выпишем ЗЛП на максимум, поскольку c_{ij} – это эффективность:

$$z = 5x_{11} + 3x_{12} + 8x_{13} + 2x_{21} + 2x_{22} + 4x_{23} + 6x_{31} + x_{32} + 9x_{33} \to max$$

$$\sum_i x_{ij} = 1, \ j = 1,2,3$$

$$\sum_j x_{ij} = 1, \ i = 1,2,3$$

$$x_{ij} \geq 0$$

Можно считать, что задача целочисленная (тогда к условию на неотрицательность добавляется еще условие целочисленности) или даже бинарная (тогда вместо условия на неотрицательность нужно писать, что $x_{ij} \in \{0;1\}$). Все три варианта допустимы.

(б) Выпишем двойственную задачу:

$$w = u_1 + u_2 + u_3 + v_1 + v_2 + v_3 \to min$$

$$u_1 + v_1 \geq 5$$

$$u_1 + v_2 \geq 3$$

$$u_1 + v_3 \geq 8$$

$$u_2 + v_1 \geq 2$$

$$u_2 + v_2 \geq 2$$

$$u_2 + v_3 \geq 4$$

$$u_3 + v_1 \geq 6$$

$$u_3 + v_2 \geq 1$$

$$u_3 + v_3 \geq 9$$

u_i, v_j – произвольного знака.

(в) Так как Оля выполняет работу 1, Ира – работу 2, Ася – работу 3, то по второй теореме двойственности

$$u_1 + v_2 = 3$$

$$u_2 + v_1 = 2$$

$$u_3 + v_3 = 9$$

Выражаем из равенств u_i и подставляем в оставшиеся неравенства. Получаем из $u_1 + v_1 \geq 5$, что $v_1 - v_2 \geq 2$, а из $u_2 + v_2 \geq 2$, что $v_1 - v_2 \leq 0$. Это противоречие, поэтому исходное распределение на работы было неоптимальным.

г) Приведем матрицу эффективностей к матрице издержек, чтобы решать задачу на минимум:

$$\begin{pmatrix} 5 & 3 & 8 \\ 2 & 2 & 4 \\ 6 & 1 & 9 \end{pmatrix} \underset{\cdot(-1)}{\sim} \begin{pmatrix} -5 & -3 & -8 \\ -2 & -2 & -4 \\ -6 & -1 & -9 \end{pmatrix} \underset{(+10)}{\sim} \begin{pmatrix} 5 & 7 & 2 \\ 8 & 8 & 6 \\ 4 & 9 & 1 \end{pmatrix}$$

Первоначальное заполнение проведем методом северо-западного угла, поставим «базисные» нули в клетки с минимальной стоимостью так, чтобы не возникало циклов из базисных переменных.

	1	1	1
1	$\boxed{5}$ 1	$\boxed{7}$ –	$\boxed{2}$ 0
1	$\boxed{8}$ –	$\boxed{8}$ 1	$\boxed{6}$ 0
1	$\boxed{4}$ –	$\boxed{9}$ –	$\boxed{1}$ 1

Строим первый цикл:

	1	1	1
1	$\boxed{5}$ $1-\varepsilon_1$	$\boxed{7}$ –	$\boxed{2}$ $0+\varepsilon_1$
1	$\boxed{8}$ $+\varepsilon_1$	$\boxed{8}$ 1	$\boxed{6}$ $0-\varepsilon_1$
1	$\boxed{4}$ –	$\boxed{9}$ –	$\boxed{1}$ 1

Для пересчета берем $\varepsilon_1 = 0$. Получаем новое допустимое базисное решение (при $t = 0$), оно является оптимальным. Строим цикл с параметром, чтобы выписать неединственное решение:

	1	1	1
1	$\boxed{5}$ $1-t$	$\boxed{7}$ –	$\boxed{2}$ $0+t$
1	$\boxed{8}$ 0	$\boxed{8}$ 1	$\boxed{6}$ –
1	$\boxed{4}$ $+t$	$\boxed{9}$ –	$\boxed{1}$ $1-t$

Если мы считаем задачу целочисленной, то $t = 0\ 8;81$, следовательно, существует два оптимальных распределения на работы:

Ира – первая работа, Оля – вторая работа, Ася – третья работа;

Ира – третья работа, Оля – вторая работа, Ася – первая работа.

$$z_{max} = 16$$

Можно также выписывать ответ для непрерывной задачи, тогда $t \in [0,1]$ и $x^* = (1-t, 0,\ t, 0, 1, 0,\ t, 0, 1-t)$.

Задание 6.3

(а) Решите транспортную задачу, заполнив исходную таблицу методом минимального элемента.

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$
$a_1 = 17$	4	5	8
$a_2 = 15$	6	4	7
$a_3 = 19$	2	3	1
$a_4 = 16$	9	9	6

(б) Стало известно, что из a_1 в b_2 можно поставлять не более 5 единиц. Формализуйте и решите задачу, учитывающую данное условие.

(в) Дополнительно (по отношению к пункту б) стало известно, что запрещены перевозки из a_4 в b_3. Формализуйте и решите задачу, учитывающую одновременно оба условия.

(г) Стало известно (по отношению к пункту а), что из a_1 в b_2 нужно поставлять не менее 6 единиц. Формализуйте и решите задачу, учитывающую данное условие.

Решение задания 6.3

(а) Задача является сбалансированной.

Требуется заполнить $m + n - 1 = 6$ клеток, так как в задаче 6 базисных переменных. Заполнение получается вырожденным, следовательно, нужно поставить «базисный 0»; по договоренностям ставим его в клетку с минимальной стоимостью, но так, чтобы не возникал цикл (то есть линейная зависимость) между базисными переменными:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$
$a_1 = 17$	4 17	5 –	8 –
$a_2 = 15$	6 –	4 15	7 –
$a_3 = 19$	2 –	3 0	1 19
$a_4 = 16$	9 8	9 –	6 8

По базисным клеткам находим частное решение двойственной задачи:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$	
$a_1 = 17$	4 17	5 –	8 –	$u_1 = 0$
$a_2 = 15$	6 –	4 15	7 –	$u_2 = 1$
$a_3 = 19$	2 –	3 0	1 19	$u_3 = 0$
$a_4 = 16$	9 8	9 –	6 8	$u_4 = 5$
	$v_1 = 4$	$v_2 = 3$	$v_3 = 1$	

Для свободных клеток считаем $u_i + v_j - c_{ij}$, которое должно быть ≤ 0 в оптимальной точке:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$	
$a_1 = 17$	4 17	5 –	8 –	$u_1 = 0$
$a_2 = 15$	6 –	4 15	7 –	$u_2 = 1$
$a_3 = 19$	2 –	3 0	1 19	$u_3 = 0$
$a_4 = 16$	9 8	9 –	6 8	$u_4 = 5$
	$v_1 = 4$	$v_2 = 3$	$v_3 = 1$	

Базисное решение не является оптимальным, следовательно, строим цикл пересчета по клетке «31»:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$	
$a_1 = 17$	4 17	5 –	8 –	$u_1 = 0$
$a_2 = 15$	6 –	4 15	7 –	$u_2 = 1$
$a_3 = 19$	2 $+\varepsilon$	3 0	1 $19-\varepsilon$	$u_3 = 0$
$a_4 = 16$	9 $8-\varepsilon$	9 –	6 $8+\varepsilon$	$u_4 = 5$
	$v_1 = 4$	$v_2 = 3$	$v_3 = 1$	

В новом базисном решении $\varepsilon = 8$.

Решение после пересчета. Сразу проверяем его на оптимальность:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$	
$a_1 = 17$	4 17	5 0	8 –	$u_1 = 0$
$a_2 = 15$	6 –	4 15	7 –	$u_2 = -1$
$a_3 = 19$	2 8	3 0	1 11	$u_3 = -2$
$a_4 = 16$	9 –	9 –	6 16	$u_4 = 3$
	$v_1 = 4$	$v_2 = 3$	$v_3 = 1$	

Решение является оптимальным, так как все потенциалы $u_i + v_j - c_{ij} \le 0$.

Присутствует неединственность, так как один из потенциалов равен 0, но она является ложной, так как соответствующее циклу пересчета по этой переменной $\varepsilon = 0$.

(б) Из a_1 в b_2 можно поставлять не более 5 единиц – формализуем это ограничение на перевозки. Для этого разбиваем первую строку на две (можно аналогично разбить второй столбец):

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$
$a_1' = 5$	4	5	8
$a_1'' = 12$	4	M	8
$a_2 = 15$	6	4	7
$a_3 = 19$	2	3	1
$a_4 = 16$	9	9	6

где M – достаточно большое число.

(в) Дополнительно запрещены перевозки из a_4 в b_3 – формализуем это условие:

	$b_1 = 25$	$b_2 = 15$	$b_3 = 27$
$a_1' = 5$	4	5	8
$a_1'' = 12$	4	M	8
$a_2 = 15$	6	4	7
$a_3 = 19$	2	3	1
$a_4 = 16$	9	9	M

где M – достаточно большое число.

(г) Чтобы учесть данное ограничение на перевозки, вычтем из a_1 и b_2 6 единиц.

	$b_1 = 25$	$b_2 - 6 = 9$	$b_3 = 27$
$a_1 - 6 = 11$	4	5	8
$a_2 = 15$	6	4	7
$a_3 = 19$	2	3	1
$a_4 = 16$	9	9	6

Задания для самостоятельного решения

Задание 6.4

(а) Решите транспортную задачу, заполнив исходную таблицу методом минимального элемента.

	9	13	11
11	4	5	8
4	6	4	7
8	2	3	1
10	9	9	6

(б) Выпишите матрицу специальных ограничений данной транспортной задачи.

Задание 6.5. Решите транспортную задачу, начав с вырожденного базисного решения.

	9	13	12
11	4	7	9
4	1	5	6
8	2	7	8
11	9	5	7

Задание 6.6. Решите транспортную задачу, заполнив исходную таблицу методом северо-западного угла.

	7	6	3	4
4	4	6	5	3
7	1	4	8	9
9	9	3	2	6

Задание 6.7. Найдите все оптимальные решения транспортной задачи, взяв за исходное указанное базисное решение.

	7	5	2	1
3	1 –	3 3	8 –	2 –
5	4 4	3 –	7 –	3 1
7	4 3	2 2	3 2	5 –

Задание 6.8. Найдите все оптимальные решения транспортной задачи, взяв за исходное указанное базисное решение. Укажите, с обоснованием, количество базисных оптимальных решений. Укажите, с обоснованием, все базисные оптимальные решения.

	17	11	5	10
9	5 9	2 –	3 –	5 –
18	6 3	7 –	4 5	2 10
16	9 5	4 11	6 –	3 –

Задание 6.9.

а) Решите задачу о назначении четырех исполнителей на четыре вида работ. Степень эффективности выполнения i-м исполнителем j-го вида работ задана матрицей.

$$\begin{pmatrix} 6 & 8 & 3 & 1 \\ 5 & 6 & 1 & 2 \\ 7 & 9 & 1 & 8 \\ 5 & 2 & 3 & 4 \end{pmatrix}$$

Начальное заполнение таблицы произведите одним из двух стандартных способов.

б) Решите задачу о назначениях, считая, что матрица задает плату за выполнение i-м исполнителем j-го вида работ. Начальное заполнение таблицы произведите методом северо-западного угла.

Задание 6.10. Дана транспортная задача:

	9	13
13	4 –	7 13
2	1 2	5 –
7	9 7	5 –

(а) Представьте данную транспортную задачу в виде задачи линейного программирования.

(б) Запишите задачу, двойственную к данной.

(в) Проверьте предлагаемое решение исходной задачи на оптимальность, используя вторую теорему двойственности.

Задание 6.11. Решите несбалансированную транспортную задачу, заполнив таблицу методом северо-западного угла.

	10	3	17
3	1	2	3
5	2	2	1
17	1	3	2

ГЛАВА 7
Решение практических задач на языке R

Один из важных навыков для экономиста – умение найти оптимальное решение для слабо формализованной проблемы, заданной лишь общим словесным описанием.

Первым шагом[1] является формализация текстового описания проблемы. Этот шаг включает подготовку альтернатив решения, определение целевой функции и построение системы ограничений, в условиях которых мы будем искать экстремум целевой функции.

Второй шаг – это перевод формализованной задачи на четкий язык математических соотношений: введение переменных, запись ограничений в виде равенств и неравенств.

На *третьем шаге* происходит решение полученной математической задачи. Если она является задачей линейного программирования (или может быть сведена к ней с помощью адекватных упрощающих предположений), то ее можно решать методами, изложенными в предыдущих разделах. При этом в реальных задачах число переменных и ограничений, как правило, слишком велико для решения вручную, и необходимо использовать какой-либо программный пакет. Можно автоматизировать решение ЗЛП в Microsoft Excel, существуют подходящие библиотеки на широко распространенных среди экономистов языках программирования Python и R. Менее известные, но релевантные программы — это, например, TOra. Важными критериями выбора программного пакета являются его бесплатное свободное распространение, а также постоянное обновление до самых свежих алгоритмов. В данном разделе будет продемонстрирована техника решения задач в среде программирования R Studio, подходящей под данные критерии, где реализованы и автоматизированы изученные в предыдущих главах алгоритмы[2].

---

[1] По мотивам: Таха Х. (1985). Введение в исследование операций (в 2 книгах). М.: Мир, с. 21.

[2] При работе на языке Python можно воспользоваться библиотекой scipy, в которой для решения задач оптимизации есть раздел optimize, содержащий функцию linprog для решения задач линейного программирования различными методами.

Четвертый шаг предполагает проверку адекватности полученного решения, его соответствие интуитивным результатам, реальным данным. Современный уровень развития технологий позволяет использовать для этого исследование устойчивости модели к варьированию ее экзогенных параметров, анализ чувствительности.

Наконец, *на пятом шаге* на основе полученного решения предлагаются конкретные практические рекомендации, которые лицо, принимающее решения, может применять.

Для комфортного чтения дальнейшего текста читатель должен быть знаком с основами языка программирования R. Ссылки на соответствующие пособия можно найти, например, на официальном сайте R[1].

Базовый пакет, который мы будем использовать, – это пакет lpSolve. Он подходит для решения задач линейного программирования общего вида, а также задач с ограничением на целочисленность части (или всех) переменных. Дополнительно в этом пакете есть функции, которые позволяют компактно решать транспортную задачу и задачу о назначениях. Кроме того, после подключения пакета доступны функции, позволяющие искать двойственные переменные, а также определять границы применения третьей теоремы двойственности (что полезно для проведения анализа чувствительности).

Для использования этого (и любого другого) пакета его нужно предварительно загрузить (это необходимо сделать один раз для каждого устройства), а затем распаковать (это необходимо делать в начале каждой сессии работы в R):

```
install.packages("lpSolve") # загрузка пакета lpSolve
library(lpSolve) # распаковка пакета lpSolve
```

Общий вид команды для решения произвольной задачи линейного программирования имеет вид:

```
lp (direction = «min», objective.in, const.mat, const.dir, const.rhs,
int.vec, compute.sens=0, binary.vec, all.int=FALSE, all.bin=FALSE, ...)
```

Расшифруем аргументы этой команды:

- direction = «min»: направление оптимизации целевой функции, здесь можно указать минимум или максимум, по умолчанию задача решается на минимум;
- objective.in: вектор коэффициентов целевой функции;
- const.mat: матрица коэффициентов специальных ограничений задачи линейного программирования; отметим, что все переменные

[1] https://www.r-project.org/doc/bib/R-books.html

задачи предполагаются неотрицательными (в главе 1 обсуждалось, как преобразовать задачу к такому виду);

- const.dir: вектор знаков ограничений, каждый знак может принимать значения «<=», «=» или «>=»; отметим, что в ЗЛП все ограничения типа «неравенство» должны быть нестрогими, и поэтому, даже если указать знак «<», команда воспримет соответствующее ограничение как «<=»;
- const.rhs: вектор «правой» части специальных ограничений задачи;
- compute.sens=0: бинарный параметр, по умолчанию равный нулю; при его равенстве единице дополнительно проводится анализ чувствительности (считаются границы экзогенных параметров модели, при которых возможно применение третьей теоремы двойственности);
- int.vec: вектор, задающий индексы переменных, которые должны быть целочисленными; например, int.vec = c(1,0,1,0) обозначает, что переменные x_1 и x_3 должны быть целыми, а на переменные x_2 и x_4 ограничение целочисленности не наложено;
- binary.vec: вектор, задающий индексы переменных, которые должны быть равны либо 0, либо 1; например, binary.vec = c(1,0,1,0) обозначает, что переменные x_1 и x_3 должны быть бинарными, то есть принимать всего два значения: 0 или 1;
- all.int = FALSE: бинарный (логический) параметр, по умолчанию равный нулю (FALSE); при его равенстве единице (TRUE) все переменные считаются целыми;
- all.bin= FALSE: бинарный (логический) параметр, по умолчанию равный нулю (FALSE); при его равенстве единице (TRUE) все переменные считаются бинарными.

Общий вид команды для решения произвольной транспортной задачи имеет вид:

lp.transport (cost.mat, direction=«min», row.signs, row.rhs, col.signs, col.rhs, compute.sens=0, integers = 1:(nc*nr))

Расшифруем аргументы этой команды:

- cost.mat: матрица издержек перевозки, элемент с индексом ij соответствует издержкам перевозки одной единицы продукции от i-го производителя j-му потребителю;
- direction=«min»: направление оптимизации целевой функции, здесь можно указать минимум или максимум, по умолчанию задача решается на минимум;

- row.signs: вектор знаков ограничений по строкам, каждый знак может принимать значения «<=», «=» или «>=»; напомним, что ограничения по строкам соответствуют ограничениям производителей;
- row.rhs: вектор производственных мощностей (запасов) производителей;
- col.signs: вектор знаков ограничений по столбцам, каждый знак может принимать значения «<=», «=» или «>=»; напомним, что ограничения по столбцам соответствуют ограничениям потребителей;
- col.rhs: вектор потребностей потребителей;
- compute.sens=0: бинарный параметр, по умолчанию равный нулю; при его равенстве единице дополнительно проводится анализ чувствительности (считаются границы экзогенных параметров модели, при которых возможно применение третьей теоремы двойственности);
- integers = 1:(nc*nr): вектор, i-й элемент которого задает индекс i-й целочисленной переменной; таким образом, длина вектора совпадает с числом целочисленных переменных задачи; по умолчанию вектор имеет длину mn, то есть все переменные являются целочисленными; если условие целочисленности необходимо наложить, например, только на переменные x_1, x_2 и x_5, то необходимо задать integers = c(1,2,5); если условие целочисленности не накладывается ни на одну переменную, то необходимо задать integers = NULL.

В пакете lpSolve доступна также команда lp.assign, предназначенная для решения задачи о назначениях, она представляет собой упрощенный вариант разобранной команды lp.transport. Любую задачу о назначениях можно решить как с ее помощью, так и с помощью команды lp.transport, и мы не будем останавливаться на ней подробно. Результатом выполнения обеих разобранных команд (и lp, и lp.transport) является список, содержащий следующие элементы:

- direction: направление оптимизации целевой функции (заданное при вызове функции);
- x.count: число переменных в целевой функции;
- objective: вектор коэффициентов целевой функции (заданный при вызове функции);
- const.count: число ограничений (заданное при вызове функции);
- constraints: матрица ограничений (заданная при вызове функции, только для команды lp);
- int.count: число целочисленных переменных;
- int.vec: вектор индексов целочисленных переменных (заданный при вызове функции);

- objval: оптимальное значение целевой функции;
- solution: вектор оптимальных значений переменных;
- num.bin.solns: число возвращенных решений;
- status: число, указывающее на результат решения задачи: 0 (задача разрешима), 2 (задача недопустима) или 3 (задача не ограничена).

Подробнее про обе команды и про возвращаемый ими результат можно прочитать в руководстве[1]. Примеры задач линейного программирования, решенных в среде R, можно найти в (Sallan, Lordan, Fernandez, 2015).

Описанные пакеты являются базовыми для решения задач линейного программирования в среде R. Существует множество других пакетов, решающих более частные постановки, а также решающих задачу линейного программирования и транспортную задачу с помощью других алгоритмов. Подробнее прочитать про другие алгоритмы решения задачи линейного программирования и их сложность можно в (Дасгупта, Пападимитриу, Вазирани, 2014).

Примеры решения заданий

В Приложении В для удобства читателя находится листинг решения всех задач из данного раздела.

Задание 7.1

Формализуйте описанную ниже ситуацию[2] в виде задачи линейного программирования.

Фармацевтическая фирма Roza ежедневно должна производить не менее 800 кг некой пищевой добавки, которая состоит из смеси двух компонентов: A и B, состав которых представлен в следующей таблице:

Компонент	Белок	Клетчатка	Стоимость (в руб. за кг)
	(в кг на кг компонента)		
A	0,09	0,02	0,30
B	0,60	0,06	0,90

1 The Comprehensive R Archive Network, https://cran.r-project.org/web/packages/lpSolve/lpSolve.pdf (дата обращения: 9 мая 2025 г.).

2 По мотивам примера из: Таха Х. (1985). Введение в исследование операций (в 2 книгах). М.: Мир.

Диетологи требуют, чтобы в пищевой добавке было не менее 30% белка и не более 5% клетчатки. Фирма Roza хочет определить рецептуру смеси наименьшей стоимости с учетом требований диетологов.

Решение задания 7.1

Пусть x_1 – это количество (в кг) компонента A, используемого в производстве добавки, x_2 – это количество (в кг) компонента B, используемого в производстве добавки.

Целевая функция представляет собой общую стоимость пищевой добавки, что можно записать следующим образом:

$$z = 0{,}3x_1 + 0{,}9x_2 \to min$$

Фирма должна выпускать не менее 800 кг пищевой добавки в день:

$$x_1 + x_2 \geq 800$$

Рассмотрим ограничение, связанное с количеством белка в пищевой добавке. Общее количество белка в смеси равно $0{,}09x_1 + 0{,}6x_2$, эта величина должна составлять не менее 30% от общего веса смеси, равного $x_1 + x_2$. Получаем ограничение:

$$0{,}09x_1 + 0{,}6x_2 \geq 0{,}3(x_1 + x_2)$$

$$\text{или } -0{,}21x_1 + 0{,}3x_2 \geq 0$$

Аналогично строится ограничение, связанное с количеством клетчатки:

$$0{,}02x_1 + 0{,}06x_2 \leq 0{,}05(x_1 + x_2)$$

$$\text{или } -0{,}03x_1 + 0{,}01x_2 \leq 0$$

Объединяя целевую функцию и ограничения, получаем следующую задачу линейного программирования:

$$z = 0{,}3x_1 + 0{,}9x_2 \to min$$

$$x_1 + x_2 \geq 800$$

$$-0{,}21x_1 + 0{,}3x_2 \geq 0$$

$$-0{,}03x_1 + 0{,}01x_2 \leq 0$$

$$x_1 \geq 0, x_2 \geq 0$$

Запишем алгоритм ее решения на языке программирования R. Необходимые вектора и матрицы можно задать следующим образом:

```
C <- c(0.3,0.9) # вектор коэффициентов целевой функции
A <- matrix(c(1,   1,
              -0.21, 0.3,
              -0.03, 0.01), byrow = TRUE, ncol = 2, nrow = 3) # матрица специальных ограничений
b <- c(800, 0, 0) # вектор «правой» части специальных ограничений
constraints_direction <- c(«>=», «>=», «<=») # вектор знаков ограничений специального вида
```

Для решения задачи применим разобранную выше функцию «lp» из пакета lpSolve, результат сохраним в объект «mod»:

```
mod <- lp(direction = «min», objective.in = C,
          const.mat = A, const.dir = constraints_direction, const.rhs = b,
          compute.sens = FALSE)
```

Проверим, является ли задача разрешимой:

```
mod$status
```

Значение (при запуске этой команды оно выводится на экран) равно 0, это значит, что задача разрешима.

Посмотрим на оптимальное значение переменных и наилучшее значение целевой функции:

```
mod$solution
mod$objval
```

При запуске данных команд на экран выводятся оптимальное решение и оптимальное значение целевой функции. Можно не выводить значение на экран, а сохранить в памяти. Например, чтобы создать переменную z_min и присвоить ей оптимальное значение целевой функции, необходимо использовать команду

```
z_min<-mod$objval
```

Для рассматриваемой задачи получаем, что $x_1^{opt} = 470{,}6$ кг, $x_2^{opt} = 329{,}4$ кг, $z_{min} = 437{,}65$ руб.

Задание 7.2

Компания производит две модели стульев: 4Р и 3Р. Модель 4Р состоит из 4 ножек, 1 сиденья и 1 спинки, модель 3Р состоит из 3 ножек и 1 сиденья. У компании есть первоначальный запас из 200 ножек, 500 сидений и 100 спинок. Если компании нужно больше комплектующих, то она может приобрести деревянный блок по цене 80 рублей/штука. Из одного блока можно сделать 10 сидений, 20 ножек и 2 спинки (одновременно, выбирать между комплектующими не нужно). Стоимость производства одного стула модели 4Р составляет 30 рублей за стул, а для модели 3Р – 40 рублей за стул. Наконец, за рассматриваемый период компания должна произвести минимум 1000 стульев любой модели. Формализуйте данную ситуацию в виде задачи линейного программирования, где целевая функция представляет собой минимум производственных издержек, и ответьте на вопросы.

Вопросы:

1. Сколько стульев модели 4Р произведет компания?
2. Сколько деревянных блоков закупит компания?
3. Каковы оптимальные издержки производства?
4. Как изменятся минимальные издержки, если компания должна будет произвести 1100 стульев вместо 1000?
5. Предположим, что минимальное число стульев для производства меняется от 500 до 1500. Как в данном случае будут меняться оптимальные издержки компании? Проиллюстрируйте ответ с помощью графика.

Решение задания 7.2

Отметим, что в данном примере нужно не только сформулировать и решить задачу линейного программирования, но и провести анализ чувствительности решения к изменению ограничения с помощью теории двойственности.

Пусть x_4 – количество стульев модели 4Р, которые произведет компания, x_3 – количество стульев модели 3Р, которые произведет компания, x_{wood} – количество деревянных блоков, которые компания закупит для дополнительного производства комплектующих.

Ножки для производства стульев можно брать из запаса в 200 штук, а также из деревянных блоков (из одного блока получается 20 ножек). Тогда ограничение по используемым ножкам имеет вид:

$$4x_4 + 3x_3 \leq 200 + 20x_{wood}$$

Аналогично ограничение по сиденьям имеет вид:

$$x_4 + x_3 \leq 500 + 10x_{wood}$$

Ограничение по спинкам имеет вид (учтем, что спинки используются только для модели 4P):

$$x_4 \leq 100 + 2x_{wood}$$

Ограничение по требуемому объему производства имеет вид:

$$x_4 + x_3 \geq 1000$$

Целевая функция представляет собой издержки на производство моделей 4P и 3P, включая закупку деревянных блоков, и исследуется на минимум:

$$z = 30x_4 + 40x_3 + 80x_{wood} \to min$$

Объединяя целевую функцию и ограничения, получаем следующую задачу линейного программирования (с учетом неотрицательности всех переменных):

$$z = 30x_4 + 40x_3 + 80x_{wood} \to min$$

$$4x_4 + 3x_3 - 20x_{wood} \leq 200$$

$$x_4 + x_3 - 10x_{wood} \leq 500$$

$$x_4 - 2x_{wood} \leq 100$$

$$x_4 + x_3 \geq 1000$$

$$x_4 \geq 0, x_3 \geq 0, x_{wood} \geq 0$$

Запишем алгоритм ее решения на языке программирования R. Необходимые вектора и матрицы можно задать следующим образом:

```
C <- c(30, 40, 80) # вектор коэффициентов целевой функции
A <- matrix(c(4, 3,-20,
              1, 1,-10,
              1, 0, -2,
              1, 1,0), byrow = TRUE, ncol = 3, nrow = 4) # матрица специаль-
ных ограничений
b <- c(200, 500, 100, 1000) # вектор «правой» части специальных ограни-
чений
```

```
constraints_direction <- c(«<=», «<=», «<=», «>=») # вектор знаков ограничений специального вида
```

Для решения задачи применим разобранную выше функцию «lp» из пакета lpSolve, результат сохраним в объект «mod»:

```
mod <- lp(direction = «min», objective.in = C,
          const.mat = A, const.dir = constraints_direction, const.rhs = b,
          compute.sens = TRUE)
```

С помощью команды mod$status убедимся в том, что задача разрешима. Посмотрим на оптимальное значение переменных и наилучшее значение целевой функции:

```
mod$solution
mod$objval
```

В результате выполнения этих двух команд найдем ответы на 1, 2 и 3-й вопросы:

$$x_4^{opt} = 422{,}2 \text{ (стула)}$$

$$x_{wood}^{opt} = 161{,}1 \text{ (стула)}$$

$$z_{min} = 48666{,}7 \text{ (рубля)}$$

Отметим, что при необходимости данную задачу можно было решать в целых числах. Для этого аргумент all.int нужно положить равным TRUE:

```
mod_new <- lp(direction = «min», objective.in = C,
              const.mat = A, const.dir = constraints_direction, const.rhs = b,
              all.int = TRUE)
```

Решение целочисленного варианта задачи:

$$x_4^{opt} = 420 \text{ (стульев)}$$

$$x_{wood}^{opt} = 161 \text{ (стул)}$$

$$z_{min} = 48680 \text{ (рублей)}$$

В рамках первоначальной постановки (без требования целочисленности переменных) проведем анализ чувствительности, чтобы дать ответ на 4 и 5-й вопросы.

С помощью команды mod$duals можно посмотреть оптимальные значения двойственных переменных. С помощью команд

```
mod$duals.from
mod$duals.to
```

можно найти интервалы применимости третьей теоремы двойственности для любого ограничения, то есть, в частности, выяснить, при каких изменениях правой части ограничения решение двойственной задачи не меняется.

В рассматриваемой задаче начальное значение правой части ограничения по объему производства равно 1000 стульев, третья теорема двойственности применима при минимальном объеме производства, принадлежащем множеству $[133{,}3;+\infty)$, соответствующая двойственная переменная принимает значение 50. Изменение требования по объему до 1100 стульев лежит в этих границах, значит, для нахождения изменения оптимального значения минимальных издержек можем воспользоваться третьей теоремой двойственности:

$$\Delta z_{min} = y^{opt} \cdot \Delta b = 50 \cdot (1100 - 1000) = 5000$$

Это число является ответом на вопрос 4 – минимальные издержки вырастут на 5000 рублей, если компания должна будет произвести 1100 стульев вместо 1000.

Для ответа на вопрос 5 запустим цикл, в котором правая часть ограничения по объему меняется от 500 до 1500 с шагом 1. На каждом шаге решим задачу и сохраним оптимальное значение целевой функции, в результате получим вектор z, содержащий оптимальные значения целевой функции для каждого значения правой части ограничения:

```
b4 <- 500:1500
z <- NULL # зададим пустой вектор, куда будем сохранять оптимальные
значения целевой функции
for(i in 1:length(b4)) {
      b <- c(200, 500, 100, b4[i])
      mod <- lp(direction = "min", objective.in = C,
                  const.mat = A, const.dir = constraints_direction, const.rhs = b,
                  compute.sens = TRUE)
      z[i] <- mod$objval
      }
```

Построим график изменения оптимальных издержек при изменении требований о минимальном объеме производства с 500 до 1500 стульев с помощью команды

```
plot(b4, z, type = "l")
```

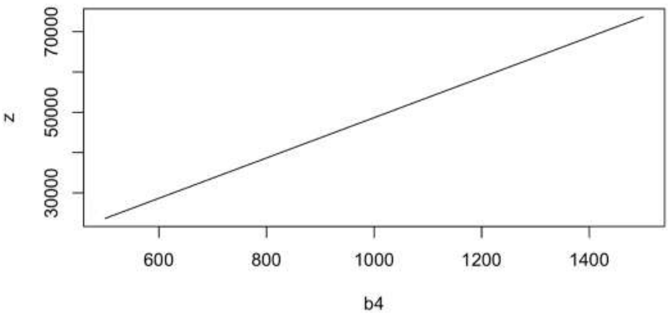


Рис. 7.1.

Итак, оптимальные издержки компании меняются линейно (что следует из применимости третьей теоремы двойственности при таких значениях правой части, $[500,1500] \subset [133,3;+\infty)$), возрастая с 23 666,7 рубля до 73 666,7 рубля. Требование производства каждого дополнительного стула приводит к росту издержек на 50 рублей.

Задание 7.3

Компания, занимающаяся производством сладких напитков, имеет четыре завода $C1 - C4$. Производительность заводов составляет 170, 150, 190 и 200 тыс. литров ежемесячно. Сладкие напитки направляются на три принадлежащих этой компании распределительных центра $S1 - S3$, мощности которых составляют 250, 150 и 270 тыс. литров в месяц. Транспортные затраты (в тыс. руб.) на перевозку 1 тыс. литров напитков с заводов в центры указаны в следующей таблице:

Завод \ Центр	$S1$	$S2$	$S3$
$C1$	7	3	8
$C2$	5	4	6
$C3$	4	5	9
$C4$	6	2	5

Определите план перевозок сладких напитков в распределительные центры, который обеспечивает минимальные совокупные транспортные издержки, и ответьте на вопросы.

Вопросы:

1. Сколько напитков следует перевозить с завода $C1$ в центр $S2$?
2. Сколько напитков следует перевозить с завода $C4$ в центр $S3$?
3. Какова общая минимальная стоимость перевозок?
4. Стало известно, что поставки с завода $C1$ в центр $S2$ нужно ограничить объемом 50 тыс. литров. К тому же из-за плохого состояния дороги перевозки с завода $C4$ в центр $S3$ невозможны.
   (а) Запишите таблицу транспортной задачи, учитывающей эти условия.
   (б) Найдите новый оптимальный план перевозок, учитывающий эти условия.
   (в) Насколько возрастет стоимость перевозок?
5. Сколько напитков следуют перевозить с завода $C4$ в центр $S2$ с учетом дополнительной информации?

Решение задания 7.3

Напомним общий вид команды для решения транспортной задачи (функция «lp.transport» из пакета lpSolve):

```
lp.transport (cost.mat, direction=”min”, row.signs, row.rhs, col.signs,
              col.rhs, compute.sens=0, integers = 1:(nc*nr) )
```

Обратим внимание, что сумма производственных возможностей превышает суммарные потребности распределительных центров на 40 тыс. литров, то есть задача не сбалансирована и, вообще говоря, неразрешима:

```
sum(c(170, 150, 190, 200)) == sum(c(250, 150, 270)) # в результате получим
FALSE
sum(c(170, 150, 190, 200)) - sum(c(250, 150, 270)) # в результате получим 40
```

Чтобы решить задачу без требования вывезти всю произведенную продукцию (что невозможно), введем фиктивного потребителя (распределительный центр) с потребностью (мощностью) в размере 40 тыс. литров в месяц, издержки перевозки в который от каждого из заводов нулевые (подробнее о решении несбалансированных транспортных задач см. главу 6). Зададим необходимые аргументы.

Сначала опишем производительность заводов, то есть ограничения по строкам:

```
row.signs <- rep(«=»,4)  # знаки ограничений по строкам
row.rhs <- c(170, 150, 190, 200) # правые части ограничений по строкам
```

Аналогично сформулируем потребности распределительных центров, то есть ограничения по столбцам (учтем, что их вместе с фиктивным потребителем 4):

```
col.signs <- rep(«=», 4) # знаки ограничений по столбцам
col.rhs <- c(250, 150, 270, 40) # правые части ограничений по столбцам
```

Зададим матрицу издержек перевозок:

```
cost.mat <- matrix(c(7, 3, 8, 0,
                     5, 4, 6, 0,
                     4, 5, 9, 0,
                     6, 2, 5, 0), nrow = 4, byrow = TRUE)
```

Применим функцию «lp.transport», результат сохраним в объект «mod»:

```
mod <- lp.transport(cost.mat = cost.mat,
                    direction = “min”,
                    row.signs = row.signs,
                    row.rhs = row.rhs,
                    col.signs = col.signs,
                    col.rhs = col.rhs)
```

Чтобы ответить на вопросы 1 и 2, воспользуемся командой

```
mod$solution
```

Результатом выполнения команды является таблица перевозок:

```
     [,1] [,2] [,3] [,4]
[1,]    0  130    0   40
[2,]   60    0   90    0
[3,]  190    0    0    0
[4,]    0   20  180    0
```

Интересующие нас числа находятся на пересечении первой строки и второго столбца (вопрос 1) и на пересечении четвертой строки и третьего столбца (вопрос 2). Таким образом, с завода *C1* в центр *S2* нужно перевозить 130 тыс. литров напитков ежемесячно, а с завода *C4* в центр *S3* нужно перевозить 180 тыс. литров напитков ежемесячно.

Оптимальное значение целевой функции (ответ на вопрос 3) можно узнать, выполнив команду

```
mod$objval
```

Оно равно 2930 тыс. рублей.

Для ответа на вопрос 4 формализуем условие, добавив ограничения на перевозки (подробнее о том, как это сделать, см. главу 6). Для учета первого ограничения разделим производственные возможности завода *C*1 (170 единиц) на 50 единиц, которые можно перевозить по первоначальным ценам, и оставшиеся 120 единиц, которые нельзя перевозить во второй центр (для этого в соответствующую клетку ставим запретительную цену **B**). Для учета второго ограничения (запрет на перевозку с завода 4 в центр 3) выставим запретительную цену перевозки **C** в соответствующую клетку таблицы:

Центр / **Завод**	***S*1**	***S*2**	***S*3**
*C*1 = 50	7	3	8
*C*1 = 170–50 = 120	7	B	8
*C*2	5	4	6
*C*3	4	5	9
*C*4	6	2	C

Зададим новые ограничения по строкам и столбцам:

```
row.signs1 <- rep("=",5)
row.rhs1 <- c(50, 120, 150, 190, 200)
col.signs1 <- rep("=", 4)
col.rhs1 <- c(250, 150, 270, 40)
B <- sum(cost.mat)*sum(row.rhs)
C <- sum(cost.mat)*sum(row.rhs)
cost.mat1 <- matrix(c(7, 3, 8, 0,
```

```
              7, B, 8, 0,
              5, 4, 6, 0,
              4, 5, 9, 0,
              6, 2, C, 0), nrow = 5, byrow = TRUE)
```

Ищем решение новой задачи:

```
mod1 <- lp.transport(cost.mat = cost.mat1,
                     direction = "min",
                     row.signs = row.signs1,
                     row.rhs = row.rhs1,
                     col.signs = col.signs1,
                     col.rhs = col.rhs1)
```

Чтобы найти новый оптимальный план перевозок, воспользуемся командой

```
mod$solution
```

Результатом выполнения команды является таблица перевозок:

```
     [,1] [,2] [,3] [,4]
[1,]    0    0   50    0
[2,]    0    0   80   40
[3,]   10    0  140    0
[4,]  190    0    0    0
[5,]   50  150    0    0
```

Ответ на вопрос 5 находится на пересечении последней строки (напомним, что она относится к заводу $C4$) и второго столбца (центр $S2$): с завода $C4$ в центр $S2$ с учетом дополнительной информации следует перевозить 150 тыс. литров напитков.

Отметим, что при этом минимальные издержки на перевозки возрастут на 360 тыс. рублей ($z_{new} - z_{old} = 3290 - 2930 = 360$) – такова «цена» введенных ограничений.

Задачи для самостоятельного решения

Задание 7.4[1]

Петр подобрал себе пять вариантов витаминных комплексов. Каждый комплекс содержит витамины и вещества, наиболее важные для Петра, готовящегося к соревнованиям на выносливость. Необходимо определить, какие комплексы и в каком количестве следует принимать Петру. В следующей таблице указано количество витаминов и веществ (в мг), которое должен получить Петр за курс приема витаминов, а также данные о содержании витаминов и веществ в витаминных комплексах (в мг на 1 г) и цены за 1 г комплекса (в руб.):

Комплекс / Витамин	1	2	3	4	5	Необходимо
A	1,1	1,2	1,8	1,1	1,3	250
B	0,9	1,1	0,7	1	1,1	128
C	50	60	40	30	60	7000
Железо	24	45	18	12	37	3700
Кальций	210	340	150	260	300	32 000
Цена	3,4	4,3	2,4	2,2	3,7	

Определите, какие витаминные комплексы следует принимать, чтобы с минимальными затратами пройти курс подготовки, и ответьте на вопросы.

Вопросы:

1. Какое количество комплекса 4 следует принять?
2. Какое общее количество витаминных комплексов следует принять?
3. Какова минимальная стоимость курса подготовки?
4. До какого значения должна снизиться цена на комплекс 2, чтобы его следовало включить в курс подготовки?

[1] По мотивам: Афанасьев М. Ю., Суворов Б. П. (2003). Исследование операций в экономике: модели, задачи, решения. ИНФРА-М.

Задание 7.5[1]

Посадская фирма по производству платков планирует освоение новых рынков сбыта в пяти городах. Возможности сбыта невелики, так что в каждый город достаточно направить одного торгового представителя фирмы для заключения с магазинами договоров о поставках.

В следующей таблице указан объем спроса (в млн руб.):

Город	Москва	Санкт-Петербург	Новгород	Самара	Ростов
Объем спроса	9	5	4	3	6

Фирма располагает данными о профессиональных возможностях шести своих сотрудников. В следующей таблице содержатся оценки степени освоения рынка, которую может обеспечить соответствующий торговый представитель фирмы:

Представитель	П1	П2	П3	П4	П5	П6
Оценка степени освоения рынка	70%	60%	50%	80%	40%	50%

Так, представитель П1 может освоить 70% от объема спроса в любом городе. Например, если направить его в Москву, то доход фирмы на этом рынке составит 6,3 млн рублей.

Распределите торговых агентов по городам таким образом, чтобы фирма получила максимальный доход, и ответьте на вопросы.

Вопросы:

1. Чему равен максимальный доход фирмы?
2. В какой город следует направить торгового представителя П1?
3. Кто из торговых представителей не будет использован?

[1] По мотивам: Афанасьев М. Ю., Суворов Б. П. (2003). Исследование операций в экономике: модели, задачи, решения. ИНФРА-М.

СПИСОК ЛИТЕРАТУРЫ

Основной список литературы

1. Акулич И. Л. (2011). Математическое программирование в примерах и задачах. СПб: Лань. 352 с.
2. Количественные методы в экономических исследованиях (2013). Под ред. М. В. Грачевой, Л. Н. Фадеевой, Ю. Н. Черемных. М.: ЮНИТИ-ДАНА. 678 с.
3. Таха Х. (1985). Введение в исследование операций (в 2 книгах). М.: Мир. 479 и 496 с.
4. Юдин Д. Б., Гольштейн Е. Г. (1963). Линейное программирование. Теория и конечные методы. М.: ФизМатЛит. 776 с.
5. Sallan J. M., Lordan O., Fernandez V. (2015). Modeling and Solving Linear Programming with R. OmniaScience; 1st edition. 108 с.

Дополнительный список литературы

1. Афанасьев М. Ю., Суворов Б. П. (2003). Исследование операций в экономике: модели, задачи, решения. ИНФРА-М. 444 с.
2. Ашманов С. А., Тимохов А. В. (2012). Теория оптимизации в задачах и упражнениях. СПб: Лань. 448 с.
3. Богданова Е. Л. (2017). Оптимизация в проектном менеджменте: линейное программирование: учебное пособие / Е. Л. Богданова, К. А. Соловейчик, К. Г. Аркина. СПб.: Университет ИТМО. 165 с.
4. Васильев Ф. П., Иваницкий А. Ю. (2008). Линейное программирование. М.: Факториал Пресс. 347 с.
5. Вентцель Е. С. (2010). Исследование операций. Задачи, принципы, методология. М.: КноРус. 208 с.
6. Дасгупта С., Пападимитриу Х., Вазирани У. (2014) Алгоритмы. Пер. с англ. под ред. А. Шеня. М.: МЦНМО. 320 с.
7. Писарук Н. Н. (2015) Исследование операций. Минск: БГУ. 304 с.

П Р И Л О Ж Е Н И Е А

Образец программы курса «Исследование операций»

Тема 1. Задача линейного программирования

Постановка задачи линейного программирования. Основные понятия, примеры задач линейного программирования: задача планирования производства, задача о диете, транспортная задача. Геометрическая интерпретация и геометрическое решение задачи линейного программирования в случае двух переменных.

Тема 2. Геометрия линейного программирования

Понятие отрезка в n-мерном пространстве. Понятие выпуклого множества. Выпуклость множества допустимых решений и множества оптимальных решений задачи линейного программирования. Теоремы о соответствии крайних точек и допустимых базисных решений, о существовании допустимого и оптимального базисного решения задачи линейного программирования. Многогранное множество, многогранник. Теорема о представлении многогранника. Представление допустимого и оптимального множеств задачи линейного программирования.

Тема 3. Симплексный метод

Алгебра симплексного метода. Симплексная таблица и работа с ней. Признак оптимальности допустимого базисного решения. Признак неограниченности целевой функции. Признак неединственности оптимального решения. Нахождение всех оптимальных решений и всех базисных оптимальных решений. Понятие вырожденного базисного решения. Проблема зацикливания. Дополнительные переменные и их использование в симплексном методе. Метод искусственного базиса. Двойственный симплекс-метод.

Тема 4. Теория двойственности и анализ чувствительности

Двойственность в линейном программировании. Построение сопряженной задачи для исходной задачи в стандартной, канонической и общей формах. Первая теорема двойственности. Вторая теорема двойственности. Условия дополняющей нежесткости. Теорема о маргинальных значениях. Экономическая и геометрическая интерпретация двойственных переменных. Анализ устойчивости. Вывод функций спроса на ресурсы и предложения товаров. Связь между вырожденностью и неединственностью решения. Двойственный симплексный метод.

Тема 5. Целочисленное программирование

Целочисленные задачи линейного программирования. Сведение ЗЦЛП к ЗЛП. Метод отсечений. Отсечение Данцига. Отсечение Гомори, правильность отсечения Гомори, лемма Гомори. Комбинаторные методы дискретного программирования. Метод ветвей и границ. Некоторые экономические задачи целочисленного программирования.

Тема 6. Транспортная задача

Различные формы транспортной задачи. Сбалансированность и допустимость транспортной задачи. Ранг матрицы ограничений транспортной задачи. Нахождение исходного допустимого базисного решения методом северо-западного угла и методом минимального элемента. Понятие цикла. Метод потенциалов решения транспортной задачи. Вырожденность и неединственность в транспортной задаче. Транспортная задача с ограничениями на пропускную способность. Задача о назначениях. Составление расписания.

Тема 7. Решение практических задач на языке R

Формализация реальных практических ситуаций в виде задач линейного программирования.

Геометрическое решение задачи линейного программирования в случае двух переменных. Решение практических задач симплексным методом. Решение практических задач на анализ чувствительности. Решение практических задач методами целочисленного программирования. Решение практических транспортных задач и задач о назначении.

Тема 8. Элементы теории графов и оптимизация на сетях

Основные понятия теории графов. Задача о кратчайшем пути. Алгоритм Дейкстры и его сложность. Сведение задачи ЦЛП к задаче о кратчайшем пути. Потоки в сетях. Задача о максимальном потоке. Понятие увеличивающей цепи. Понятие минимального разреза. Теорема Форда–Фалкерсона. Анализ социальных взаимодействий на основе сетей. Применение графов в управлении временем проекта, метод критического пути.

Тема 9. Элементы динамического программирования

Постановка задачи динамического программирования. Принцип оптимальности Беллмана для решения задач динамического программирования с конечным и бесконечным горизонтом. Существование и единственность решения уравнения Беллмана. Алгоритмы решения оптимизационных задач, основанные на принципе Беллмана. Методы поиска функции ценности (value function). Задача о распределении ресурсов. Задача о рюкзаке.

П Р И Л О Ж Е Н И Е Б

Листинг решения задач, разобранных в главе 7

Листинг задания 7.1:

```
install.packages("lpSolve") # загрузка пакета lpSolve
library(lpSolve) # распаковка пакета lpSolve
C <- c(0.3,0.9) # вектор коэффициентов целевой функции
A <- matrix(c(1,   1,
              -0.21, 0.3,
              -0.03, 0.01), byrow = TRUE, ncol = 2, nrow = 3) # матрица спе-
циальных ограничений
b <- c(800, 0, 0) # вектор «правой» части специальных ограничений
constraints_direction <- c(«>=», «>=», «<=») # вектор знаков ограничений
специального вида
mod <- lp(direction = «min», objective.in = C,
          const.mat = A, const.dir = constraints_direction, const.rhs = b,
          compute.sens = FALSE)
mod$status
mod$solution
mod$objval
z_min<-mod$objval
```

Листинг задания 7.2:

```
install.packages("lpSolve") # загрузка пакета lpSolve
library(lpSolve) # распаковка пакета lpSolve
C <- c(30, 40, 80) # вектор коэффициентов целевой функции
A <- matrix(c(4, 3,-20,
              1, 1,-10,
              1, 0, -2,
              1, 1,0), byrow = TRUE, ncol = 3, nrow = 4) # матрица специаль-
ных ограничений
b <- c(200, 500, 100, 1000) # вектор «правой» части специальных ограни-
чений
```

```
constraints_direction <- c(«<=», «<=», «<=», «>=») # вектор знаков ограни-
чений специального вида
mod <- lp(direction = «min», objective.in = C,
          const.mat = A, const.dir = constraints_direction, const.rhs = b,
          compute.sens = TRUE)
mod$solution
mod$objval
mod_new <- lp(direction = «min», objective.in = C,
              const.mat = A, const.dir = constraints_direction, const.rhs = b,
              all.int = TRUE)
mod$duals.from
mod$duals.to
b4 <- 500:1500
z <- NULL # зададим пустой вектор, куда будем сохранять оптимальные
значения целевой функции
for(i in 1:length(b4)) {
    b <- c(200, 500, 100, b4[i])
    mod <- lp(direction = "min", objective.in = C,
              const.mat = A, const.dir = constraints_direction, const.rhs = b,
              compute.sens = TRUE)
     z[i] <- mod$objval
     }
plot(b4, z, type = "l")
```

Листинг задания 7.3:

```
install.packages("lpSolve") # загрузка пакета lpSolve
library(lpSolve) # распаковка пакета lpSolve
sum(c(170, 150, 190, 200)) == sum(c(250, 150, 270)) # в результате получим
FALSE
sum(c(170, 150, 190, 200)) - sum(c(250, 150, 270)) # в результате получим 40
row.signs <- rep(«=»,4)  # знаки ограничений по строкам
row.rhs <- c(170, 150, 190, 200) # правые части ограничений по строкам
col.signs <- rep(«=», 4) # знаки ограничений по столбцам
col.rhs <- c(250, 150, 270, 40) # правые части ограничений по столбцам
cost.mat <- matrix(c(7, 3, 8, 0,
                     5, 4, 6, 0,
                     4, 5, 9, 0,
                     6, 2, 5, 0), nrow = 4, byrow = TRUE)
```

```
mod <- lp.transport(cost.mat = cost.mat,
                    direction = "min",
                    row.signs = row.signs,
                    row.rhs = row.rhs,
                    col.signs = col.signs,
                    col.rhs = col.rhs)
mod$solution
mod$objval
  row.signs1 <- rep("=",5)
  row.rhs1 <- c(50, 120, 150, 190, 200)
  col.signs1 <- rep("=", 4)
  col.rhs1 <- c(250, 150, 270, 40)
  B <- sum(cost.mat)*sum(row.rhs)
  C <- sum(cost.mat)*sum(row.rhs)
  cost.mat1 <- matrix(c(7, 3, 8, 0,
                        7, B, 8, 0,
                        5, 4, 6, 0,
                        4, 5, 9, 0,
                        6, 2, C, 0), nrow = 5, byrow = TRUE)
mod1 <- lp.transport(cost.mat = cost.mat1,
                     direction = "min",
                     row.signs = row.signs1,
                     row.rhs = row.rhs1,
                     col.signs = col.signs1,
                     col.rhs = col.rhs1)
mod$solution
```

Скачать коды ко всем задачам одновременно можно, перейдя по ссылке:

https://disk.yandex.ru/d/NTUmcF141vV2Fw

Электронное издание сетевого распространения.
9,5 печ. л. Опубликовано 01.08.2025.
Издательство «ЭФ МГУ имени М.В. Ломоносова»;
www.econ.msu.ru; +7 (495) 939-17-15

Клачкова О. А., Кострикин И. А., Рощина Я. А.

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Учебное пособие

ISBN 978-5-907690-75-2

9 785907 690752