Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)

Кафедра «Прикладная математика»

В.М. Гончаренко, А.К. Керимов, В.Ю. Попов

ЛИНЕЙНЫЕ МОДЕЛИ ОПТИМИЗАЦИИ

Учебное пособие для подготовки бакалавров

Москва 2013

УДК
ББК

Рецензенты:
И.А. Александрова, доцент, к. ф.-.м. н. (Финуниверситет)
В.И. Троицкий, профессор, д. т. н. (МИИГАиК)

Гончаренко В.М., А.К. Керимов, В.Ю. Попов. Линейные модели оптимизации. Учебное пособие для подготовки бакалавров. М.: Финуниверситет, 2013. с.

В пособии рассматриваются различные линейные модели оптимизации, естественно возникающие в экономике, и основные методы их решения. Последовательно излагаются балансовые модели межотраслевого анализа; методы решения задач линейного программирования и их применение при решении экономических задач; теория взаимно-двойственных задач и основные типы транспортных задач; оптимизационные алгоритмы на графах.

Изложение сопровождается различными примерами задач, возникающих из практики, и составлено в полном соответствии с программой дисциплины «Линейная алгебра - 2», входящей в базовую часть математического цикла ФГОС по направлению «Экономика» и с программой дисциплины «Исследование операций», входящей в вариативную часть математического цикла ООП по направлению «Прикладная математика и информатика» подготовки бакалавров.

Пособие предназначено для подготовки бакалавров различных направлений Финуниверситета. Оно может быть использовано как для проведения практических занятий, так и для организации самостоятельной работы студентов.

ISBN

ВВЕДЕНИЕ

Пособие предназначено для подготовки бакалавров и содержит основные навыки

- составления математических моделей экономических задач, естественно возникающих из практики;
- практической реализации различных методов их решения;
- применения компьютерных пакетов для реализации оптимизационных алгоритмов.

В пособии последовательно излагаются балансовые модели межотраслевого анализа, рассматриваются примеры составления математических моделей, естественно возникающих в экономических задачах, подробно излагаются основные методы решения задач линейного программирования и применения теории взаимно двойственных задач, варианты постановки и методы решения транспортной задачи, изучаются оптимизационные алгоритмы на графах.

Изложение сопровождается различными примерами задач, возникающих из практики, и составлено в соответствии с программами дисциплин «Линейная алгебра-2», входящей в базовую часть математического цикла ФГОС по направлению «Экономика» и «Исследование операций» - дисциплиной вариативной части математического цикла ВПО по направлению «Прикладная математика и информатика» подготовки бакалавров.

ГЛАВА 1. НЕОТРИЦАТЕЛЬНЫЕ МАТРИЦЫ И ЛИНЕЙНЫЕ ЭКОНОМИЧЕСКИЕ МОДЕЛИ

В этой главе рассматриваются линейные балансовые модели (модель Леонтьева, модель международной торговли и т.д.), широко используемые в различных разделах макроэкономики. Для их изучения используются неотрицательные матрицы и их максимальное по модулю неотрицательное собственное значение – число Фробениуса и соответствующий собственный вектор.

1.1. Число и вектор Фробениуса

Квадратная $n \times n$ матрица $A = (a_{ij})$ называется *неотрицательной* ($A \geq 0$), если ее элементы неотрицательны: $a_{ij} \geq 0$.

Если все элементы матрицы A положительны, то она называется *положительной*: $A > 0$. Соответственно, вектор $x = (x_1, x_2, \ldots, x_n) \in R^n$ называется *положительным* (*неотрицательным*), если все его компоненты $x_i > 0$ ($x_i \geq 0$). Оказывается, у неотрицательных матриц существует собственное значение и собственный вектор, обладающие специальными свойствами.

Теорема 1.1 (Фробениуса-Перрона). Неотрицательная матрица A имеет такое собственное значение $\lambda_A \geq 0$, что $\lambda_A \geq |\lambda|$ для любого собственного значения λ матрицы A. Также существует неотрицательный собственный вектор $x_A \geq 0$, соответствующий собственному числу λ_A.

Определение 1.1. Собственное значение $\lambda_A \geq 0$ неотрицательной матрицы A называется *числом Фробениуса*, а собственный вектор $x_A \geq 0$ – *вектором Фробениуса* матрицы A.

Пример 1.1. Найти число и вектор Фробениуса матрицы $A = \begin{pmatrix} 3 & 5 \\ 5 & 3 \end{pmatrix}$.

Решение. Составив характеристическое уравнение для матрицы A

$$\begin{vmatrix} 3-\lambda & 5 \\ 5 & 3-\lambda \end{vmatrix} = 0,$$

найдем ее собственные значения: $\lambda_1 = 8$, $\lambda_2 = -2$ и соответствующие собственные вектора $x_1 = (t,t)^T$ и $x_2 = (t,-t)^T$, где $t \neq 0$. Таким образом, матрица A имеет число Фробениуса $\lambda_A = 8$ и вектор Фробениуса $x_A = (t,t)^T$, $t > 0$. При этом, очевидно, выполняется неравенство $\lambda_A > |\lambda_2|$.

Пример 1.2. Найти число Фробениуса матрицы $A \geq 0$, если сумма элементов любой строки A равна $\alpha > 0$.

Решение. Заметим, что условие задачи можно записать в виде матричного равенства

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ \dots \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \dots \\ \alpha \end{pmatrix}.$$

А так как $(\alpha,\dots,\alpha)^T = \alpha(1,\dots,1)^T$, то $\alpha > 0$ – собственное значение матрицы A с положительным вектором $(1,\dots,1)$. Таким образом, $\lambda_A = \alpha$ и $x_A = (t,\dots,t)^T$, $t > 0$.

Заметим, что задача из примера 1.1 удовлетворяет условию примера 1.2, и может быть решена, используя доказанное выше свойство. Оказывается, верно и более общее утверждение.

Если сумма элементов любой строки или любого столбца неотрицательной матрицы A равна α, то ее число Фробениуса $\lambda_A = \alpha$.

Замечание 1.1. Если сумма элементов любого столбца матрицы A равна положительному числу α, то, в общем случае, ее вектор Фробениуса $x_A \neq (t,\dots,t)^T$, $t > 0$, и требует отдельного вычисления.

Задачи для самостоятельного решения

1. Найдите число и вектор Фробениуса матрицы A, если

а) $A = \begin{pmatrix} 2 & 3 & 0 \\ 3 & 2 & 0 \\ 0 & 0 & 7 \end{pmatrix}$; б) $A = \begin{pmatrix} 1 & 3 & 4 \\ 8 & 0 & 0 \\ 2 & 3 & 3 \end{pmatrix}$;

2. Найдите число и вектор Фробениуса данной матрицы:

$$\text{а)} \begin{pmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix}; \text{ б)} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \text{ в)} \begin{pmatrix} 2 & 4 & 5 \\ 4 & 1 & 1 \\ 2 & 3 & 2 \end{pmatrix}; \text{ г)} \begin{pmatrix} 2 & 3 & 3 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix};$$

1.2. Балансовые модели и их продуктивность

Балансовые модели являются одним из основных инструментов экономического анализа. В частности, модель Леонтьева позволяет рассчитывать объемы валового выпуска нескольких отраслей производства, если известна их количественная связь между собой и объем конечного потребления. С другой стороны, с помощью модели равновесных цен можно, зная величины норм добавленной стоимости, прогнозировать цены на продукцию отраслей.

Пусть производственный сектор экономики состоит из n отраслей, каждая из которых производит однородный продукт. Обозначим x_{ij} – объем продукции i-ой отрасли, используемой в отрасли j, x_j – валовой выпуск отрасли j.

Рассмотрим *матрицу Леонтьева* $A = \left(a_{ij}\right)$, где $a_{ij} = \dfrac{x_{ij}}{x_j}$ – стоимость продукции отрасли i, затрачиваемой на производство единицы (стоимости) продукции отрасли j. Элементы a_{ij} матрицы A называют *коэффициентами прямых затрат*, и В.В. Леонтьев показал, что они достаточно устойчивы во времени.

Пусть $x = \left(x_1,\ x_2,\ \ldots,\ x_n\right)^T$ – вектор валового выпуска всех отраслей, а $d = \left(d_1, d_2, \ldots, d_n\right)^T$ – вектор конечного потребления. Тогда уравнения межотраслевого баланса (*уравнение Леонтьева*) в матричной форме имеют вид:

$$x = Ax + d. \tag{1.1}$$

Если известна матрица Леонтьева A и объемы конечного потребления d, то, используя уравнение (1.1), можно найти планируемые объемы валового выпуска x всех отраслей производства. Действительно, при условии невырожденности матрицы $\left(E - A\right)$ (1.1) можно переписать в виде

$$x = \left(E - A\right)^{-1} d.$$

Матрица $H = (E - A)^{-1}$ называется *матрицей коэффициентов полных затрат*, и основной результат балансового анализа можно представить в виде равенства

$$x = Hd\,.$$

Пример 1.3. Пусть в двухотраслевой модели дана матрица Леонтьева $A = \begin{pmatrix} 0,2 & 0,5 \\ 0,3 & 0,1 \end{pmatrix}$ и вектор конечного потребления $d = (50;\,30)^T$.

Найти объемы валового выпуска каждой отрасли. На сколько процентов должны измениться объемы валового выпуска каждой отрасли, если нужно удвоить выпуск конечного продукта второй отрасли?

Решение. Так как $A = \begin{pmatrix} 0,2 & 0,4 \\ 0,3 & 0,1 \end{pmatrix}$, то $E - A = \begin{pmatrix} 0,8 & -0,4 \\ -0,3 & 0,9 \end{pmatrix}$, и

$$H = (E - A)^{-1} = \frac{1}{0,6}\begin{pmatrix} 0,9 & 0,4 \\ 0,3 & 0,8 \end{pmatrix} = \begin{pmatrix} 3/2 & 2/3 \\ 1/2 & 4/3 \end{pmatrix}.$$

Тогда вектор валового выпуска равен

$$x = Hd = \begin{pmatrix} 3/2 & 2/3 \\ 1/2 & 4/3 \end{pmatrix}\begin{pmatrix} 50 \\ 30 \end{pmatrix} = \begin{pmatrix} 95 \\ 65 \end{pmatrix}.$$

Таким образом, валовой выпуск первой отрасли равен 95, второй – 65.

Если $d = (50;\,60)^T$, то новый валовый выпуск находится аналогично

$$x = Hd = \begin{pmatrix} 3/2 & 2/3 \\ 1/2 & 4/3 \end{pmatrix}\begin{pmatrix} 50 \\ 60 \end{pmatrix} = \begin{pmatrix} 115 \\ 105 \end{pmatrix}.$$

Таким образом, при изменении вектора конечного потребления-объем валового выпуска первой отрасли увеличивается на $\frac{115-95}{95}\cdot 100\% \approx 21.05\%$, а второй отрасли – на $\frac{105-65}{65}\cdot 100\% \approx 61.54\%$.

Двойственной к модели Леонтьева является *модель равновесных цен*, которая описывается равенством

$$p = A^{\mathrm{T}} p + v, \tag{1.2}$$

где $p = \left(p_1, \ldots, p_n\right)^{\mathrm{T}}$ – вектор цен, где p_i – цена единицы продукции i-ой отрасли, а $v = \left(v_1, \ldots, v_n\right)^{\mathrm{T}}$ – вектор *норм добавленной стоимости*. При этом, как легко показать, если известен вектор v, то цены находятся из уравнения

$$p = H^{T} v, \tag{1.3}$$

Пример 1.4. В трехотраслевой балансовой модели дана матрица Леонтьева

$$A = \begin{pmatrix} 0,1 & 0,1 & 0,2 \\ 0,3 & 0,2 & 0,2 \\ 0,2 & 0,2 & 0,4 \end{pmatrix}$$

и вектор норм добавленной стоимости по каждой отрасли $v = \left(3, 6, 9\right)$. Найти равновесные цены и изменение равновесных цен при увеличении нормы добавленной стоимости второй отрасли на 15%.

Решение. Найдем матрицу полных затрат H

$$H = \left(E - A\right)^{-1} = \begin{pmatrix} 0,9 & -0,1 & -0,2 \\ -0,3 & 0,8 & -0,2 \\ -0,2 & -0,2 & 0,6 \end{pmatrix}^{-1} = \begin{pmatrix} 4/3 & 10/33 & 6/11 \\ 2/3 & 50/33 & 8/11 \\ 2/3 & 20/33 & 23/11 \end{pmatrix}.$$

Откуда, согласно (1.3),

$$p = \begin{pmatrix} 4/3 & 2/3 & 2/3 \\ 10/33 & 50/33 & 20/33 \\ 6/11 & 8/11 & 23/11 \end{pmatrix} \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix} = \begin{pmatrix} 14 \\ 170/11 \\ 273/11 \end{pmatrix}.$$

После изменения нормы добавленной стоимости второй отрасли, находим равновесные цены в этом случае

$$p = \begin{pmatrix} 4/3 & 2/3 & 2/3 \\ 10/33 & 50/33 & 20/33 \\ 6/11 & 8/11 & 23/11 \end{pmatrix} \begin{pmatrix} 3 \\ 6,9 \\ 9 \end{pmatrix} = \begin{pmatrix} 73/5 \\ 185/11 \\ 1401/55 \end{pmatrix}.$$

Таким образом, продукция первой отрасли подорожала примерно на 14,29%, второй – на 8,82%, третьей – на 2,64%.

Естественно возникает вопрос об устойчивости экономики к изменениям вектора конечного потребления (норм добавленной стоимости). Для ответа на него введем следующее определение.

Матрица $A \geq 0$ называется *продуктивной*, если для любого вектора $y \geq 0$ существует решение $x \geq 0$ уравнения Леонтьева $x = Ax + y$.

Теорема 1.2 (первый критерий продуктивности**)**. *Матрица $A \geq 0$ продуктивна тогда и только тогда, когда матрица $(E - A)^{-1}$ существует и неотрицательна.*

Теорема 1.3 (второй критерий продуктивности). *Неотрицательная квадратная матрица A продуктивна тогда и только тогда, когда ее число Фробениуса меньше единицы.*

Следствие 1.1. *Если сумма элементов любого столбца (или любой строки) матрицы $A \geq 0$ меньше 1, то матрица A продуктивна.*

Используя следствие 1.1, легко доказать продуктивность некоторых матриц. Например, матрица $A = \begin{pmatrix} 0,1 & 0 & 0,5 \\ 0,3 & 0,6 & 0,1 \\ 0,4 & 0,2 & 0,3 \end{pmatrix}$ продуктивна, так как сумма элементов каждого из ее столбцов меньше 1.

Следствие 1.2. *Если для неотрицательной матрицы A и некоторого положительного вектора y^* уравнение $x = Ax + y^*$ имеет неотрицательное решение x^*, то матрица A продуктивна.*

Пусть $A \geq 0$ – продуктивная матрица. *Запасом продуктивности* матрицы A назовается такое число $\alpha > 0$, что все матрицы λA, где $1 < \lambda < 1 + \alpha$, продуктивны, а матрица $(1 + \alpha)A$ непродуктивна. Запас продуктивности α матрицы A можно найти по формуле

$$\alpha = \lambda_A^{-1} - 1. \qquad (1.4)$$

Пример 1.5. Исследовать на продуктивность матрицу $A = \begin{pmatrix} 0,1 & 0,6 \\ 1,0 & 0,3 \end{pmatrix}$.

Решение. Последовательно находим $E - A = \begin{pmatrix} 0,9 & -0,6 \\ -1,0 & 0,7 \end{pmatrix}$.

Тогда

$$(E - A)^{-1} = \frac{1}{0,03}\begin{pmatrix} 0,7 & 0,6 \\ 1,0 & 0,9 \end{pmatrix} = \begin{pmatrix} 70/3 & 20 \\ 100/3 & 30 \end{pmatrix}.$$

Матрица $(E-A)^{-1}$ неотрицательна, и, следовательно, по теореме 1.2 матрица A придуктивна.

Пример 1.6 ([6]). Выяснить, при каких значениях $a>0$ матрица $A=a\begin{pmatrix}1 & 2 & 0\\ 2 & 1 & 0\\ 7 & 6 & 9\end{pmatrix}$ будет продуктивной. Является ли матрица A продуктивной при $a=0{,}1$?

Решение. Найдем характеристический многочлен матрицы A:

$$|A-\lambda E|=\begin{vmatrix}a-\lambda & 2a & 0\\ 2a & a-\lambda & 0\\ 7a & 6a & 9a-\lambda\end{vmatrix}=$$

$$=(9a-\lambda)\left((a-\lambda)^2-4a^2\right)=(9a-\lambda)\left(\lambda^2-2a\lambda-3a^2\right),$$

Корнями характеристического уравнения являются числа $\lambda_1=9a$, $\lambda_2=3a$ и $\lambda_3=-a$. Таким образом, так как $a>0$, числом Фробениуса матрицы A будет $\lambda_A=9a$.

Для продуктивности матрицы A, согласно теореме 1.3, необходимо и достаточно, чтобы выполнялось условие $9a<1$, т.е. $a<\dfrac{1}{9}$. Число $a=\dfrac{1}{10}$ удовлетворяет последнему условию, и при $a=\dfrac{1}{10}$ получим продуктивную матрицу $A=\begin{pmatrix}0{,}1 & 0{,}2 & 0\\ 0{,}2 & 0{,}1 & 0\\ 0{,}7 & 0{,}6 & 0{,}9\end{pmatrix}$.

Пример 1.7. Найти запас продуктивности матрицы $A=\begin{pmatrix}0{,}1 & 0{,}6\\ 0{,}8 & 0{,}3\end{pmatrix}$.

Решение. Так как сумма каждого из столбцов матрицы A равна 0,9, то ее число Фробениуса равно $\lambda_A=0{,}9$. Запас продуктивности можно найти по формуле (1.4):

$$\alpha=\lambda_A^{-1}-1=10/9-1=1/9.$$

Задачи для самостоятельного решения [6]

3. Дана балансовая таблица в двухотраслевой модели

Производители	Потребители		Потребление	Валовой выпуск
	I	II		
I	15	60	25	100
II	25	5	20	50

Постройте структурную матрицу и рассчитать валовой выпуск на новый вариант потребления: $d = (20, 25)$.

4. Для балансовой таблицы

Производители	Потребители		Потребление	Валовой выпуск
	I	II		
I	5	60	35	100
II	25	5	20	50
III	50	10	–	60

рассчитайте выпуск загрязняющих веществ, соответствующий варианту потребления $d = (45, 15)$.

5. Приведите пример продуктивной матрицы *A*, для которой одна из отраслей нерентабельна.

6. В двухотраслевой модели дана матрица Леонтьева A и вектор конечного потребления d: $A = \begin{pmatrix} 0,4 & 0,2 \\ 0,3 & 0,15 \end{pmatrix}$, $d = (90; 45)^T$. Найдите соответствующие объемы валового выпуска каждой отрасли. Если выпуск конечного продукта второй отрасли удваивается, то на сколько процентов должны измениться объемы валового выпуска каждой отрасли?

7. Для трехотраслевой балансовой модели дана матрица Леонтьева A и вектор норм добавленной стоимости по каждой отрасли $\vec{v}$:

$$A = \begin{pmatrix} 0,2 & 0,1 & 0,2 \\ 0,1 & 0,3 & 0,2 \\ 0,2 & 0,2 & 0,3 \end{pmatrix}, \ \vec{v} = (10; 2; 6).$$

а) Найдите равновесные цены.

б) Пусть произошло увеличение нормы добавленной стоимости первой отрасли на 1,1. На сколько процентов возрастут равновесные цены каждой отрасли?

8. Используя первый критерий продуктивности матрицы, исследуйте на продуктивность матрицу A:

а) $A=\begin{pmatrix}0,5 & 0,6\\ 0,4 & 0,5\end{pmatrix}$; б) $A=\begin{pmatrix}0,5 & 0,2 & 0,3\\ 0,3 & 0,5 & 0,2\\ 0,2 & 0,4 & 0,5\end{pmatrix}$;

в) $A=\begin{pmatrix}0,3 & 0,5 & 0,1\\ 0,5 & 0,2 & 0,3\\ 0,2 & 0,4 & 0,5\end{pmatrix}$.

9. Используя второй критерий продуктивности, установите продуктивность матрицы A и найдите ее запас продуктивности

а) $A=\begin{pmatrix}0,1 & 0 & 0,5\\ 0,3 & 0,8 & 0,2\\ 0,5 & 0,1 & 0,2\end{pmatrix}$; б) $A=\begin{pmatrix}0,1 & 0,3 & 0,5\\ 0,3 & 0,4 & 0,2\\ 0,7 & 0,1 & 0,1\end{pmatrix}$.

10. Выясните, при каких значениях $a>0$ матрица A продуктивна:

$$A=a\begin{pmatrix}1 & 2 & 4\\ 3 & 2 & 5\\ 0 & 0 & 6\end{pmatrix}?$$

ГЛАВА 2. ВВЕДЕНИЕ В ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

В этой главе рассматриваются постановка задачи, примеры и основные методы решения задач линейного программирования. Для задач линейного программирования существует универсальный метод решения – симплекс-метод, который приводит к точному решению задачи.

2.1. Постановка задачи линейного программирования

Пусть M – некоторое множество, точка $x \in M$ и $f(x)$ – функция на M. Тогда задача

$$f(x) \to \max(\min),\ x \in M$$

называется *задачей оптимизации.* Множество M называется *допустимым множеством*, а функция $f(x)$ – *целевой функцией*. Точку $x_{\max}$ или $x_{\min}$, в которой достигается оптимальное значение, называют *оптимальным решением*, а множество всех оптимальных точек X^* – *оптимальным множеством*.

Задача линейного программирования (далее: задача ЛП) является важным частным случаем общей задачи оптимизации и имеет вид

$$f = c_1x_1 + c_2x_2 + \ldots + c_nx_n + c_0 \to \max\ (\min)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n * b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n * b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n * b_m, \\ x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0. \end{cases} \tag{2.1}$$

где «*» обозначает «$\ge$», «$\le$» или «=». При этом условия типа

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n * b_i,\ i = 1,\ldots,m$$

называются *нетривиальными ограничениями*; условия $x_j \geq 0$, $j = 1,\ldots,n$ – *тривиальными ограничениями* (они могут отсутствовать). Таким образом, целевая функция задачи (2.1) имеет вид

$$f = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + c_0$$

и является линейной функцией, а система ограничений задает в пространстве R^n *выпуклое многогранное множество* M.

Замечание 2.1. Заметим, что нетривиальные *ограничения типа неравенств*, т.е. вида $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq b_i$, сводятся к условию $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i$ умножением на (-1), а *условие типа равенств* $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$ можно представить как систему ограничений

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i, \\ a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq b_i, \end{cases} \text{или} \begin{cases} a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i, \\ -a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n \leq -b_i, \end{cases}$$

При этом задачи оптимизации $f(x) \to \max$ и $f(x) \to \min$ также сводятся друг к другу умножением на (-1), поэтому в дальнейшем при общей постановке задачи мы будем, не нарушая общности рассуждений, рассматривать одно из них.

Задачу ЛП можно записать в матричном виде: введем строку $C = (c_1, c_2, \ldots, c_n)$, столбец щ(индекс обозначает транспонирование), $m \times n$ матрицу $A = (a_{ij})$ и столбец $B = (b_1, b_2, \ldots, b_m)^T$. Тогда задача ЛП может быть записана в виде

$$f = CX + c_0 \to \max$$

$$\begin{cases} AX \leq B, \\ X \geq 0. \end{cases}$$

Если система ограничений состоит только из уравнений и тривиальных неравенств, то говорят, что задача имеет *каноническую форму*. Если же в системе ограничений имеются только неравенства, то говорят о *стандартной форме* задачи линейного программирования. В зависимости от метода решения, задачу необходимо привести к канонической или стандартной форме.

Пример 2.1. Привести к канонической форме задачу линейного программирования

$$z = 4x_1 + 8x_2 - 10x_3 - 3x_4 - x_5 \to \max$$
$$\begin{cases} x_1 - 3x_2 + x_3 - 4x_4 \le 11, \\ 3x_1 + 3x_2 - x_3 + 8x_5 = 15, \\ 10x_1 + 8x_2 + 4x_3 - 3x_4 - 2x_5 \ge 25, \\ x_i \ge 0, i = 1, \dots, 5. \end{cases}$$

Решение. Согласно входящим в ограничения неравенствам, можно ввести *балансовые* переменные $x_6 \ge 0, x_7 \ge 0$, такие, что эти неравенства приводятся к виду

$$x_1 - 3x_2 + x_3 - 4x_4 + x_6 = 11,$$
$$10x_1 + 8x_2 + 4x_3 - 3x_4 - 2x_5 - x_7 = 25$$

и исходная задача переписывается в канонической форме

$$z = 4x_1 + 8x_2 - 10x_3 - 3x_4 - x_5 \to \max$$
$$\begin{cases} x_1 - 3x_2 + x_3 - 4x_4 + x_6 = 11, \\ 3x_1 + 3x_2 - x_3 + 8x_5 = 15, \\ 10x_1 + 8x_2 + 4x_3 - 3x_4 - 2x_5 - x_7 = 25, \\ x_i \ge 0, i = 1, \dots, 5. \end{cases}$$

Пример 2.2. Привести к стандартной форме задачу линейного программирования

$$z = x_1 - x_2 + 4x_3 + x_4 \to \max$$
$$\begin{cases} x_1 + 2x_2 + 5x_3 - x_4 = 4, \\ x_1 - x_2 - x_3 + 2x_4 = 1, \\ x_i \ge 0, i = 1, \dots, 4. \end{cases}$$

Решение. В системе нетривиальных ограничений

$$\begin{cases} x_1 + 2x_2 + 5x_3 - x_4 = 4, \\ x_1 - x_2 - x_3 + 2x_4 = 1, \end{cases}$$

выделим (методом Гаусса) базисные и свободные переменные:

$$\left(\begin{array}{cccc|c} 1 & 2 & 5 & -1 & 4 \\ 1 & -1 & -1 & 2 & 1 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 2 & 5 & -1 & 4 \\ 0 & -3 & -6 & 3 & -3 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & -1 & 1 \end{array}\right).$$

В полученной системе

$$\begin{cases} x_1 + x_3 + x_4 = 2, \\ x_2 + 2x_3 - x_4 = 1, \end{cases}$$

выразим базисные неизвестные

$$\begin{cases} 2 - x_3 - x_4 = x_1 \ge 0, \\ 1 - 2x_3 + x_4 = x_2 \ge 0. \end{cases}$$

Исключая их из целевой функции

$$z = x_1 - x_2 + 4x_3 + x_4 = 2 - x_3 - x_4 - (1 - 2x_3 + x_4) + 4x_3 + x_4 = 3x_3 - x_4 + 1,$$

получим задачу в стандартной форме, равносильную исходной:

$$z = 3x_3 - x_4 + 1 \to \max$$

$$\begin{cases} x_3 + x_4 \le 2, \\ 2x_3 - x_4 \le 1, \\ x_3 \ge 0, x_4 \ge 0. \end{cases}$$

Задачи для самостоятельного решения

Найти каноническую форму следующих задач линейного программирования

11. $\begin{cases} z = -2x_1 + 7x_2 + 3x_3 - x_5 \to \max \\ x_1 + x_3 - x_4 - x_5 \ge 15, \\ -x_1 + 3x_2 - x_3 + 4x_5 \le 10, \\ 2x_1 + 5x_2 - 42x_3 + 6x_4 \ge 34, \\ x_i \ge 0, i = 1, \ldots, 5. \end{cases}$

12. $\begin{cases} z = -9x_1 + 2x_2 - 5x_3 - x_4 \to \max \\ 17x_1 + x_3 + 4x_5 = 31, \\ 12x_1 + 13x_2 + 14x_3 - 15x_4 \ge 18, \\ x_1 + 4x_2 + 11x_3 - 6x_5 \le 6, \\ x_1 \ge 0, x_3 \le 0, x_4 \ge 0. \end{cases}$

13. $\begin{cases} z = 3x_1 - x_4 + x_5 \to \min \\ -3x_1 - x_2 - 5x_3 + 4x_6 \le 22, \\ 2x_1 + 3x_3 - 4x_4 + 7x_5 = 18, \\ 3x_2 - 5x_4 + 2x_5 - 6x_6 \ge 27, \\ x_1 \ge 0, x_2 \ge 0, x_4 \ge 0, x_6 \le 0. \end{cases}$

Привести задачи линейного программирования к стандартному виду:

$$14.\quad \begin{cases} z = x_1 - 2x_2 - 2x_3 - 3x_4 \to \max \\ x_1 - 2x_2 - 3x_3 + 3x_4 = 0, \\ x_1 - x_2 + 2x_3 - 3x_4 = -3, \\ x_i \ge 0, i = 1,\dots,4. \end{cases}$$

$$15.\ \begin{cases} z = 3x_2 + x_4 \to \max \\ x_1 - x_2 + 3x_3 - x_4 = -3, \\ 4x_1 - 3x_2 + 3x_3 + 2x_4 = -2, \\ x_i \ge 0, i = 1,\dots,4. \end{cases}$$

$$16.\ \begin{cases} z = 3x_1 - x_2 + x_3 + x_4 \to \max \\ x_1 + x_2 - x_3 - 2x_4 = 2, \\ -5x_1 - 4x_2 - 3x_3 - 3x_4 = 2, \\ x_i \ge 0, i = 1,\dots,4. \end{cases}$$

2.2. Примеры задач линейного программирования

Рассмотрим теперь примеры задач, математическая модель которых сводится к задаче ЛП.

Пример 2.3 (задача о банке). Пусть собственные средства банка составляют 400 у.е. Часть этих средств, но не менее 200 у.е., должна быть размещена в кредитах, а вложения в ценные бумаги должны составлять не менее 30% средств, размещенных в кредитах и ценных бумагах. Если c_1 – доходность кредитов, c_2 – доходность ценных бумаг (как правило, $c_1 > c_2$), то каково должно быть размещение средств, чтобы прибыль банка была максимальной?

Решение. Если x_1 – средства, размещенные в кредитах, а x_2 – средства, вложенные в ценные бумаги, то прибыль банка можно записать в виде $f = c_1x_1 + c_2x_2$. Учитывая ограничения на средства банка, размещения в кредитах и ликвидное ограничение, получим следующую задачу ЛП

$$f = c_1x_1 + c_2x_2 \to \max$$

$$\begin{cases} x_1 + x_2 \le 400, \\ x_1 \ge 200, \\ x_2 \ge 0.3(x_1 + x_2), \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$$

Пример 2.4 (транспортная задача). На двух складах A_1 и A_2 имеется 70 и 120 тонн некоторого товара. Потребности магазинов B_1,

B_2, B_3 в этом товаре равны 50, 80 и 60 тонн соответственно. Тарифы перевозок заданы матрицей $C = \begin{pmatrix} 2 & 13 & 8 \\ 7 & 10 & 5 \end{pmatrix}$, где c_{ij} – стоимость перевозки одной тонны (единицы) товара со склада A_i в магазин B_j. Найти минимальный по стоимости план перевозки товара со складов в магазины.

Решение. Пусть x_{ij} – количество товара в тоннах, предназначенное к перевозке из *i*-го склада в *j*-й магазин. Матрица перевозок

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

определяет план перевозок.

Замечание 2.2. В данном примере суммарные запасы 70+120=190 равны суммарным потребностям 50 + 80 + 60. Такая задача называется *транспортной задачей с правильным балансом*, т.е. все товары со складов должны быть вывезены, а потребности магазинов полностью удовлетворены.

Условия замечания 2.2 приводят к уравнениям

$$x_{11} + x_{12} + x_{13} = 70,$$
$$x_{21} + x_{22} + x_{23} = 120$$

и

$$x_{11} + x_{21} = 50,$$
$$x_{12} + x_{22} = 80,$$
$$x_{13} + x_{23} = 60.$$

Суммарная стоимость всех перевозок определяется функцией

$$f = \sum c_{ij} x_{ij} = 2x_{11} + 13x_{12} + 8x_{13} + 7x_{21} + 10x_{22} + 5x_{23}.$$

В итоге получаем каноническую задачу ЛП

$$f = 2x_{11} + 13x_{12} + 8x_{13} + 7x_{21} + 10x_{22} + 5x_{23} \to \min$$

$$\begin{cases} x_{11} + x_{12} + x_{13} = 70, \\ x_{21} + x_{22} + x_{23} = 120, \\ x_{11} + x_{21} = 50, \\ x_{12} + x_{22} = 80, \\ x_{13} + x_{23} = 60, \\ x_{ij} \ge 0, i = 1,2; j = 1,2,3. \end{cases}$$

Пример 2.5 (задача о диете). Известно, что 1 кг яблок стоит 60 руб., а 1 кг абрикосов 120 руб. Сколько яблок и абрикосов должен потреблять человек в сутки, чтобы получить не менее 80 мг витамина С и не менее 3 мг витамина А при минимальных затратах на яблоки и абрикосы? Содержание витаминов А и С в яблоках и абрикосах указано в таблице 2.1.

	А (мг/кг)	С (мг/кг)
Яблоки	1	30
Абрикосы	24	45
Таблица 2.1		

Р е ш е н и е. Все данные задачи удобно представить в виде таблицы

x_1	1	30	60
x_2	24	45	120
	3	80	

где x_1, x_2 – необходимое количество яблок и абрикосов. Тогда условие получения необходимого количества витаминов А и С можно записать в виде

$$1 \cdot x_1 + 24x_2 \ge 3,$$
$$30x_1 + 45x_2 \ge 80,$$

а стоимость купленных продуктов равна $f = 60x_1 + 120x_2$. Получаем задачу ЛП в стандартной форме

$$f = 60x_1 + 120x_2 \to \min$$

$$\begin{cases} x_1 + 24x_2 \ge 3, \\ 30x_1 + 45x_2 \ge 80, \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$$

Пример 2.6 (задача об использовании ресурсов). Для изготовления двух видов продукции P_1, P_2 используется три вида ресурсов R_1, R_2, R_3. Пусть $A = \begin{pmatrix} 2 & 3 \\ 5 & 2 \\ 4 & 8 \end{pmatrix}$ – матрица норм расхода сырья, где a_{ij} – количество (в кг) ресурса R_i, необходимого для производства единицы продукта P_j. Известно, что доход от реализации единицы продукта P_1 равен 20 у.е., а P_2 – 25 у.е., а запасы ресурсов R_1, R_2, R_3 равны соответственно 300, 450 и 350 кг. Найти план производства, максимизирующий доход предприятия от реализации выпускаемой продукции.

Решение. Если x_1, x_2 – план выпуска продукции P_1, P_2, то доход от ее реализации определяется функцией $f = 20x_1 + 25x_2$, а в силу ограниченности ресурсов имеем следующую систему нетривиальных ограничений

$$\begin{cases} 2x_1 + 3x_2 \le 300, \\ 5x_1 + 2x_2 \le 450, \\ 4x_1 + 8x_2 \le 350. \end{cases}$$

Таким образом, получаем следующую задачу ЛП

$$\begin{cases} f = 20x_1 + 25x_2 \to \max \\ \quad 2x_1 + 3x_2 \le 300, \\ \quad 5x_1 + 2x_2 \le 450, \\ \quad 4x_1 + 8x_2 \le 350, \\ \quad x_1 \ge 0, x_2 \ge 0. \end{cases}$$

В общем случае, если $X = (x_1, x_2, \ldots, x_n)^T$ – план выпуска продуктов $P_1, P_2, \ldots, P_n$, $A = (a_{ij})$ – матрица норм расхода сырья, $C = (c_1, c_2, \ldots, c_n)$ – вектор доходов от реализации единицы этих продуктов, а $B = (b_1, b_2, \ldots, b_m)^T$ – запасы ресурсов $R_1, R_2, \ldots, R_n$ то доход предприятия от его реализации равен $f = CX$. Модель соответсвующей задачи об использовании ресурсов имеет вид

$$f = CX \to \max$$

$$\begin{cases} AX \le B, \\ X \ge 0. \end{cases}$$

Пример 2.7. Стальные прутья длиной 100 см требуется разрезать на заготовки длиной 50, 45 и 30 см. Заготовок длиной 50 должно быть изготовлено не менее 15, длиной 45 – не менее 35, длиной 30 – не менее 50. Сколько прутьев и каким способом следует разрезать, чтобы получить необходимое число заготовок при минимальных отходах?

Решение. Для решения задачи сначала необходимо найти все возможные варианты разреза. Как видно из таблицы 2.2, возможно 6 вариантов разрезов.

Варианты разреза	Количество заготовок			Отходы
	50	45	30	
x_1	2	–	–	0
x_2	–	2	–	10
x_3	1	–	1	20
x_4	–	–	3	10
x_5	–	1	1	25
x_6	1	1	–	5

Таблица 2.2

Пусть x_i – количество прутьев, разрезаемых по каждому из вариантов. Числа $x_1, x_2, ..., x_6$ составляют план разреза. Тогда из условий задачи вытекают следующие ограничения на неизвестные $x_1, x_2, ..., x_6$:

$$\begin{cases} 2x_1 + x_3 + x_5 \geq 15, \\ 2x_2 + x_5 + x_6 \geq 35, \\ x_3 + 3x_4 + 2x_5 \geq 60, \\ x_i \geq 0, \ i = 1, 2, ..., 6. \end{cases}$$

Общее количество отходов записывается функцией

$$f = 10x_2 + 20x_3 + 10x_4 + 25x_5 + 5x_6.$$

Получаем следующую стандартную задачу ЛП:

$$f = 10x_2 + 20x_3 + 10x_4 + 25x_5 + 5x_6 \to \min$$

$$\begin{cases} 2x_1 + x_3 + x_5 \geq 15, \\ 2x_2 + x_5 + x_6 \geq 35, \\ x_3 + 3x_4 + 2x_5 \geq 60, \\ x_i \geq 0, \ i = 1, 2, ..., 6. \end{cases}$$

Задачи для самостоятельного решения

Составьте математическую модель следующих задач ЛП:

17. Для изготовления изделий двух видов имеется 100 кг сырья. На изготовление одного изделия первого вида расходуется 2 кг сырья, а изделия второго вида – 4 кг. Составьте план производства, обеспечивающий получение наибольшей выручки от продажи изделий, если отпускная стоимость одного изделия первого вида составляет 3000 руб., а изделия второго вида – 2000 руб., причем изделий первого вида требуется изготовить не более 40, а изделий второго вида – не более 20.

18. Известно, что 1 кг яблок стоит 30 руб., а 1 кг абрикосов 60 руб. Сколько яблок и абрикосов должен потреблять человек в сутки, чтобы получить не менее 70 мг витамина С и не менее 2 мг витамина А при минимальных затратах на яблоки и абрикосы? Содержание витаминов А и С в яблоках и абрикосах указано в таблице 2.3.

	А (мг/кг)	С (мг/кг)
Яблоки	1	70
Абрикосы	24	75

Таблица 2.3

19. Стальные прутья длиной 110 см необходимо разрезать на заготовки длиной 45 и 35 см. Требуемое количество заготовок данного вида составляет соответственно 40 и 30 штук. Определите, сколько прутьев по каждому из возможных способов следует разрезать, чтобы получить не менее необходимого количества заготовок каждого вида при минимальных отходах.

20. Чаеразвесочная фабрика выпускает чай сортов А и Б, смешивая три ингредиента: индийский, цейлонский и английский чаи. В таблице 2.4 приведены нормы расхода ингредиентов, объем запасов каждого ингредиента и прибыль от реализации 1 т чая сортов А и Б.

Ингредиенты	Норма расхода (т/т)		Объем запасов (т)
	А	Б	
Индийский чай	0,5	0,2	600
Цейлонский чай	0,2	0,6	870
Английский чай	0,3	0,2	430
Прибыль от реализации одной тонны продукции (тыс. руб.)	320	290	

Таблица 2.4

Требуется составить план производства чая сортов А и Б с целью получения максимальной прибыли.

21. Завод специализируется на капитальном ремонте автомашин типов А и В. Производственные мощности цехов приведены в таблице 2.5.

Определите наиболее рентабельную производственную программу при условии, что прибыль от ремонта одной машины типов А и В соответственно равна 20 000 и 24 000 руб.

Название цехов	Количество машин за год	
	Типа А	Типа В
Кузовной	80	320
Шасси	110	110
Двигательный	240	120
Сборочный	160	80

Таблица 2.5

22. Для обеспечения перевозок нужно ежедневно формировать пассажирские и скорые поезда. В таблице 2.6 указаны наличный парк вагонов разных типов, из которых комплектуются данные поезда, и количество пассажиров, вмещающихся в каждый из типов вагонов:

Поезда	Вагоны				
	Багажный	Почтовый	Плацкартный	Купейный	Спальный
Скорый	1	1	5	6	3
Пассажирский	1	–	8	4	1
Число пассажиров	–	–	58	40	32
Парк вагонов	12	8	81	70	26

Таблица 2.6

Определите число скорых и пассажирских поездов, при которых количество перевозимых пассажиров достигает максимума.

2.3. Решения задач линейного программирования средствами EXCEL

В EXCEL для решения задач линейного программирования используется программа **Поиск решения**. Ниже на конкретной задаче ЛП подробно описываются все этапы, необходимые для применения этой программы.

2.3.1. Постановка задачи

Рассмотрим следующую задачу.

Машиностроительное предприятие для изготовления 4-х видов продукции использует токарное, фрезерное, сверлильное, расточное и шлифовальное оборудование, а также комплектующие изделия.

Кроме того, сборка изделий требует выполнения определенных сборочно-наладочных работ. Нормы затрат всех видов ресурсов приведены в таблице вместе с ограничениями на выпуск продукции и доступными запасами ресурсов приведены в таблице ниже. Найти план выпуска продукции, максимизирующий выручку предприятия. Объем продукции измеряется в штуках.

Ресурсы	**1**	**2**	**3**	**4**	**Запасы**
Производительность оборудования (ч-ч) :					
токарного	550	0	620	0	64270
фрезерного	40	30	20	20	4800
сверлильного	86	110	150	52	22360
расточного	160	92	158	128	26240
шлифовального	0	158	30	50	7900
Комплетующие изделия (шт.)	3	4	3	3	520
Сборочно-наладочные работы (ч-ч)	4,5	4,5	4,5	4,5	720
Выпуск (шт.):					
минимальный	0	40	0	0	0
максимальный			120		
Цена изделия (т. руб.) – p	315	278	573	370	

Таблица 2.7. Исходные данные

Пусть $x = \left(x_1, x_2, x_3, x_4\right)^T$ – план выпуска продукции, *k*-я компонента которого равна объему выпуска изделия номера *k*, $k = 1, 2, 3, 4$. При заданном плане *x*, объем потребляемых ресурсов определяется произведением технологической матрицы *A* на *x*, то есть компоненты вектора *Ax* определяют объемы потребляемых ресурсов при заданном плане. Таким образом, математическая модель задачи оптимизации плана имеет вид:

$$f\left(x\right) = p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 \to \max$$

$$\begin{cases} Ax \le b, \\ x \ge 0, x_2 \ge 40, x_3 \le 120, \\ x \in Z \end{cases} \tag{2.2}$$

Компоненты вектора цен *p* заданы последней строкой таблицы 2.7, технологическая матрица *A* в таблице выделена, вектор-столбец *b* составлен из элементов столбца «запасы».

2.3.2. Табличное представление задачи.

Размещение исходных данных и необходимых формул на листе EXCEL представлено в таблице 2.8.

	B	C	D	E	F	G	H
5	Ресурсы	Нормы затрат на одно изделие				Запасы	Расход по плану
6		1	2	3	4		
7	Производительность (ч-ч) :						
8	токарного	550	0	620	0	64270	64270
9	фрезерного	40	30	20	20	4800	4800
10	сверлильного	86	110	150	52	22360	17098
11	расточного	160	92	158	128	26240	21860
12	шлифовального	0	158	30	50	7900	7900
13	Комплетующие изделия (шт.)	3	4	3	3	520	505
14	Сборочно-наладочные работы (ч-ч)	4,5	4,5	4,5	4,5	720	697,5
15	**Выпуск (шт.):**						
16	минимальный		40				
17	максимальный			120			
18	Цена изделия (тыс. руб.)	315	278	573	370		
19	**План производства (шт.)**						
20	Изделие 1	65					
21	Изделие 2	40					
22	Изделие 3	46					
23	Изделие 4	4					
24	**Выручка**	**59433**	<----- =МУМНОЖ(C18:F18;C20:C23)				

Таблица 2.8

План размещается в столбце C20:C23. Выручка по заданному плану размещается в ячейке C24. Для расчета выручки используется функция умножения матриц МУМНОЖ, которая умножает строку C18:F18 на столбец C20:C23.

Столбец «расход по плану» таблицы 2.8 получается умножением матрицы A = {C8:F14} на столбец x = {C20:C23}, представляющий план. Расход по плану определяется умножением матрицы A на столбец x с использованием функции МУМНОЖ. В этом столбце размещаются ресурсы, необходимые для реализации плана, определенного в ячейках C20:C23.

Алгоритм получения этого столбца следующий.

1. Выделить столбец H8:H14.

2. Вызвать функцию МУМНОЖ, используя вкладку «Формулы» и кнопку «Вставить функцию» раздела библиотека функций. Искомая функция находится в разделе «Математические» (рис. 2.1)

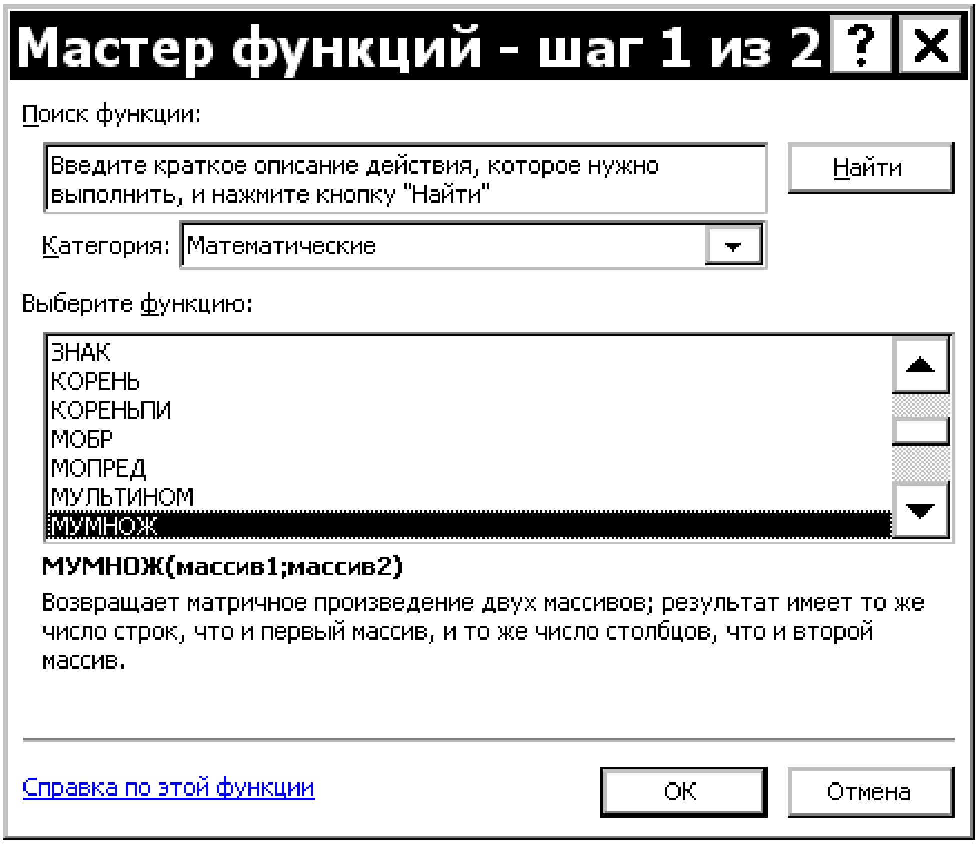


Рис. 2.1.

3. Вызов этой функции приводит к появлению диалогового окна Аргументы функции (рис. 2.2) . В окно **масс ив1** вводится матрица *A*, для этого достаточно выделить эту матрицу, удерживая правую кнопку мыши или ввести в окно C8:F14. Аналогично ввести в окно **массив2** столбец C20:C23.

4. Нажать комбинацию Ctr+Shift+Enter.

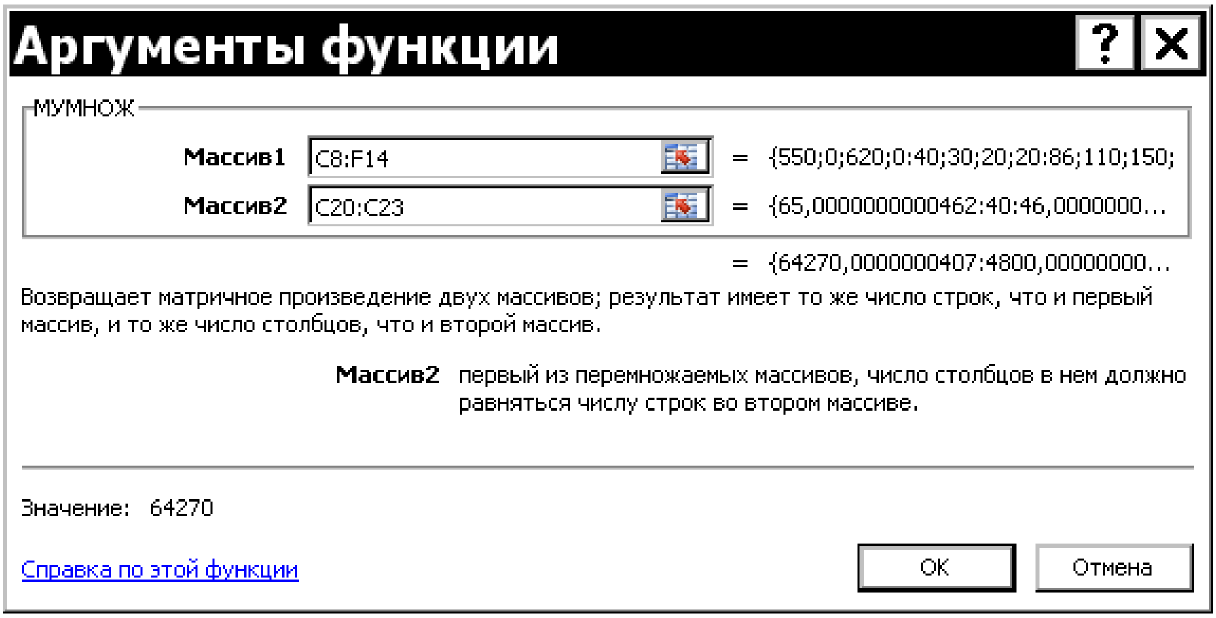


Рис. 2.2. Определение столбца «Расход по плану» таблицы 2.8.

2.3.3. Настройка программы Поиск решения.

После подготовки табличного представления задачи переходят к решению задачи оптимизации с помощью программы **Поиск решения**. Для ее вызова необходимо выбрать вкладку **Данные** главного меню, программа *Поиск решения* располагается в разделе **Анализ**. Диалоговое окно **Поиска решения**, представлено на рис.3.

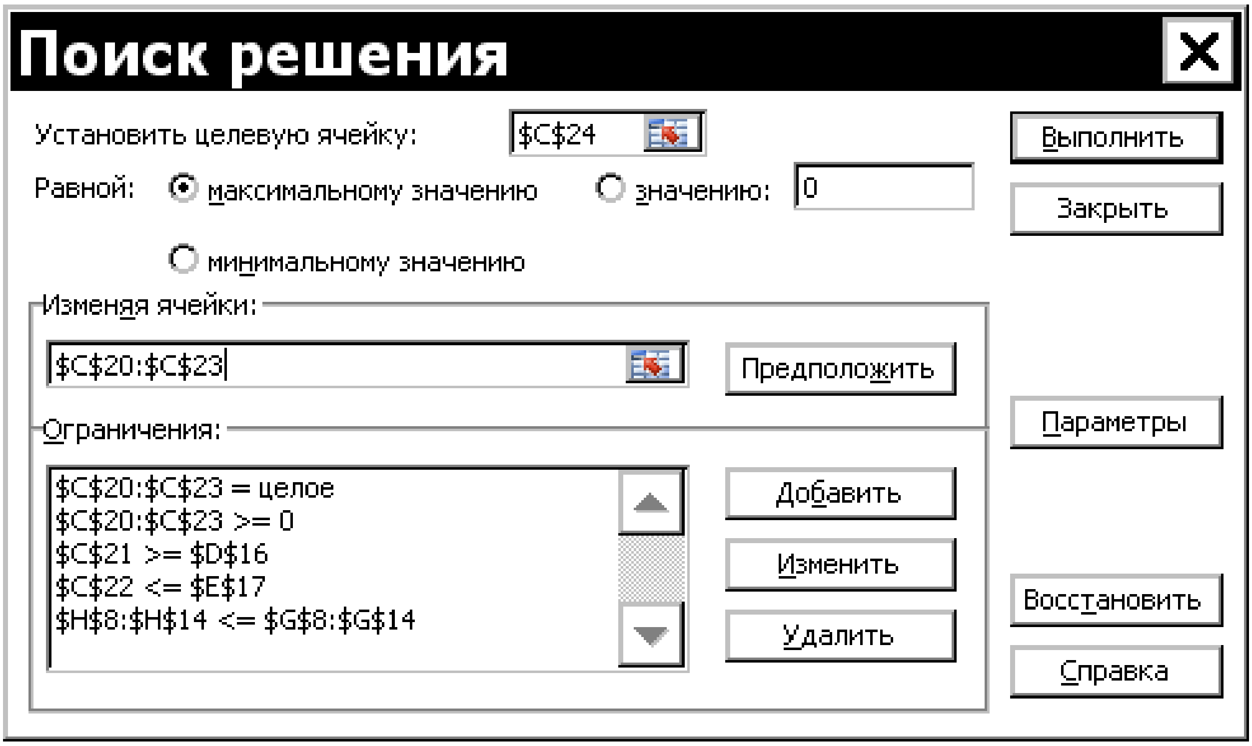


Рис. 2.3. Диалоговое окно *Поиска решения*.

Поле **Установить целевую ячейку** служит для указания целевой ячейки, значение которой необходимо максимизировать, минимизировать или установить равным заданному числу. В нашем примере это ячейка C24 (выручка).

Кнопка **Равной** используется для выбора варианта оптимизации целевой функции: максимизации, минимизации или установить равным заданному числу. В нашем случае для максимизации выручки нажимается кнопка **максимальному значению**.

В поле **Изменяя ячейки** указываются ячейки, значения в которых изменяются в процессе поиска до тех пор пока не будет найдено оптимальное решение. В нашем примере изменяется план, поэтому в это поле вводится диапазон ячеек C20:C23. Изменяемые ячейки должны быть связаны с целевой ячейкой. Допускается до 200 изменяемых ячеек, то есть максимальная размерность экстремальной задачи равна 200.

Поле ограничений используется для отображения ограничений экстремальной задачи. В нашем случае **Поле ограничений** содержит пять строк, и они отображают ограничения задачи ЛП (2.2):

1-я строка требует целочисленность плана (диапазон ячеек C20:C23 должен содержать целые числа);

2-я строка означает неотрицательность компонент плана;

3-я строка означает, что число изделий номера 2 в плане должно быть не меньше содержимого ячейки D16, то есть 40 единиц;

4-я строка означает, что число изделий номера 3 в плане должно быть не больше содержимого ячейки E17, то есть 120 единиц;

5-я строка отображает тот факт, что расход ресурсов плана H8:H14, не должен превышать его запасов G8:G14.

2.3.4. Ввод и редактирование ограничений

Чтобы приступить к набору нового условия или изменению выбранной строки ограничений необходимо нажать кнопку **Добавить** или **Изменить**. Диалоговые окна изменения и добавления ограничений одинаковы см. рис. 2.3.

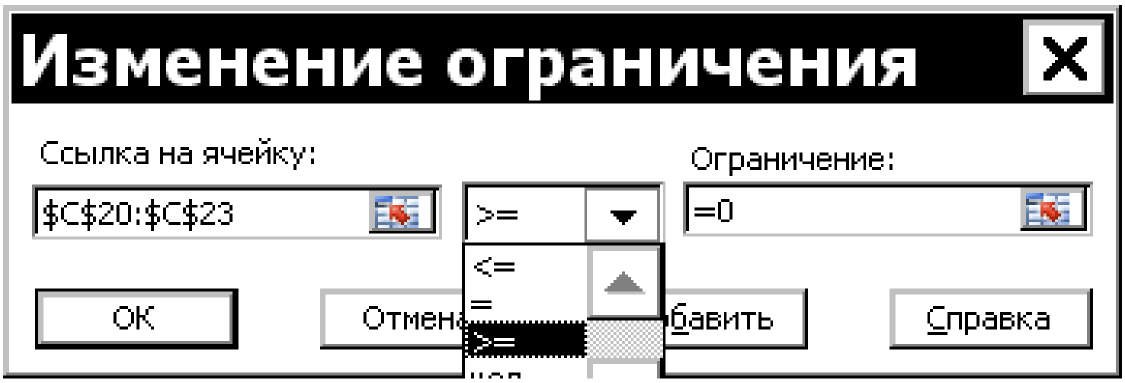


Рис. 2.4. Добавление и изменение ограничений.

В поле **Ссылка на ячейку** вводится диапазон ячеек, на значения которых накладываются ограничения. Диапазон ячеек можно вводить либо набрав его в окне вручную, либо выделив его с помощью мыши. Знаки доллара перед адресами появляются автоматически.

Из раскрывающегося списка выбирается условный оператор, который необходимо разместить между ссылкой и ограничением на нее. Это знаки: <= (не более), => (не менее), = (равно), цел (целое) и двоич (двоичное). Отметим, что ограничения типа целое и двоичное можно применять только при наложении ограничений на изменяемые ячейки.

В поле **Ограничение** вводится число или диапазон, содержащий или вычисляющий ограничивающие значения.

2.3.5. Запуск и сохранения результатов.

Для запуска **Поиска решения** необходимо нажать кнопку **Выполнить**. По окончании счет появится диалоговое окно **Результаты поиска решения**.

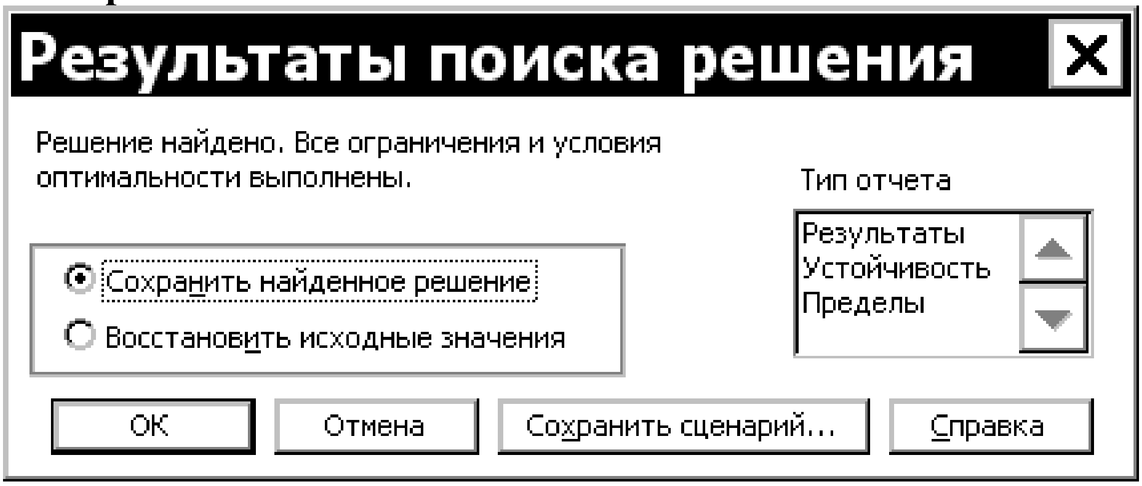


Рис. 2.5. Окно результаты поиска решения.

Выбрав кнопку **Сохранить найденное решение** и нажав кнопку ОК получим решение задачи (см. таблицу 2.8). Из нее следует, что максимальную выручку обеспечивает план, полученный в ячейках C20:C23. Максимальная выручка составляет 59433 тыс. рублей. При этом полностью израсходованы ресурсы по фрезерному, токарному и шлифовальному оборудованию. С другой стороны ресурсы по сверлильному и расточному оборудованию используются только на 76% и 83% соответственно.

ГЛАВА 3. МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

3.1. Графический метод решения задач линейного программирования

Если задача линейного программирования задана в стандартной форме в R^2, то для ее решения используют графический метод, который состоит в следующем.

Строится допустимое множество M, заданное системой ограничений, как пересечение полуплоскостей, определяемых каждым из входящих в эту систему неравенств. Если M – пустое множество, то задача решений не имеет.

Если M – непустое множество, то рассматриваются линии уровня целевой функции $f = c_1x_1 + c_2x_2 + c_0$. Они определяются как прямые вида $c_1x_1 + c_2x_2 = \text{const}$ с общим вектором нормали $n = (c_1, c_2)$, определяющим направление роста функции f. Смещая линии уровня в направлении вектора n, находим первую точку $x^* = (x_1^*, x_2^*)$ пересечения такой линии с множеством M. Тогда $f_{\min} = f(x^*)$ является минимальным значением функции f на M. Аналогично, если $x^* = (x_1^*, x_2^*)$ – последняя точка пересечения линии уровня с множеством M, то $f_{\max} = f(x^*)$ – максимальное значение функции f на M. Если при перемещении линии уровня в направлении n последняя имеет пересечения с M при сколь угодно большом значении константы, то $f_{\max} = +\infty$. Если же, наоборот, линии уровня имеют пересечения с M при сколь угодно большом по модулю отрицательном значении постоянной, то $f_{\min} = -\infty$.

Пример 3.1. Решить графически следующую задачу ЛП

$$z = 3x_1 + x_2 - 10 \to \max$$

$$\begin{cases} 7x_1 + x_2 \ge 29, \\ 3x_1 + 2x_2 \le 25, \\ 4x_1 - x_2 \le 15, \\ x \ge 0. \end{cases}$$

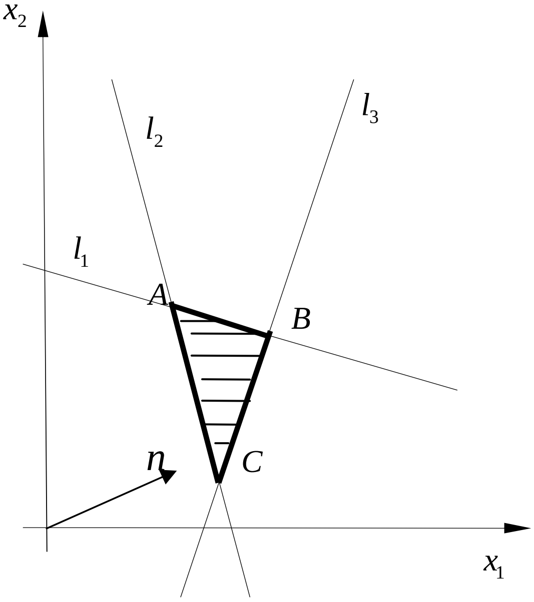


Рис. 3.1

Р е ш е н и е. Построим допустимую область M, т.е. множество на плоскости, определяемое системой

$$\begin{cases} 7x_1 + x_2 \ge 29, \\ 3x_1 + 2x_2 \le 25, \\ 4x_1 - x_2 \le 15, \\ \vec{x} \ge 0. \end{cases}$$

Для этого построим сначала прямые $l_1 : 7x_1 + x_2 = 29$, $l_2 : 3x_1 + 2x_2 = 25$ и $l_3 : 4x_1 - x_2 = 15$ на плоскости (x_1, x_2), а затем найдем их точки пересечения. Точку пересечения прямых l_1, l_2 находим как решение системы уравнений

$$\begin{cases} 7x_1 + x_2 = 29, \\ 3x_1 + 2x_2 = 25, \end{cases}$$

Получаем, что $l_1 \cap l_2 = A(3;8)$. Аналогично находим, что $l_2 \cap l_3 = B(5;5)$, $l_1 \cap l_3 = C(4;1)$.

Каждая из прямых разбивает плоскость на две полуплоскости, а каждое неравенство, входящее в систему ограничений, задает одну из полуплоскостей. Для того чтобы установить, какая из полуплоскостей определяется неравенством, необходимо взять произвольную («пробную») точку, не лежащую на прямой и проверить, удовлетворяет ли она соответствующему неравенству. Если неравенство выполнено, то неравенство определяет полуплоскость, содержащую «пробную» точку, а если неравенство не выполняется, то его решением является другая полуплоскость. Решением системы неравенств будет пересечение соответствующих плоскостей. В нашем случае это треугольник ABC. Построив вектор нормали $n = (3,1)$, убеждаемся, что решением задачи на максимум является точка $B(5;5)$ и $z(B)=3\cdot 5+5-10=10$.

Пример 3.2. Решить графически задачу ЛП

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

$$\begin{cases} x_1 + 2x_2 + 5x_3 - x_4 = 4, \\ x_1 - x_2 - x_3 + 2x_4 = 1, \\ x_i \ge 0, i = 1, \dots, 4. \end{cases}$$

Решение. Формально задача является, вообще говоря, задачей линейного программирования в пространстве R^5. Но ее можно свести к задаче на плоскости R^2, приведя к стандартной форме. Исключая переменные x_1, x_2, получаем задачу

$$z = -x_3 + 5 \to \max$$

$$\begin{cases} x_3 + x_4 \le 2, \\ 2x_3 - x_4 \le 1, \\ x_3 \ge 0, x_4 \ge 0. \end{cases}$$

Строим на плоскости прямые $l_1: x_3 + x_4 = 2$, $l_2: 2x_3 - x_4 = 1$, находим их точку пересечения и в качестве допустимого множества получаем

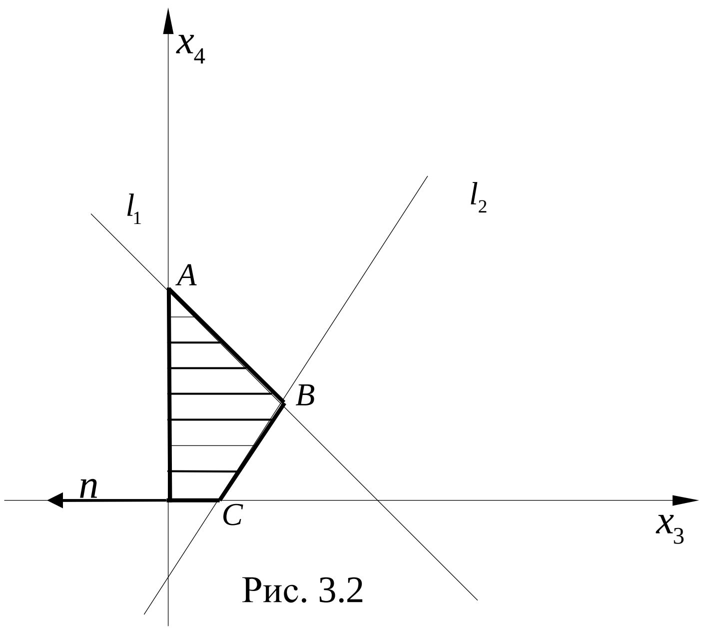


Рис. 3.2

четырехугольник $OABC$ с угловыми точками $O(0,0)$, $A(0,2)$, $B(1,1)$ и $C(\frac{1}{2},0)$.

Вектор нормали имеет координаты $n=(-1,0)$ и мы видим, что оптимальным множеством, на котором достигается максимальное значение функции z является отрезок OA, т.е. множество $X^*=(1-t)A+tB=(1-t)(0,0)+t(0,2)=(0,2t), t\in[0,1]$. Отсюда находим

$$\begin{cases} x_1 = 2 - x_3 - x_4 = 2 - 2t, \\ x_2 = 1 - 2x_3 + x_4 = 1 + 2t, \end{cases}$$

И окончательно получаем, что $z_{\max} = z(X^*) = 5$ при $X^* = (2-2t, 1+2t, 0, 2t), t\in[0,1]$.

Пример 3.3. Решить задачу ЛП

$$z = -x_1 + x_2 + 4 \to \min\ (\max)$$

$$\begin{cases} x_1 - x_2 + x_3 = 3, \\ -2x_1 + x_2 + x_4 = 2, \\ \bar{x} \geq 0. \end{cases}$$

графическим методом.

Решение. Выразим базисные неизвестные x_3, x_4 из ограничений

$$\begin{cases} x_3 = 3 - x_1 + x_2 \geq 0, \\ x_4 = 2 + 2x_1 - x_2 \geq 0, \end{cases}$$

и получаем задачу в стандартной форме

$$z = -x_1 + x_2 + 4 \to \min\ (\max)$$

$$\begin{cases} x_1 - x_2 \leq 3, \\ 2x_1 - x_2 \geq -2, \\ \vec{x} \geq 0. \end{cases}$$

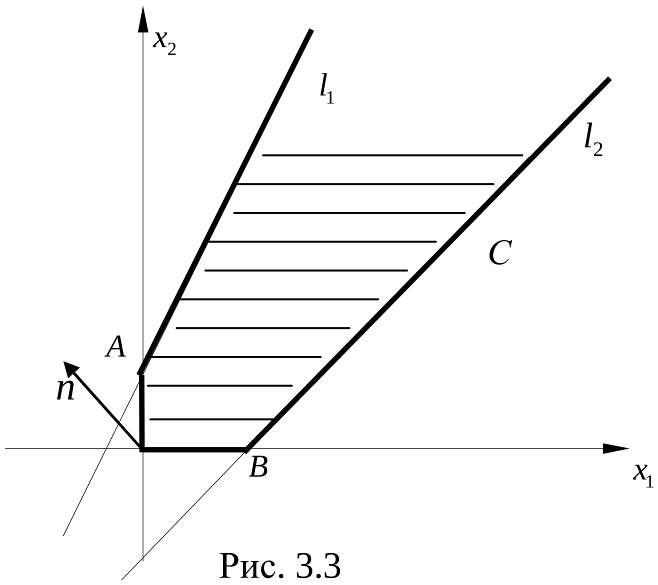


Рис. 3.3

Построив на плоскости прямые $l_1 : x_1 - x_2 = 3,\ \ l_2 : 2x_1 - x_2 = -2$, находим допустимое множество с угловыми точками $O(0,0)$, $A(0,2)$ и $B(3,0)$. Вектор нормали имеет координаты $n = (-1,1)$, и, как легко видеть, луч BC с началом в точке B является линией уровня функции z и определяет множество, на котором z достигает минимального значения. Так как направляющим вектором луча BC является вектор $m = (1,1) \geq 0$, то

$$X_{\min} = B + t \cdot m = (3,0) + t(1,1) = (3+t, t),\ t \geq 0.$$

и $z_{\min} = z(B) = 1$. Кроме этого, так как при движении вдоль вектора n каждая линия уровня пересекается с допустимым множеством, то $z_{\max} = \infty$. Для окончательного ответа находим $x_3 = 0, x_4 = 8 + t.$ Таким образом, $z_{\min} = z(X_{\min}) = 1$ при $X_{\min} = (3+t, t, 0, 8+t), t \geq 0,\ z_{\max} = \infty$.

Задачи для самостоятельного решения

Решить задачи ЛП графическим способом:

23. $z = 4x + 3y \to \max\ (\min)$ при $x \geq 0, y \geq 0$ и $\begin{cases} 5x + 3y \geq 23 \\ 9x - y \leq 35 \\ 4x - 4y \geq -20 \end{cases}$.

24. $z=-3x+2y \to \max \ (\min)$ при $x \ge 0, y \ge 0$ и $\begin{cases} 5x+5y \ge 55 \\ 9x-3y \le 63 \\ 4x-8y \ge -52 \end{cases}$.

25. $z=-4x-3y \to \max \ (\min)$ при $x \ge 0, y \ge 0$ и $\begin{cases} 5x+2y \ge 21 \\ 10x-2y \le 24 \\ 5x-4y \ge -27 \end{cases}$.

26. $z=x-3y \to \max \ (\min)$ при $x \ge 0, y \ge 0$ и $\begin{cases} 5x+5y \ge 55 \\ 9x-2y \le 88 \\ 4x-7y \ge -22 \end{cases}$.

27. $z=-2x+4y \to \max \ (\min)$ при $x \ge 0, y \ge 0$ и $\begin{cases} 4x+2y \ge 28 \\ 7x-y \le 40 \\ 3x-3y \ge -6 \end{cases}$.

28. $z=-3x-3y \to \max \ (\min)$ при $x \ge 0, y \ge 0$ и $\begin{cases} 4x+2y \ge 30 \\ 7x-4y \ge 45 \\ 3x-6y \le -15 \end{cases}$.

29. $z=3x_1-x_2+2x_3-x_4-7 \to \max \ (\min)$ при $\begin{cases} x_1+4x_2-x_3=1, \\ 2x_1-2x_2+x_5=5, \\ x_1+2x_2-x_4=2, \\ x \ge 0. \end{cases}$

30. $z=x_1+x_2+2x_3+x_4+22 \to \max \ (\min)$ при $\begin{cases} x_1+3x_2+2x_3+2x_4=5, \\ 2x_1-2x_3+x_4=4, \\ x \ge 0. \end{cases}$

31. $z=-x_1-x_4+5 \to \max \ (\min)$ при $\begin{cases} x_1+2x_2+x_3=10, \\ x_1+x_2-x_4=2, \\ 2x_1-x_2+x_5=5, \\ x \ge 0. \end{cases}$

32. $z=x_1+x_2+x_3+2x_4+7 \to \max \ (\min)$ при $\begin{cases} 2x_1-x_2+x_3+2x_4=15, \\ x_1-5x_2-x_3+4x_4=3, \\ x \ge 0. \end{cases}$

33. $z = 3x_1 - 4x_2 - x_4 + 11 \to \max\ (\min)$ при $\begin{cases} x_1 + 2x_2 - x_3 - x_4 = 14, \\ 3x_1 - 4x_2 + x_3 - x_4 = 2, \\ x \geq 0. \end{cases}$

34. $z = -104 - 25x_1 + 24x_2 \to \max(\min)$ при $\begin{cases} -10x_1 + 7x_2 + x_3 = 11, \\ 5x_1 + 3x_2 + x_4 = 79, \\ -5x_1 + 10x_2 - x_5 = 25, \\ x \geq 0. \end{cases}$

3.2. Симплекс-метод решения задач линейного программирования

Алгоритм решения задачи симплекс-методом сначала изложим на конкретном примере.

Пример 3.4. Симплекс-методом решить задачу ЛП

$$\begin{cases} z = 3x_1 - x_2 - 2x_3 + 6x_4 - 10 \to \min \\ x_1 + 3x_2 + 7x_3 - x_4 = 6, \\ x_1 - x_2 - x_3 + 3x_4 = 2, \\ x \geq 0. \end{cases}$$

Решение. Начальным этапом решения задачи симплекс-методом является приведение ее к допустимому виду и формирование симплекс-таблицы. Это означает, что задача должна быть приведена к каноническому виду, в системе нетривиальных ограничений должен быть выделен допустимый базис (т.е. базисное решение должно быть неотрицательным, или, что равносильно, все правые части – неотрицательные числа) и из целевой функции должны быть исключены базисные переменные.

В нашем примере система ограничений уже приведена к каноническому виду. Для выделения базиса используем метод Гаусса,

$$\left(\begin{array}{cccc|c} 1 & 3 & 7 & -1 & 6 \\ 1 & -1 & -1 & 3 & 2 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 3 & 7 & -1 & 6 \\ 0 & -4 & -8 & 4 & -4 \end{array}\right) : (-4) \sim$$

$$\sim \left(\begin{array}{cccc|c} 1 & 3 & 7 & -1 & 6 \\ 0 & 1 & 2 & -1 & 1 \end{array}\right) \sim \left(\begin{array}{cccc|c} 1 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & -1 & 1 \end{array}\right) \sim \begin{cases} x_1 + x_3 + 2x_4 = 3, \\ x_2 + 2x_3 - x_4 = 1. \end{cases}$$

Выражая базисные неизвестные, получим $\begin{cases} x_1 = 3 - x_3 - 2x_4, \\ x_2 = 1 - 2x_3 + x_4, \end{cases}$ и, подставляя в функцию z, имеем

$$z = 3(3 - x_3 - 2x_4) - (1 - 2x_3 + x_4) - 2x_3 + 6x_4 - 10 = -3x_3 - x_4 - 2.$$

Таким образом, получаем, что исходная задача эквивалентна следующей

$$\begin{cases} z = -3x_3 - x_4 - 2 \to \min \\ x_1 + x_3 + 2x_4 = 3, \\ x_2 + 2x_3 - x_4 = 1, \\ x \geq 0. \end{cases}$$

Переписываем z в виде $z + 3x_3 + x_4 = -2$ и формируем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	3	1	0	1	2
x_2	1	0	1	2	-1
z	-2	0	0	3	1

Из симплекс-таблицы можно сделать вывод, что базисным решением является $\overline{x} = (3,1,0,0)$ и $z(x) = -2$. Так как в последней строке (строке оценок) есть положительные элементы (коэффициенты при x_3, x_4), то решение можно улучшить с помощью шага симплекс-метода. Выбираем разрешающий элемент для первого шага. В качестве разрешающего возьмем столбец, отвечающий переменной x_3 (как содержащий положительный элемент в оценочной строке), а для выбора разрешающей строки рассмотрим отношение элементов столбца свободных членов (столбца b_i) к положительным элементам разрешающего. Разрешающий элемент выбирается в строке, дающей минимум этого отношения: $\min\left\{\frac{1}{2}, \frac{3}{1}\right\} = \frac{1}{2}$, т.е. выбирается вторая строка. Иначе говорят: мы выводим из базиса переменную x_2 и вводим в базис x_3. Делим вторую строку на 2, получаем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	3	1	0	1	2
x_2	1/2	0	1/2	1	-1/2
z	-2	0	0	3	1

с выделенной разрешающей единицей и делаем шаг симплекс-метода (из первой строки вычитаем вторую и из строки оценок вычитаем разрешающую, умноженную на 3)

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	5/2	1	-1/2	0	5/2
x_2	1/2	0	1/2	1	-1/2
z	-7/2	0	-3/2	0	5/2

Замечание 3.1. Легко видеть, что шаг симплекс-метода во многом сходен с итерацией метода Гаусса. Отличие состоит в том, что разрешающий элемент выбирается не произвольно, а согласно вышеизложенным правилам, которые гарантируют то, что вновь полученная таблица имеет допустимый вид.

Из симплекс-таблицы видно, что базисным решением является $x=\left(\frac{5}{2},0,\frac{1}{2},0\right)$ и $z(x)=-\frac{7}{2}$. Так как в строке оценок снова есть положительный элемент (коэффициент при x_4), то решение можно улучшить. Выбор разрешающего столбца теперь однозначен (отвечает переменной x_4), также как и выбор разрешающей строки – в столбце единственный положительный элемент $\frac{5}{2}$. Итак, выводим из базиса переменную x_1 и вводим в базис x_4. Умножаем первую строку на $\frac{2}{5}$, получаем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	1	2/5	-1/5	0	1
x_2	1/2	0	1/2	1	-1/2
z	-7/2	0	-3/2	0	5/2

и делаем шаг симплекс-метода с отмеченным разрешающим элементом (ко второй строке прибавляем первую, умноженную на $\frac{1}{2}$, а из строки оценок вычитаем первую, умноженную на $\frac{5}{2}$). Получаем

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	1	2/5	-1/5	0	1
x_2	1	1/5	2/5	1	0
z	-6	-1	-1	0	0

В последней строке нет положительных элементов, поэтому оптимальное решение найдено. Таковым является базисное решение $x^* = (0,0,1,1)$ и $z_{\min} = z(x^*) = -6$.

Сформулируем теперь общий алгоритм решения задачи линейного программирования симплекс-методом. Предположим, система ограничений приведена к каноническому виду, в ней выделен допустимый базис (все правые части – неотрицательные числа), из целевой функции исключены базисные переменные и все слагаемые, кроме константы, перенесены в левую часть. Записав все данные в симплекс-таблицу, получим (далее предполагается, что рассматривается задача на максимум, а альтернатива в скобках дана для задачи на минимум)

Если в последней строке нет отрицательных (положительных) оценок, то оптимальное решение достигнуто.

Если в оценочной строке есть хотя бы одна отрицательная (положительная) оценка, то решение может быть улучшено. Для этого выбирается разрешающий столбец (пусть он имеет номер j), содержащий отрицательную (положительную) оценку, а в качестве разрешающего выбирается положительный элемент $a_{lj} > 0$, дающий минимум отношения элемента свободного столбца b_i к a_{ij}:

$$a_{lj} = \min_{a_{ij}>0} \left\{ \frac{b_i}{a_{ij}} \right\}.$$

Если в симплекс-таблице имеется отрицательная (положительная) оценка, а в соответствующем столбце нет положительных элементов, то исходная задача не имеет решения, т.е. $z_{\max} = +\infty$ $(z_{\min} = -\infty)$.

Если оптимальное решение найдено, но при этом у одной (или нескольких) свободной переменной оценка равна 0, то задача имеет альтернативное решение, для получения которого следует сделать шаг симплекс-метода, выбрав разрешающий элемент (по общему правилу) в столбце с нулевой оценкой. При этом множество оптимальных решений совпадает с выпуклой оболочкой всех альтернативных решений.

Заметим, что последнее верно не всегда. Возможна ситуация, когда при поиске альтернативных решений в столбце, содержащем нулевую оценку, все элементы отрицательны (см. пример 8).

Пример 3.5. Решить задачу ЛП симплекс-методом

$$\begin{cases} z = 3x_1 + 3x_2 + 21 \to \max \\ -2x_1 + x_2 + x_3 = 1, \\ x_1 - x_2 + x_4 = 3, \\ x_1 + x_2 + x_5 = 7, \\ \overline{x} \geq 0. \end{cases}$$

Р е ш е н и е. Задача приведена к каноническому виду, допустимый базис уже выделен (переменные x_3, x_4, x_5) и из целевой функции исключены базисные переменные. Поэтому переписываем функцию z в виде $z - 3x_1 - 3x_2 = 21$ и формируем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_3	1	-2	1	1	0	0
x_4	3	1	-1	0	1	0
x_5	7	1	1	0	0	1
z	21	-3	-3	0	0	0

В последней строке есть отрицательные элементы, поэтому решение может быть улучшено. Вводим в базис переменную x_2, а так как $\min\left\{\frac{1}{1}, \frac{7}{1}\right\} = 1$, выводим из базиса переменную x_3.

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	1	-2	1	1	0	0
x_4	4	-1	0	1	1	0
x_5	2	1	0	-1/3	0	1/3
z	24	-9	0	3	0	0

Так как в строке оценок есть единственный отрицательный элемент, а выбор разрешающего элемента однозначен, то выводим из базиса переменную x_5 и вводим в базис переменную x_1. Делим разрешающую строку на 3, и после шага симплекс-метода получаем

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	5	0	1	1/3	0	2/3
x_4	6	0	0	2/3	1	1/3
x_1	2	1	0	-1/3	0	1/3
z	42	0	0	0	0	3

В последней строке нет отрицательных элементов, поэтому оптимальным решением является базисное решение $X_1 = (2,5,0,6,0)$ и $z_{\max} = z(X_1) = 42$.

С другой стороны, в столбце свободной переменной x_3 в строке оценок есть нулевая оценка, а, значит, имеется альтернативное решение. Для того чтобы его найти, выбираем по общему правилу разрешающий элемент в этом столбце: так как $\min\left\{\frac{5}{1/3}, \frac{6}{2/3}\right\} = \frac{6}{2/3}$, умножаем вторую строку на $3/2$, получаем

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	5	0	1	1/3	0	2/3
x_4	9	0	0	1	3/2	1/2
x_1	2	1	0	-1/3	0	1/3
z	42	0	0	0	0	3

и делаем шаг симплекс—метода (вводим в базис x_3 и выводим из базиса переменную x_4)

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	2	0	1	0	1/2	1/2
x_3	9	0	0	1	3/2	1/2
x_1	5	1	0	0	1/2	1/2
z	42	0	0	0	0	3

и альтернативным оптимальным решением является $X_2 = (5, 2, 9, 0, 0)$. Заметим, что других альтернатив нет, так как, вводя в базис переменную x_4, мы вновь получаем альтернативное решение X_1. Итак, оптимальным множеством исходной задачи является отрезок, соединяющий точки X_1 и X_2:

$$X^* = (1-t)X_1 + tX_2 = (2+3t, 5-3t, 9t, 6-6t, 0), t \in [0,1].$$

Получаем, что $z_{\max} = 42$ при $X^* = (2+3t, 5-3t, 9t, 6-6t, 0)$, $t \in [0,1]$.

Пример 3.6. Симплекс-методом решить задачу ЛП

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

Решение. Переписываем функцию z в виде $z + x_1 - x_2 = 4$ и составляем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_3	3	1	-1	1	0
x_4	2	-2	1	0	1
z	4	1	-1	0	0

В последней строке есть положительный элемент, поэтому решение может быть улучшено. Вводим в базис переменную x_1 и выводим из базиса переменную x_3:

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_1	3	1	-1	1	0
x_4	8	0	-1	2	1
z	1	0	0	-1	0

В последней строке нет положительных оценок, поэтому оптимальным решением является базисное решение $X_1 = (3, 0, 0, 8)$ и $z_{\min} = z(X_1) = 1$. Замечаем, что в столбце свободной переменной x_2 в строке оценок есть нулевая оценка, но все элементы этого столбца отрицательные. С одной стороны, это указывает на то, что оптимальное множество состоит из бесконечного множества точек, а с другой стороны, у этого множества больше нет угловых точек (например, если это луч; см. пример 3.3).

Для записи общего решения находим другую (не базисную) оптимальную точку, т.е. выражаем, используя заключительную симплекс-таблицу, базисные переменные через свободные

$$\begin{cases} x_1 = 3 + x_2 - x_3, \\ x_4 = 8 + x_2 - 2x_3, \end{cases}$$

и полагаем, например, $x_2 = 1, x_3 = 0 \Rightarrow X_2 = (4,1,0,9)$. Получаем, что решением задачи является луч X_1X_2 с началом в точке X_1 (для геометрической интерпретации задачи см. пример 3.3) $X_{\min} = tX_2 + (1-t)X_3 = t(4,1,0,9) + (1-t)(3,0,0,8) = (3+t, t, 0, 8+t),\ t \ge 0$ Таким образом, $z_{\min} = z(X_{\min}) = 1$ при $X_{\min} = (3+t, t, 0, 8+t)$, $t \ge 0$, $z_{\max} = \infty$.

Задачи для самостоятельного решения

Решить задачи ЛП симплексным методом

35. $z = 3x_3 - 4x_4 + 2x_5 + 9 \to \max$ при $x \ge 0$ и $\begin{cases} 3x_1 + x_2 + x_3 = 6, \\ x_1 + x_2 + x_4 = 4, \\ -x_1 + x_3 + 2x_5 = 4. \end{cases}$

36. $z = x_1 - 4x_2 - 2x_4 + x_5 - 11 \to \min$ при $x \ge 0$ и $\begin{cases} x_1 - 5x_2 + x_3 = 5, \\ x_1 - x_2 - x_4 = -4, \\ x_1 + x_2 + x_5 = 8. \end{cases}$

37. $z = 2x_1 - 2x_2 - x_3 - x_5 - 13 \to \min$ при $x \ge 0$ и $\begin{cases} -3x_1 + 2x_2 + x_3 = 4, \\ -5x_1 - 3x_2 - x_4 = -25, \\ 2x_1 - 3x_2 + x_5 = 6. \end{cases}$

38. $z = 3x_2 + x_4 + 8 \to \max$ при $x \geq 0$ и $\begin{cases} 2x_1 - 2x_2 + x_3 = 1, \\ 3x_1 - 2x_2 \leq 4, \\ -x_1 + x_2 + x_4 = 1. \end{cases}$

39. $z = -x_1 - x_3 - 10 \to \min$ при $x \geq 0$ и $\begin{cases} 2x_1 - 2x_3 + 3x_4 \leq 12, \\ x_1 + x_2 + 2x_3 - x_4 = 1. \end{cases}$

40. $z = 2x_2 - x_4 + 12 \to \max$ при $x \geq 0$ и $\begin{cases} 4x_1 + 3x_2 \leq 12, \\ 4x_1 + x_2 + x_3 = 8, \\ 4x_1 - x_2 + x_4 = 8. \end{cases}$

3.3. Метод искусственного базиса

Если в исходной системе ограничений не выделен допустимый базис, как того требует алгоритм симплекс—метода, для его нахождения можно решить вспомогательную задачу, которая ставится следующим образом.

Пусть исходная система нетривиальных ограничений задана в общем виде

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

где $b_i \geq 0, i = 1, \ldots, m.$ Выполнения последнего условия всегда можно добиться, умножив уравнения на -1. Введем в систему новые (искусственные) переменные $y_1, y_2, \ldots, y_m$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + y_m = b_m, \end{cases}$$

так что новая система имеет допустимое базисное решение $(0, 0, \ldots, 0; b_1, b_2, \ldots, b_m) \in \mathrm{R}^{n+m}$. Рассмотрим вспомогательную целевую функцию $F\left(x_1, x_2, \ldots, x_n; y_1, y_2, \ldots, y_m\right) = y_1 + y_2 + \ldots + y_m$ и решим симплекс-методом задачу

$$\begin{cases} F = y_1 + y_2 + \ldots + y_m \to \min \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + y_m = b_m. \end{cases}$$

Если последняя задача имеет решение, то возможны два случая:

- Если $\min F > 0$, то система ограничений не имеет допустимого базиса и задача не имеет решений.
- Если $\min F = 0$, то система ограничений имеет неотрицательное базисное решение. Чтобы получить систему ограничений, эквивалентную исходной, но с выделенным допустимым базисом, необходимо, чтобы в заключительной симплекс—таблице все искусственные переменные были свободными.

Пример 3.7. Рассмотрим задачу из примера 3.4

$$\begin{cases} z = 3x_1 - x_2 - 2x_3 + 6x_4 - 10 \to \min \\ x_1 + 3x_2 + 7x_3 - x_4 = 6, \\ x_1 - x_2 - x_3 + 3x_4 = 2, \\ \overline{x} \geq 0. \end{cases}$$

и выделим допустимый базис с помощью вспомогательной задачи. Итак, рассмотрим задачу

$$\begin{cases} F = y_1 + y_2 \to \min \\ x_1 + 3x_2 + 7x_3 - x_4 + y_1 = 6, \\ x_1 - x_2 - x_3 + 3x_4 + y_2 = 2. \end{cases}$$

В системе выделен допустимый (искусственный) базис y_1, y_2, поэтому выражаем из уравнений базисные переменные

$$\begin{cases} y_1 = 6 - x_1 - 3x_2 - 7x_3 + x_4, \\ y_2 = 2 - x_1 + x_2 + x_3 - 3x_4 \end{cases}$$

и подставляем в выражение для $F: F = y_1 + y_2 = 8 - 2x_1 - 2x_2 - 6x_3 - 2x_4$. Переписываем это равенство в виде $F + 2x_1 + 2x_2 + 6x_3 + 2x_4 = 8$ и формируем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4	y_1	y_2
y_1	6	1	3	7	-1	1	0
y_2	2	1	-1	-1	3	0	1
F	8	2	2	6	2	0	0

Выводим из базиса переменную y_2 и вводим в базис x_1. Получим

Б.П.	с.ч.	x_1	x_2	x_3	x_4	y_1	y_2
y_1	4	0	4	8	-4	1	-1
x_1	2	1	-1	-1	3	0	1
F	4	0	4	8	-4	0	-2

Далее, выводим из базиса переменную y_1 и вводим в базис x_2. Для этого делим первую строку на 4, получаем таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4	y_1	y_2
y_1	1	0	1	2	-1	1/4	-1/4
x_1	2	1	-1	-1	3	0	1
F	4	0	4	8	-4	0	-2

и делаем шаг симплекс-метода

Б.П.	с.ч.	x_1	x_2	x_3	x_4	y_1	y_2
x_2	1	0	1	2	-1	1/4	-1/4
x_1	3	1	0	1	2	1/4	3/4
F	0	0	0	0	0	-1	-1

Данная симплекс-таблица – заключительная, искусственные переменные y_1, y_2 стали свободными, и $F_{\min} = 0$. Опуская последнюю строку и столбцы, получаем таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4
x_2	1	0	1	2	-1
x_1	3	1	0	1	2

т.е. получаем систему ограничений

$$\begin{cases} x_2 + 2x_3 - x_4 = 1, \\ x_1 + x_3 + 2x_4 = 3, \end{cases}$$

с выделенным допустимым базисным решением, которая равносильна исходной системе ограничений и совпадает с системой, полученной (другим способом) в примере 3.4.

Пример 3.8. Симплекс-методом решить задачу

$$\begin{cases} z = x_1 + 3x_2 + 8 \to \min \\ -x_1 + x_2 + x_3 = 1, \\ x_1 + 3x_2 - x_4 = 19, \\ 3x_1 + x_2 + x_5 = 33, \\ \overline{x} \geq 0 \end{cases}$$

симплекс-методом.

Решение. В данной задаче допустимый базис выделен лишь частично $(\text{переменные } x_3, x_5)$, поэтому ограничимся лишь введением одной искусственной переменной y и решим следующую задачу

$$\begin{cases} F = y \to \min, \\ -x_1 + x_2 + x_3 = 1, \\ x_1 + 3x_2 - x_4 + y = 19, \\ 3x_1 + x_2 + x_5 = 33 \end{cases}$$

Так как $F = y = 19 - x_1 - 3x_2 + x_4$, то $F + x_1 + 3x_2 - x_4 = 19$. Аналогично, $z = x_1 + 3x_2 + 10$, и $z - x_1 - 3x_2 = 10$. Решая задачу для функции F, внесем в таблицу строку для функции z, одновременно преобразуя и ее. Получим

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5	y
x_3	1	-1	1	1	0	0	0
y	19	1	3	0	-1	0	1
x_5	33	3	1	0	0	1	0
z	8	-1	-3	0	0	0	0
F	19	1	3	0	-1	0	0

Сделаем шаг симплекс—метода с выделенным разрешающим элементом

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5	y
x_2	1	-1	1	1	0	0	0
y	16	4	0	-3	-1	0	1
x_5	32	4	0	-1	0	1	0
z	11	-4	0	3	0	0	0
F	16	4	0	-3	-1	0	0

Выводим из базиса переменную y и вводим в базис x_1.

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5	y
x_2	5	0	1	1/4	-1/4	0	1/4
x_1	4	1	0	-3/4	-1/4	0	1/4
x_5	16	0	0	2	1	1	-1
z	27	0	0	0	-1	0	1
F	0	0	0	0	0	0	-1

Вспомогательная задача $F \to \min$ решена и искусственная переменная y – свободная. Поэтому опускаем последнюю строку и столбец, и получаем симплекс-таблицу для исходной задачи с выделенным допустимым базисом:

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	5	0	1	1/4	-1/4	0
x_1	4	1	0	-3/4	-1/4	0
x_5	16	0	0	2	1	1
z	27	0	0	0	-1	0

Более того, данная симплекс-таблица – заключительная, поэтому решением задачи является $X_1 = (4,5,0,0,16)$, а так как имеется свободный столбец x_3 с нулевой оценкой, то у задачи есть альтернативное решение. Выводим из базиса переменную x_5 и вводим в базис x_3:

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_2	3	0	1	0	-3/8	-1/8
x_1	10	1	0	0	1/8	3/8
x_5	8	0	0	1	1/2	1/2
z	27	0	0	0	-1	0

и получаем альтернативное решение $X_2 = (10,3,8,0,0)$. Поэтому общее решение задачи

$$X^* = (1-t)X_1 + tX_2 = (1-t)(4,5,0,0,16) + t(10,3,8,0,0) =$$
$$= (4+6t, 5-2t, 8t, 0, 16-16t), t \in [0,1]$$

и $z_{\min} = z\left(X^*\right) = 27$.

Пример 3.9. Симплекс-методом решить задачу ЛП

$$\begin{cases} z = x_1 + 2x_2 + 9 \to \max \\ x_1 + x_2 - x_3 = 2, \\ x_1 + 4x_2 + x_4 = 1, \\ -x_1 + x_2 - x_5 = 3, \\ x \geq 0. \end{cases}$$

Решение. Вводим две искусственные переменные y_1, y_2 и решим следующую задачу

$$\begin{cases} F = y_1 + y_2 \to \min \\ x_1 + x_2 - x_3 + y_1 = 2, \\ x_1 + 4x_2 + x_4 = 1, \\ -x_1 + x_2 - x_5 + y_2 = 3, \\ x \geq 0, y \geq 0. \end{cases}$$

Из системы ограничений преобразуем $F = y_1 + y_2 = 5 - 2x_2 + x_3 + x_5$, т.е. $F + 2x_2 - x_3 - x_5 = 5$. Аналогично, $z - x_1 - 2x_2 = 9$ и мы решаем задачу для функции F, включив при этом в таблицу строку для функции z:

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5	y_1	y_2
y_1	2	1	1	-1	0	0	1	0
x_5	1	1	4	0	1	0	0	0
y_2	3	-1	1	0	0	-1	0	1
z	9	-1	-2	0	0	0	0	0
F	5	0	2	-1	0	-1	0	0

В строке оценок есть единственный положительный элемент, поэтому вводим в базис x_2 и выводим из базиса x_5

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5	y_1	y_2
y_1	7/4	3/4	0	-1	-1/4	0	1	0
x_2	1/4	1/4	1	0	1/4	0	0	0
y_2	11/4	-5/4	0	0	-1/4	-1	0	1
z	19/2	-1/2	0	0	1/2	0	0	0
F	9/2	-1/2	0	-1	-1/2	-1	0	0

Все элементы строки оценок для задачи $F \to \min$ отрицательны, поэтому симплекс—таблица – заключительная, но, так как $\min F = 9/2 > 0$, то исходная задача не имеет ни одного допустимого базиса и решений не имеет.

Задачи для самостоятельного решения

Решить задачи линейного программирования, используя метод искусственного базиса

41. $z = -x_1 + 3x_2 + 5x_3 + x_4 - 7 \to \min$ при $\begin{cases} 2x_1 + 11x_2 + 12x_3 + 3x_4 = 14, \\ 9x_2 + 12x_3 + 3x_4 = 12, \\ \bar{x} \geq 0. \end{cases}$

42. $z = 6x_2 + x_3 - x_4 + 13 \to \max$ при $\begin{cases} 3x_1 - x_2 + x_3 + 6x_4 + x_5 = 6, \\ x_1 + 5x_3 + x_4 - 7x_5 = 6, \\ x_1 + 2x_2 + 3x_3 + x_4 + x_5 = 6, \\ \bar{x} \geq 0. \end{cases}$

43. $z = 6x_1 - x_3 + x_4 + 2x_5 - 8 \to \max$ при $\begin{cases} 4x_1 + x_2 + x_3 + 2x_4 + x_5 = 8, \\ 2x_1 - x_2 + x_4 = 2, \\ x_1 + x_2 + x_5 = 2, \\ \bar{x} \geq 0. \end{cases}$

44. $z = -5x_1 - 3x_2 - 2x_3 + x_4 - x_5 - 8 \to \min$ при $\begin{cases} 3x_1 + 4x_2 + x_3 = 12, \\ 3x_1 + 2x_2 + x_3 + x_4 + x_5 = 16, \\ x_1 - 3x_2 + x_5 = 3, \\ \bar{x} \geq 0. \end{cases}$

45. $z = 7x_1 + 2x_3 - x_4 + x_5 + 24 \to \max$ при $\begin{cases} -x_1 + x_2 + x_3 = 2, \\ 3x_1 - x_2 + x_4 = 3, \\ 5x_1 + 2x_2 + x_3 + x_4 + x_5 = 11, \\ \bar{x} \geq 0. \end{cases}$

46. $z = 7x_2 + x_3 - x_4 - x_5 + 17 \to \min$ при $\begin{cases} -x_1 + 2x_2 + x_3 = 2, \\ 9x_1 + x_2 + x_3 + x_4 + 2x_5 = 26, \\ 3x_1 - 2x_2 + x_5 = 3, \\ \bar{x} \geq 0. \end{cases}$

47. $z = x_1 + x_3 + x_4 + x_5 + 29 \to \max$ при $\begin{cases} x_1 - x_2 + x_3 = 1, \\ 3x_1 + x_2 + x_5 = 7, \\ 5x_1 + 2x_2 + 2x_3 + x_4 + 3x_5 = 17, \\ \bar{x} \geq 0. \end{cases}$

48. Суточная потребность человека в витаминах и минеральных веществах удовлетворяется за счет потребления двух продуктов – киви и гранатов. Содержание питательных веществ в продуктах (мг/100 г), суточные нормы их потребления (мг) и цена продуктов (руб. за 100 г) задаются таблицей

	Витамины	Мин. вещества	Цена
Киви	2	1	10

Гранаты	1	3	20
Норма	24	27	

При этом гранат должно быть не более 800 г. Составить математическую модель задачи и найти суточный рацион минимальной стоимости.

ГЛАВА 4. ВЗАИМНО ДВОЙСТВЕННЫЕ ЗАДАЧИ

4.1. Основные определения и теоремы

Рассмотрим пару двойственных задач ЛП

$$\begin{cases} AX \le B, \\ \tilde{X} \ge 0, \\ z = C^t X + c_0 \to \max \end{cases} \quad \text{и} \quad \begin{cases} A^t Y \ge C, \\ \tilde{Y} \ge 0, \\ T = B^t Y + c_0 \to \min \end{cases}$$

где $A = \left(a_{ij}\right)$ – $m \times n$ матрица, $X = \left(x_1, x_2, \dots, x_n\right)^t$, $Y = \left(y_1, y_2, \dots, y_m\right)^t$, $B = \left(b_1, b_2, \dots, b_m\right)^t$, $C = \left(c_1, c_2, \dots, c_n\right)^t$ – векторы-столбцы соответствующей размерности, а вектора $\tilde{X} = \left(x_{i_1}, x_{i_2}, \dots, x_{i_k}\right)^t$, $\tilde{Y} = \left(y_{j_1}, y_{j_2}, \dots, y_{j_l}\right)^t$ с индексами $1 \le i_1 \le i_2 \le \ldots \le i_k \le n$, $1 \le j_1 \le j_2 \le \ldots \le j_k \le m$ – части векторов X, Y (то есть тривиальные ограничения налагаются лишь на часть координат векторов X, Y). Нетривиальные ограничения в обеих задачах могут быть как типа неравенств (со знаком «$\le$» или «$\ge$»), так и типа уравнений. Чтобы понять связь между этими задачами, построим расширенные матрицы

$$\tilde{A} = \left[\begin{array}{cccc|c|c} a_{11} & a_{12} & \dots & a_{1n} & \le & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & = & b_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & \ge & b_m \\ \hline \ge & \sim & \dots & \ge & & \\ \hline c_1 & c_2 & \dots & c_n & & c_0 \end{array}\right]_{\to \max}$$

и

$$\tilde{A}' = \left[\begin{array}{cccc|c|c} a_{11} & a_{21} & \dots & a_{m1} & \geq & c_1 \\ a_{12} & a_{22} & \dots & a_{m2} & = & c_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} & \geq & c_n \\ \hline \geq & \sim & \dots & \geq & & \\ \hline b_1 & b_2 & \dots & b_m & & c_0 \end{array}\right]_{\to \min}$$

обеих задач. Таким образом,

1. Матрица нетривиальных ограничений двойственной задачи получается из соответствующей матрицы исходной задачи транспонированием.

2. Правые части нетривиальных ограничений двойственной задачи являются коэффициентами целевой функции исходной задачи, и, наоборот, коэффициенты целевой функции двойственной задачи совпадают с правыми частями ограничений исходной задачи.

3. Если исходная задача является задачей на максимум (на минимум), то двойственная задача будет задачей на минимум (на максимум), и строка тривиальных ограничений переходит в столбец нетривиальных ограничений без изменения знака неравенства с «$\leq$» на «$\geq$» и наоборот (соответственно, с изменением знака). Если на переменную в исходной задаче тривиальное ограничение отсутствует, то соответствующее ограничение в двойственной задаче будет типа уравнения; иными словами, «~» переходит в «=». Столбец нетривиальных ограничений переходит в строку тривиальных ограничений с изменением знака неравенства с «$\leq$» на «$\geq$» и наоборот (соответственно, без изменения знака), а «=» переходит в «~».

Пример 4.1. Для задачи

$$\begin{cases} z = 4x_1 - 5x_2 + 8x_3 - 10x_4 + x_5 + 14 \to \min \\ x_1 - 2x_2 + 7x_3 - x_5 \leq 37, \\ -4x_1 - 7x_2 + 4x_4 - 9x_5 \geq -28, \\ 2x_1 + 6x_3 - 4x_4 + x_5 = 48, \\ x_1 \geq 1, x_2 \geq 0, x_3 \leq 0, x_4 \geq 0. \end{cases}$$

составить двойственную.

Решение. Заметим сразу, что $x_1 \geq 1$ будем считать нетривиальным ограничением, поэтому задачу можно переписать в виде

$$\begin{cases} z = 4x_1 - 5x_2 + 8x_3 - 10x_4 + x_5 + 14 \to \min \\ x_1 - 2x_2 + 7x_3 - x_5 \le 37, \\ -4x_1 - 7x_2 + 4x_4 - 9x_5 \ge -28, \\ 2x_1 + 6x_3 - 4x_4 + x_5 = 48, \\ x_1 \ge 1, \\ x_2 \ge 0, x_3 \le 0, x_4 \ge 0. \end{cases}$$

Выпишем расширенную матрицу полученной задачи

$$\tilde{A} = \left[\begin{array}{ccccc|c|c} 1 & -2 & 7 & 0 & -1 & \le & 37 \\ -4 & -7 & 0 & 4 & -9 & \ge & -28 \\ 2 & 0 & 6 & -4 & 1 & = & 48 \\ 1 & 0 & 0 & 0 & 0 & \ge & 1 \\ \hline \sim & \ge & \le & \ge & \sim & & \\ \hline 4 & -5 & 8 & -10 & 1 & & 14 \end{array}\right]_{\to \min}$$

Составляем расширенную матрицу двойственной задачи согласно общим правилам и получаем

$$\tilde{A}' = \left[\begin{array}{cccc|c|c} 1 & -4 & 2 & 1 & = & 4 \\ -2 & -7 & 0 & 0 & \le & -5 \\ 7 & 0 & 6 & 0 & \ge & 8 \\ 0 & 4 & -4 & 0 & \le & -10 \\ -1 & -9 & 1 & 0 & = & 1 \\ \hline \le & \ge & \sim & \ge & & \\ \hline 37 & -28 & 48 & 1 & & 14 \end{array}\right]_{\to \max} .$$

Двойственная задача имеет вид

$$
\begin{cases}
T = 37y_1 - 28y_2 + 48y_3 + y_4 + 14 \to \max \\
y_1 - 4y_2 + 2y_3 + y_4 = 4, \\
-2y_1 - 7y_2 \le -5,\ 7y_1 + 6y_3 \ge 8, \\
4y_2 - 4y_3 \le -10, \\
-y_1 - 9y_2 - y_3 = 1, \\
y_1 \le 0, y_2 \ge 0, y_4 \ge 0.
\end{cases}
$$

Основная связь между двойственными задачами изложена в следующих утверждениях.

Теорема 4.1 (основное неравенство для двойственных задач). *Для всех допустимых решений* X, Y *пары двойственных задач имеет место неравенство* $z(X) \le T(Y)$.

Теорема 4.2 (первая теорема двойственности). *Если исходная задача имеет оптимальное решение, то и двойственная ей имеет оптимальное решение. При этом оптимальные значения обеих целевых функций равны, то есть* $z_{\max} = T_{\min}$.

Чтобы решить задачу ЛП, иногда проще решить двойственную задачу, а затем найти решение исходной задачи.

Задачи для самостоятельного решения

Для задач ЛП построить двойственные задачи

49. $$
\begin{cases}
z = 2x_1 - 7x_3 + 6x_4 - 40 \to \max \\
x_1 - 3x_2 + 4x_3 = 7, \\
-3x_1 - 8x_2 + 2x_4 \le 97, \\
17x_1 + 3x_2 - 5x_4 \le 15, \\
x_1 \ge 1, \\
x_2 \le 0, x_3 \le 0.
\end{cases}
$$

50. $$
\begin{cases}
z = -x_1 + 3x_2 + 12x_3 - x_4 + 5 \to \min \\
-7x_1 + 2x_2 - 6x_4 \ge -25, \\
4x_1 - 7x_3 + 13x_4 - 8x_5 = 16, \\
-7x_1 + 4x_3 + 3x_4 - 46x_5 \ge 15, \\
x_5 \ge 1, \\
x_1 \le 0, x_2 \le 0, x_4 \ge 0.
\end{cases}
$$

51. $\begin{cases} z=-4x_1-3x_3+12x_4-x_5-7\to\max \\ -5x_1+3x_2+8x_5\ge 5, \\ 2x_1-6x_2+7x_3-4x_5\le 70, \\ 3x_1+2x_2-4x_3+6x_4\ge 12, \\ x_2\le 2, x_4\ge 7, \\ x_1\ge 0, x_3\ge 0. \end{cases}$

52. $\begin{cases} z=23x_1+4x_2-7x_3+4x_5-28\to\min \\ 4x_1-3x_3+14x_5\ge 34, \\ -x_1-x_2-x_3+18x_5\le 5, \\ x_4\ge 3, \\ x_1\ge 0, x_2\ge 0, x_5\le 0. \end{cases}$

4.2. Решение двойственных задач с помощью теоремы равновесия.

Сформулируем теперь теорему равновесия (вторую теорему двойственности), которая позволяет не только установить связь между оптимальными значениями целевых функций, но и между точками, в которых эти значения достигаются.

Теорема 4.3 (теорема равновесия). О*птимальные решения* $X^*=\left(x_1^*,x_2^*,\ldots,x_n^*\right)$ *и* $Y^*=\left(y_1^*,y_2^*,\ldots,y_m^*\right)$ *пары двойственных задач связаны между собой равенствами*

$$\begin{cases} \left(\sum\limits_{i=1}^{m} a_{ik}y_i^*-c_k\right)\cdot x_k^*=0,\ k=1,\ldots,n, \\ \left(\sum\limits_{k=1}^{n} a_{ik}x_k^*-b_i\right)\cdot y_i^*=0,\ i=1,\ldots,m. \end{cases}$$

Пример 4.2. Решить задачу

$$\begin{cases} z = 22x_1 + 91x_2 - 37x_3 + 19 \to \min \\ -10x_1 + 7x_2 + 3x_3 \ge 1, \\ 8x_1 + 2x_2 - 10x_3 \ge 22, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

с помощью теоремы равновесия.

Решение. Составим сначала задачу, двойственную данной. Выпишем расширенную матрицу

$$\tilde{A} = \left[\begin{array}{ccc|c|c} -10 & 7 & 3 & \ge & 1 \\ 8 & 2 & -10 & \ge & 22 \\ \hline \ge & \ge & \ge & & \\ \hline 22 & 91 & -37 & & 19 \end{array}\right]_{\to\min}$$

и построим расширенную матрицу двойственной задачи

$$\tilde{A}' = \left[\begin{array}{cc|c|c} -10 & 8 & \le & 22 \\ 7 & 2 & \le & 91 \\ 3 & -10 & \le & -37 \\ \hline \ge & \ge & & \\ \hline 1 & 22 & & 19 \end{array}\right]_{\to\max}.$$

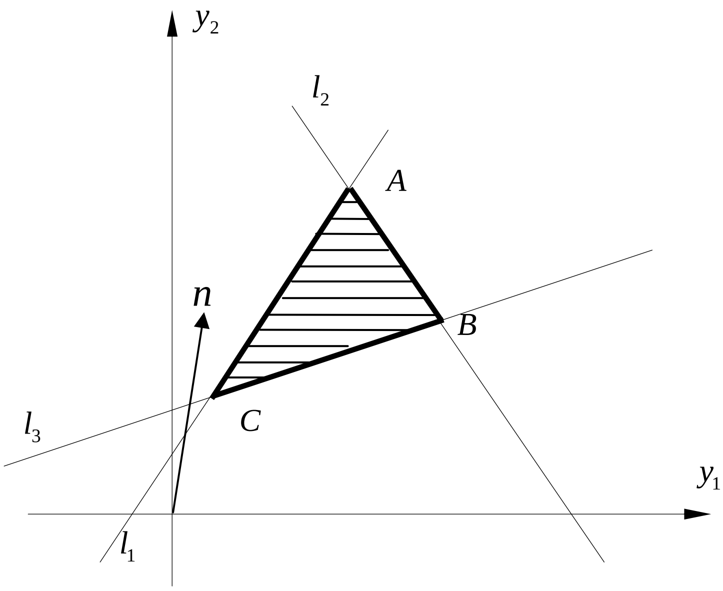
Двойственная задача имеет вид

$$\begin{cases} T = y_1 + 22y_2 + 19 \to \max \\ -10y_1 + 8y_2 \le 22, \\ 7y_1 + 2y_2 \le 91, \\ 3y_1 - 10y_2 \le -37, \\ y_1 \ge 0, y_2 \ge 0. \end{cases}$$

Решим ее графическим способом. Строим на плоскости (y_1, y_2) прямые $l_1: -10y_1 + 8y_2 = 22$, $l_2: 7y_1 + 2y_2 = 91$, $l_3: 3y_1 - 10y_2 = -37$ и убеждаемся, что нетривиальные ограничения определяют треугольник ABC, угловые точки $A = l_1 \cap l_2$, $B = l_2 \cap l_3$ и $C = l_1 \cap l_3$ которого находим из систем уравнений

$$\begin{cases}-10y_1+8y_2=22,\\ 7y_1+2y_2=91,\end{cases}\quad \begin{cases}7y_1+2y_2=91,\\ 3y_1-10y_2=-37,\end{cases}\quad \text{и}\quad \begin{cases}-10y_1+8y_2=22,\\ 3y_1-10y_2=-37,\end{cases}$$

Отсюда получаем, что $A(9,14)$, $B(11,7)$ и $C(1,4)$



Вектор нормали n имеет координаты $\overline{n}=(1,22)$. Поэтому, очевидно, максимальное значение функции $T(Y)$ достигается в точке $A(9,14)$ и

$$Y^*=(9,14),\ T_{\max}=T(9,14)=9+22\cdot 14+19=336.$$

Запишем теперь теорему равновесия для данной пары двойственных задач

$$\begin{cases}\left(-10y_1^*+8y_2^*-22\right)\cdot x_1^*=0,\\ \left(7y_1^*+2y_2^*-91\right)\cdot x_2^*=0,\\ \left(3y_1^*-10y_2^*+37\right)\cdot x_3^*=0,\\ \left(-10x_1^*+7x_2^*+3x_3^*-1\right)\cdot y_1^*=0,\\ \left(8x_1^*+2x_2^*-10x_3^*-22\right)\cdot y_2^*=0.\end{cases}$$

С учетом того, что $Y^*=(9,14)$ (то есть $y_1^*\neq 0, y_2^*\neq 0$) и $A=l_1\cap l_2$, $A\notin l_3$ (то есть $-10y_1^*+8y_2^*-22=0$, $7y_1^*+2y_2^*-91=0$, $3y_1^*-10y_2^*+37\neq 0$, в чем можно убедиться непосредственной подстановкой), выполнение первых двух уравнений очевидно, третье сводится к условию $x_3^* = 0$ и система сводится к виду

$$\begin{cases} x_3^* = 0, \\ -10x_1^* + 7x_2^* + 3x_3^* - 1 = 0, \\ 8x_1^* + 2x_2^* - 10x_3^* - 22 = 0. \end{cases} \quad \text{или} \quad \begin{cases} x_3^* = 0, \\ -10x_1^* + 7x_2^* - 1 = 0, \\ 8x_1^* + 2x_2^* - 22 = 0, \end{cases}$$

решениями которой является точка $X^* = (2, 3, 0)$. Заметим, что $z\left(X^*\right) = 22 \cdot 2 + 91 \cdot 3 - 37 \cdot 0 + 19 = 336 = T\left(Y^*\right)$, как и должно быть по первой теореме двойственности.

Задачи для самостоятельного решения

Для следующих задач линейного программирования
а) построить задачу, двойственную данной;
б) решить двойственную задачу графическим методом;
в) найти решение исходной задачи с помощью теоремы равновесия.

53. $z = x_1 + 154x_2 - 21x_3 \to \min$ при $\bar{x} \ge 0$ и $\begin{cases} -13x_1 + 8x_2 + 5x_3 \ge -18, \\ 7x_1 + 6x_2 - 13x_3 \ge 20. \end{cases}$

54. $z = x_1 + 4x_2 - 2x_3 \to \min$ при $\bar{x} \ge 0$ и $\begin{cases} -2x_1 + x_2 + x_3 \ge -4, \\ 7x_1 + 4x_2 - 11x_3 \ge 29. \end{cases}$

55. $z = 20x_1 + 108x_2 - 51x_3 \to \min$ при $\bar{x} \ge 0$ и $\begin{cases} -8x_1 + 3x_2 + 5x_3 \ge -13, \\ -x_1 + x_2 - 2x_3 \ge 3. \end{cases}$

56. $z = -10x_1 + 44x_2 + x_3 \to \min$ при $\bar{x} \ge 0$ и $\begin{cases} -5x_1 + x_2 + 4x_3 \ge -4, \\ 5x_1 + 6x_2 - 11x_3 \ge 11. \end{cases}$

57. $z = 3x_1 + 58x_2 - 13x_3 \to \min$ при $\bar{x} \ge 0$ и $\begin{cases} -9x_1 + 2x_2 + 7x_3 \ge -25, \\ 6x_1 + 4x_2 - 10x_3 \ge 22. \end{cases}$

58. $z=-10x_1+56x_2-3x_3\to\min$ при $\overline{x}\geq 0$ и $\begin{cases}-12x_1+7x_2+5x_3\geq -22,\\ x_1+3x_2-4x_3\geq 9.\end{cases}$

4.3. Решение двойственных задач с помощью симплекс-метода

Решение пары двойственных задач может быть основано на следующем правиле.

Теорема 4.4. *Если для одной из задач оптимальное решение найдено симплекс-методом, то в исходной задаче можно выделить квадратную матрицу P, образованную столбцами, соответствующими базисным переменным оптимального решения. Тогда оптимальное решение двойственной задачи находится по формуле*

$$Y^*=c_{\text{баз}}\cdot P^{-1},$$

где $c_{\text{баз}}$ – вектор-строка, образованная коэффициентами при базисных переменных заключительной симплекс-таблицы в целевой функции исходной задачи.

Пример 4.3. Для задачи

$$z=23x_1+40x_2+60x_3+2x_4-x_5-18\to\max$$

$$\begin{cases}4x_1+10x_2+11x_3+x_4+x_5=57,\\ 2x-6x_2-x_3+x_4-x_5=9,\\ x_j\geq 0,\ \ j=1,..,5.\end{cases}$$

построить двойственную, решить исходную задачу симплекс-методом и найти оптимальное решение двойственной задачи.

Решение. Составим задачу, двойственную к данной. Образуем расширенную матрицу

$$\tilde{A}=\left[\begin{array}{ccccc|c|c} 4 & 10 & 11 & 1 & 1 & = & 57\\ 2 & -6 & -1 & 1 & -1 & = & 9\\ \hline \geq & \geq & \geq & \geq & \geq & & \\ \hline 23 & 40 & 60 & 2 & -1 & & -18 \end{array}\right]_{\to\max},$$

и преобразуем ее по общему правилу

$$\tilde{A}' = \left[\begin{array}{cc|c|c} 4 & 2 & \geq & 23 \\ 10 & -6 & \geq & 40 \\ 11 & -1 & \geq & 60 \\ 1 & 1 & \geq & 2 \\ 1 & -1 & \geq & -1 \\ \hline \sim & \sim & & \\ \hline 57 & 9 & & -18 \end{array}\right]_{\to \min} .$$

Двойственная задача имеет вид

$$T = 57y_1 + 9y_2 - 18 \to \min, \text{при условиях} \begin{cases} 4y_1 + 2y_2 \geq 23, \\ 10y_1 - 6y_2 \geq 40, \\ 11y_1 - y_2 \geq 60, \\ y_1 + y_2 \geq 2, \\ y_1 - y_2 \geq -1. \end{cases}$$

Решим исходную задачу симплекс-методом. В задаче не выделен допустимый базис, поэтому для его нахождения можно использовать метод искусственного базиса или выделить его непосредственно из нетривиальных ограничений: сложим уравнения, а затем вычтем из первого второе. Получим систему

$$\begin{cases} 6x_1 + 4x_2 + 10x_3 + 2x_4 = 66, \\ 2x_1 + 16x_2 + 12x_3 + 2x_5 = 48, \end{cases} \text{т.е.} \begin{cases} 3x_1 + 2x_2 + 5x_3 + x_4 = 33, \\ x_1 + 8x_2 + 6x_3 + x_5 = 24. \end{cases}$$

с выделенным базисом x_4, x_5. Выражаем базисные переменные x_4, x_5

$$x_4 = 33 - 3x_1 - 2x_2 - 5x_3,$$
$$x_5 = 24 - x_1 - 8x_2 - 6x_3$$

и подставляем полученные выражения в формулу для z. Получим задачу

$$z = 18x_1 + 44x_2 + 56x_3 + 24 \to \max$$

$$\begin{cases} 3x_1 + 2x_2 + 5x_3 + x_4 = 33, \\ x_1 + 8x_2 + 6x_3 + x_5 = 24, \\ x_j \geq 0, \ \ j = 1,..,5 \end{cases} . \qquad (4.1)$$

Перепишем целевую функцию в виде $z - 18x_1 - 44x_2 - 56x_3 = 24$ и имеем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_4	33	3	2	5	1	0
x_5	24	1	8	6	0	1
z	24	-18	-44	-56	0	0

Вводим в базис переменную x_3. Находим $\min\left\{\frac{33}{5}, \frac{24}{6}\right\} = \frac{24}{6} = 4$, делим вторую строку на 6

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_4	33	3	2	5	1	0
x_5	4	1/6	4/3	1	0	1/6
z	24	-18	-44	-56	0	0

и выводим из базиса переменную x_5. Делим соответствующую строку на 6 и получаем

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_4	13	13/6	-14/3	0	1	-5/6
x_3	4	1/6	4/3	1	0	1/6
z	248	-26/3	-92/3	0	0	28/3

В строке оценок осталась единственная отрицательная оценка, а так как $\min\left\{\frac{13}{13/6}, \frac{4}{1/6}\right\} = 6$, то выводим из базиса переменную x_4 и вводим x_1. Умножаем первую строку на $6/13$, получаем симплекс-таблицу

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_4	6	1	-28/13	0	6/13	-5/13
x_3	4	1/6	4/3	1	0	1/6
z	248	-26/3	-92/3	0	0	28/3

и делаем шаг симплекс-метода

Б.П.	с.ч.	x_1	x_2	x_3	x_4	x_5
x_1	6	1	-28/13	0	6/13	-5/13
x_3	3	0	22/3	1	-1/13	3/13
z	300	0	12	0	4	6

Так как в строке оценок нет отрицательных элементов, то полученная симплекс-таблица – заключительная, оптимальное решение $X^* = (6,0,3,0,0)$ и $z_{\max} = z(X^*) = 300$.

Базисными переменными оптимального решения являются x_1, x_3, поэтому из исходной задачи находим $P = \begin{pmatrix} 4 & 11 \\ 2 & -1 \end{pmatrix}$, а из условия задачи $c_{\text{баз}} = (23,60)$, поэтому оптимальным решением двойственной задачи будет

$$Y^* = (23,60)\begin{pmatrix} 4 & 11 \\ 2 & -1 \end{pmatrix}^{-1} = \left(\frac{11}{2}, \frac{1}{2}\right),$$

причем $T_{\min} = T(Y^*) = 57 \cdot \dfrac{11}{2} + 9 \cdot \dfrac{1}{2} - 18 = 300 = z_{\max}$.

Если в задаче ЛП выделен допустимый базис и базисные переменные исключены из целевой функции, то оптимальным решением двойственной задачи являются элементы строки оценок последней симплекс-таблицы при базисных переменных исходной симплекс-таблицы. Например, для задачи (4.1) двойственной задачей будет

$$T = 33y_1 + 24y_2 + 24 \to \min$$

$$\begin{cases} 3y_1 + y_2 \ge 18, \\ 2y_1 + 8y_2 \ge 44, \\ 5y_1 + 6y_2 \ge 56, \\ y_1 \ge 0, \\ y_2 \ge 0. \end{cases}$$

и общая формула теоремы 4 дает

$$Y^* = (18, 56)\begin{pmatrix} 3 & 5 \\ 1 & 6 \end{pmatrix}^{-1} = (4, 6),\ T\left(Y^*\right) = 33 \cdot 4 + 24 \cdot 6 + 24 = 300.$$

С другой стороны, оптимальное решение $Y^* = (4, 6)$ легко находится из строки оценок последней симплекс-таблицы как коэффициенты при базисных переменных x_4, x_5 исходной.

Задачи для самостоятельного решения

Для следующих задач линейного программирования

а) построить задачу, двойственную данной;

б) решить исходную задачу симплекс-методом и найти решение двойственной задачи.

59. $z = x_1 + 3x_2 + 3x_3 + x_4 + x_5 + 7 \to \max$ при $\overline{x} \ge 0$ и $\begin{cases} 4x_1 + 3x_2 + x_3 - 2x_5 = 5, \\ 2x_1 + 2x_2 + x_4 + 2x_5 = 2. \end{cases}$

60. $z = 2x_1 + x_2 + x_3 - x_4 - x_5 + 8 \to \max$ при $\overline{x} \ge 0$ и $\begin{cases} 2x_1 - 3x_2 + x_3 + x_4 = 9, \\ 3x_1 + 2x_2 + x_3 + x_5 = 7. \end{cases}$

61. $z = x_1 - x_2 + 2x_3 + 2x_4 - x_5 + 13 \to \max$ при $\overline{x} \ge 0$ и $\begin{cases} 3x_1 - x_2 + 4x_3 + x_4 = 3, \\ 2x_1 + x_2 + 2x_3 + x_5 = 7. \end{cases}$

62. $z = x_1 - 2x_2 + x_3 + 2x_4 + 4x_5 + 23 \to \max$ при $\overline{x} \ge 0$ и $\begin{cases} x_1 + 4x_2 + x_4 - 2x_5 = 4, \\ -5x_2 + x_3 + 6x_4 + 5x_5 = 5. \end{cases}$

63. $z = 52x_1 + 72x_2 + 61x_3 + x_4 + x_5 + 8 \to \max$ при $\overline{x} \geq 0$ и

$$\begin{cases} 2x_1 + 6x_2 + x_3 + x_4 = 22, \\ 3x_1 + 2x_2 + 5x_3 + x_5 = 40. \end{cases}$$

64. $z = 12x_1 + 13x_2 + x_3 + 7x_4 - 3x_5 + 2 \to \max$ при $\overline{x} \geq 0$ и

ГЛАВА 5. ТРАНСПОРТНАЯ ЗАДАЧА

Под *транспортной задачей* в дальнейшем понимается задача линейного программирования, в которой требуется найти оптимальный (по стоимости) план перевозок некоторого однородного груза от конечного числа поставщиков $A_1, A_2, \ldots, A_m$ с заданными требностями $b_1, \ldots, b_n$. Стоимость c_{ij} перевозки единицы груза от поставщика A_i к потребителю B_j предполагается известной.

Отметим, что данная постановка задачи может быть значительно расширена или изменена. Например, в приложениях часто рассматриваются задачи перевозки *неоднородного* груза. Также в качестве критерия оптимальности можно рассматривать *время перевозок* (транспортная задача по критерию времени). Подобного рода задачи решаются сведением к однородной транспортной задаче, или для них разработаны другие методы, изложение которых остается за рамками данной книги.

5.1. Постановка задачи

Итак, пусть $X = \left(x_{ij}\right)$ – $m \times n$ матрица, где x_{ij} – объем перевозок от i-го поставщика к j-му потребителю. Тогда общие затраты на перевозку груза определяются функцией $z(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$. Математическая постановка транспортной задачи определяется следующей задачей линейного программирования

$$z(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min$$

при условиях

$$\begin{cases} \sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,\ldots,n, \\ \sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,\ldots,m, \\ x_{ij} \ge 0. \end{cases} \tag{5.1}$$

Первая часть нетривиальных ограничений означает, что все потребности удовлетворены, вторая часть – то, что весь груз вывезен от поставщиков.

Замечание 5.1. Если запасы и потребности задаются целыми числами, то транспортная задача имеет целочисленное оптимальное решение, поэтому транспортную задачу относят формально к задачам целочисленного линейного программирования.

Можно показать, что число базисных переменных в системе ограничений (5.1) равно $m+n-1$.

Поставщики	Потребители						Запасы
	B_1	B_2	$\cdots$	B_j	$\cdots$	B_n	
A_1	c_{11} x_{11}	c_{12} x_{12}	$\cdots$	c_{1j} x_{1j}	$\cdots$	c_{1n} x_{1n}	a_1
A_2	c_{21} x_{21}	c_{22} x_{22}	$\cdots$	c_{2j} x_{2j}	$\cdots$	c_{2n} x_{2n}	a_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
A_i	c_{i1} x_{i1}	c_{i2} x_{i2}	$\cdots$	c_{ij} x_{ij}	$\cdots$	c_{in} x_{in}	a_i
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
A_m	c_{m1} x_{m1}	c_{m2} x_{m2}	$\cdots$	c_{mj} x_{mj}	$\cdots$	c_{mn} x_{mn}	a_m
Потребности	b_1	b_2	$\cdots$	b_j	$\cdots$	b_n	

Таблица 5.1

Определение 5.1. Решение $X = \left(x_{ij}\right)$ (оптимальное решение $X^* = \left(x_{ij}^*\right)$) транспортной задачи, удовлетворяющее условиям (5.1) и имеющее не более $m+n-1$ занятой клетки (ненулевой перевозки), будем называть *опорным планом (оптимальным опорным планом)* транспортной задачи.

Исходные данные задачи представляют в виде таблицы 5.1. Общие запасы определяются суммой $\sum_{i=1}^{m} a_i$, а общая потребность – $\sum_{j=1}^{n} b_j$. Транспортная задача называется *задачей с правильным балансом*, а ее модель *закрытой*, если $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, то есть суммарные запасы поставщиков равны суммарным запросам потребителей. Если $\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{n} b_j$, то такая задача называется *задачей с неправильным балансом*, а ее модель – *открытой*.

5.2. Построение начального опорного плана транспортной задачи

Первым этапом решения транспортной задачи является построение начального опорного плана, т.е. плана перевозок, удовлетворяющего всем ограничениям конкретной транспортной задачи. Сущность методов состоит в том, что начальный опорный план находят за не более чем $m+n-1$ шагов (по числу базисных переменных), на каждом из которых в транспортной таблице заполняют одну клетку, которую называют занятой. Заполнение одной из клеток обеспечивает полностью либо удовлетворение потребности в грузе одного из пунктов назначения (того, в столбце которого находится заполненная клетка), либо вывоз груза из одного из пунктов отправления (из того, в строке которого находится заполняемая клетка). Различаются эти планы по принципам выбора заполняемых клеток и, в зависимости от этого, могут давать планы, более или менее отличные от оптимального.

	B_1	B_2	B_3	B_4	a_i
A_1	1	11	3	13	140
A_2	12	4	8	2	160
A_3	3	5	14	6	100
b_j	80	40	150	130	400

Таблица 5.2

Пример 5.1. Рассмотрим транспортную задачу, заданную таблицей 5.2. В правом нижнем углу стоит сумма запасов (и, одновременно, сумма потребностей, так как модель закрытая) 140+160+100=80+40+150+ 130=400.

Напомним сначала метод *северо-западного угла*. Заполнение таблицы начинаем с левого верхнего (северо-западного) угла таблицы. Так как потребности первого потребителя B_1 равны 80, а запасы первого поставщика A_1 равны 140, то в клетку A_1B_1 вписываем максимально возможную перевозку 80. Потребности B_1 полностью удовлетворены, поэтому первый столбец исключаем из рассмотрения, а оставшиеся запасы первого поставщика, т.е. 60, переносим следующим потребителям. Мы можем 40 записать потребителю B_2 (столбец B_2 исключается), а оставшиеся 20 – B_3 и исключить первую строку из дальнейшего рассмотрения.

	B_1	B_2	B_3	B_4	a_i
A_1	1 **80**	11 **40**	3 **20**	13	140
A_2	12	4	8 **130**	2 **30**	160
A_3	3	5	14	6 **100**	100
b_j	80	40	150	130	400

Таблица 5.3

Далее, так как потребности B_3 равны 150, а 20 единиц груза ему уже доставлены, то оставшиеся 130 единиц доставляются от второго поставщика A_2 (заполняем клетку A_2B_3). Столбец B_3 исключаем из рассмотрения, а оставшиеся запасы второго поставщика (30 единиц) записываем потребителю B_4. Окончательно потребности последнего удовлетворяются за счет поставщика A_3: вписываем в клетку A_3B_4 перевозку 100. Заметим, что, так как исходная задача - с правильным балансом, то потребности последнего потребителя B_4 равны запасам поставщика A_3, т.е. 100. Получаем таблицу 5.3 с начальным опорным планом

$$X = \begin{pmatrix} 80 & 40 & 20 & 0 \\ 0 & 0 & 130 & 30 \\ 0 & 0 & 0 & 100 \end{pmatrix}.$$

Суммарная стоимость перевозок равна

$$z(X) = 1\cdot 80 + 11\cdot 40 + 3\cdot 20 + 8\cdot 130 + 2\cdot 30 + 6\cdot 100 = 2280.$$

Из решения видно, что метод северо-западного угла, с одной стороны, достаточно прост с точки зрения построения, а с другой стороны, не учитывает стоимость пере-

A_1	1 **80**	11	3 **60**	13	140
A_2	12	4 **30**	8	2 **130**	160
A_3	3	5 **10**	14 **90**	6	100
b_j	80	40	150	130	400
	B_1	B_2	B_3	B_4	a_i

Таблица 5.4

возок. Поэтому опорный план, построенный методом северо-западного угла, как правило, далек от оптимального.

Построим теперь для этой же задачи начальный опорный план *методом минимального тарифа*. Суть этого метода состоит в том, что в клетки с наименьшими тарифами помещают максимально возможные перевозки. Итак, в таблице исходной задачи выбираем клетку с минимальным тарифом, т.е. клетку A_1B_1 с тарифом 1. Запасы поставщика A_1 равны 140, а потребности B_1 – 80, поэтому в клетку A_1B_1 вписываем максимально возможную перевозку 80, и потребителя B_1 исключаем из рассмотрения. В оставшейся части таблицы выбираем минимальный тариф, т.е. клетку A_2B_4 с тарифом 5. Запасы поставщика A_2 равны 140, а потребности B_4 - 130, поэтому в клетку A_2B_4 записываем перевозку 130 и потребителя B_4 исключаем из рассмотрения. У оставшихся потребителей B_2, B_3 выбираем клетку с минимальным тарифом. Это A_1B_3 с тарифом 3. Запасы (оставшиеся) поставщика A_1 равны 60, а потребности B_3 – 150, поэтому в клетку A_1B_3 записываем максимально возможную перевозку 60 и исключаем поставщика A_1 из дальнейшего рассмотрения. Далее, аналогично в клетку A_2B_2 записываем 30 и исключаем второго поставщика.

В оставшиеся две клетки A_3B_2 и A_3B_3 последовательно вписываем перевозку 10 в A_3B_2 и 90 в A_3B_3. Получаем таблицу 5.4 с начальным опорным планом $X = \begin{pmatrix} 80 & 0 & 60 & 0 \\ 0 & 30 & 0 & 130 \\ 0 & 10 & 90 & 0 \end{pmatrix}$. Суммарная стоимость перевозок равна

$$z(X) = 1\cdot 80 + 3\cdot 60 + 4\cdot 30 + 2\cdot 130 + 5\cdot 10 + 14\cdot 90 = 1950 < 2280.$$

Таким образом, опорный план, построенный методом минимального тарифа, лучше, чем план, полученный методом северо-западного угла.

Применим, наконец, к исходной задаче метод аппроксимации Фогеля. Для этого найдем разность между двумя минимальными тарифами для каждой строки и столбца таблицы и запишем их в дополнительно образованные строки и столбцы (см. таблицу 5.5). В строке A_1 минимальный тариф равен 1, а следующий за ним 3, поэтому разность между ними 4-2=2; в строке A_2 минимальный тариф равен 2, а следующий за ним 4, поэтому разность между равна 2;

аналогично, для строки A_3 разность между минимальным тарифом 3 и следующим за ним 5 равна 5. Итак, три двойки записываем в первый дополнительный столбец.

Аналогично для столбцов разности 3-1=2, 5-4=1, 8-3=5 и 6-2=4 записываем в первую дополнительную строку. Теперь из всех разностей выбираем *максимальную*, т.е. 5 в столбце B_3, и в клетку A_1B_3 с минимальным тарифом в этом столбце записываем максимально возможную перевозку 140. При этом поставщика A_1 исключаем из рассмотрения. Теперь аналогично вычисляем разности между оставшимися минимальными тарифами и заполняем вторые дополнительные столбец и строку, не учитывая тарифы в строке A_1. Видим, что теперь максимальная разность получается в столбце B_1 и перевозку 80 записываем в клетку A_3B_1 с минимальным тарифом 3 в этом столбце (первую строку мы исключили из рассмотрения). Столбец B_1 аналогично исключаем из рассмотрения. Как видно из

	B_1	B_2	B_3	B_4	a_i	Разности по строкам					
A_1	1	11	3 **140**	13	140	2	–	–	–	–	–
A_2	12	4 **20**	8 **10**	2 **130**	160	2	2	2	2	0	–
A_3	3 **80**	5 **20**	14	6	100	2	2	1	1	0	0
b_j	80	40	150	130	400						
Разности по столбцам	2	1	5	4							
	9	1	6	4							
	–	1	6	4							
	–	1	–	4							
	–	1	–	–							
	–	0	–	–							

Таблица 5.5

таблицы, на следующем шаге вписываем перевозку 10 в клетку A_2B_3 и исключаем столбец B_3, затем – максимально возможную перевозку 40 в клетку A_2B_4 и исключаем из рассмотрения столбец B_4. Теперь для вычисления дальнейших разностей остается единственный столбец B_2, поэтому в качестве разностей по строкам записываем нули. Далее, в клетку A_2B_2 записываем 20, а на последнем шаге записываем перевозку 20 в клетку A_3B_2. Получаем таблицу с начальным опорным планом $X=\begin{pmatrix} 0 & 0 & 140 & 0 \\ 0 & 20 & 10 & 130 \\ 80 & 20 & 0 & 10 \end{pmatrix}$, а общая стоимость перевозок

$$z(X)=3\cdot 140+4\cdot 20+8\cdot 10+2\cdot 130+3\cdot 80+5\cdot 20=1180<1950.$$

Отметим, что методом Фогеля обычно получается план, близкий к оптимальному, или сам оптимальный план.

Замечание 5.5. *В общем случае опорный план транспортной задачи состоит из* $m+n-1$ *занятой клетки (по числу базисных переменных). Такой план называется* ***невырожденным****. Нередко при решении транспортной задачи возникает* ***вырожденный*** *план с меньшим числом занятых клеток* (*когда какие-то из базисных переменных равны* 0). *В этом случае выбирается свободная клетка (или несколько свободных клеток – в зависимости от вырожденности плана) с* ***наименьшим*** *тарифом, которая в дальнейшем формально считается занятой с* ***нулевой*** *перевозкой.*

5.3. Решение транспортной задачи методом потенциалов

Вторым этапом решения транспортной задачи является проверка построенного плана на оптимальность и его улучшение (если он не оптимален) с помощью метода потенциалов. Применение метода потенциалов основано на следующей теореме

Теорема 5.1. *Если опорный план* $X=\left(x_{ij}\right)$ *транспортной задачи является оптимальным, то существуют потенциалы поставщиков* $u_i, i=1,\ldots,m$ *и потребителей* v_j, $j=1,\ldots,n$, *удовлетворяющие условиям:*

$$u_i+v_j=c_{ij} \text{ при } x_{ij}>0 \text{ (для занятых клеток),} \tag{5.2}$$

$$\Delta_{ij}=u_i+v_j-c_{ij}\le 0 \text{ при } x_{ij}=0 \text{ (для свободных клеток).} \tag{5.3}$$

u_i \ v_j	2	4	8	2	a_i
-5	1 $\boxed{-4}$	11 $\boxed{-12}$	3 **140**	13 $\boxed{-16}$	140
0	12 $\boxed{-10}$	4 **20**	8 **10**	2 **130**	160
1	3 **80**	5 **20**	14 $\boxed{-5}$	6 $\boxed{-3}$	100
b_j	80	40	150	130	400

Таблица 5.6

Условия (5.2) образуют систему с $m+n$ неизвестными u_i, v_j и, в общем случае, $m+n-1$ уравнений. Так как число неизвестных системы на единицу больше числа уравнений, то одну из неизвестных можно задать произвольно, а остальные найти из системы.

Числа $\Delta_{ij} = u_i + v_j - c_{ij}$ называют *оценками свободных клеток*. Таким образом, согласно теореме, *опорный план будет оптимален, если для всех свободных клеток таблицы оценки неположительные.*

Проверим теперь оптимальность планов, построенных выше.

Пример 5.5. Сначала рассмотрим начальный опорный план, построенный методом минимального тарифа и методом Фогеля. Потенциалы будем записывать в первые строку и столбец вместо обозначений поставщиков и потребителей.

Так как одну из неизвестных можно задать произвольно, то удобнее всего выбирать в качестве исходной переменной тот потенциал, в строке которого больше всего занятых клеток. Здесь, таким образом, полагаем $u_2 = 0$. Из условий (5.2) $u_2 + v_2 = 4$, $u_2 + v_3 = 8$, $u_2 + v_4 = 2$

$u_i \backslash v_j$	1	-6	3	-8	a_i
0	1 **80**	11 $\boxed{-17}$	3 **60**	13 $\boxed{-21}$	140
10	12 $\boxed{-1}$	4 **30**	8 $\boxed{5}$	2 **130**	160
11	3 $\boxed{9}$	5 **10**	14 **90**	6 $\boxed{-3}$	100
b_j	80	40	150	130	400

Таблица 5.7

сразу находим, что $v_2 = 4$, $v_3 = 8$, $v_4 = 2$. Далее, из $u_3 + v_2 = 5$ получаем, что $u_3 = 1$, из $u_1 + v_3 = 3$ следует $u_1 = -5$, а из $u_3 + v_1 = 3$ заключаем, что $v_1 = 2$. Все потенциалы найдены (см. таблицу 5.6).

Теперь находим оценки для свободных клеток

$$\Delta_{11} = u_1 + v_1 - c_{11} = -4 < 0,\ \Delta_{12} = u_1 + v_2 - c_{12} = -12 < 0,$$
$$\Delta_{14} = u_1 + v_4 - c_{14} = -16 < 0,\ \Delta_{21} = u_2 + v_1 - c_{21} = -10 < 0,$$
$$\Delta_{33} = u_3 + v_3 - c_{33} = -5 < 0,\ \Delta_{34} = u_3 + v_4 - c_{34} = -3 < 0.$$

Результат записываем в таблицу 5.6 (где в свободных клетках в квадратике записаны оценки). Все оценки отрицательны, поэтому план $X^* = \begin{pmatrix} 0 & 0 & 140 & 0 \\ 0 & 20 & 10 & 130 \\ 80 & 20 & 0 & 10 \end{pmatrix}$ оптимален и $z_{\min} = z(X^*) = 1180$.

Пример 5.3. Теперь проверим на оптимальность план перевозок, полученный методом минимального тарифа. Ясно, что в силу большей суммарной стоимости перевозок план не оптимален, но вычисление

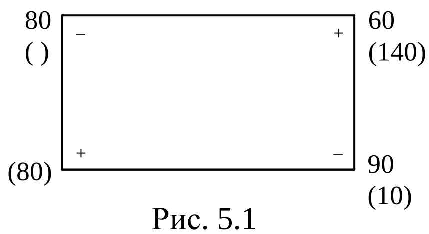


Рис. 5.1

потенциалов и оценок необходимо для того, чтобы этот начальный опорный план улучшить. Проделав вычисления, аналогичные примеру 5.2, для опорного плана примера 5.1, построенного методом минимального тарифа, получаем таблицу 5.7. Как видим, среди оценок есть положительные, поэтому, как и ожидалось, опорный план

$$X = \begin{pmatrix} 80 & 0 & 60 & 0 \\ 0 & 30 & 0 & 130 \\ 0 & 10 & 90 & 0 \end{pmatrix}$$ не оптимален.

Чтобы улучшить опорное решение X транспортной задачи, введем понятие цикла. Напомним, что *циклом* называется последовательность клеток таблицы транспортной задачи, в которой две и только две соседние клетки расположены в одной строке или столбце. Цикл обычно изображают в виде замкнутой ломаной линии, соединяющей вершины цикла, расположенные в клетках таблицы.

u_i \ v_j	3	5	14	1	a_i
-11	1 $\boxed{-9}$	11 $\boxed{-17}$	3 **140**	13 $\boxed{-25}$	140
1	12 $\boxed{-8}$	4 **30**	8 $\boxed{7}$	2 **130**	160
0	3 **80**	5 **10**	14 **10**	6 $\boxed{-5}$	100
b_j	80	40	150	130	400

Таблица 5.8

Для построения нового опорного плана в таблице выбираем свободную клетку с *максимальной положительной оценкой* (клетка A_3B_1) и формируем цикл, одной из вершин которого является выбранная клетка, а остальные клетки *занятые*. Легко видеть, что это цикл, соединяющий клетки A_3B_1, A_1B_1, A_1B_3, и A_3B_3. Кроме этого, сопоставим каждой вершине цикла знак и перевозку, при этом свободной клетке сопоставляем знак «$+$», а для остальных клеток знаки чередуются. Получим следующий цикл, изображенный на рисунке 5.1. Теперь сделаем перестановку по циклу, а именно: из всех вершин, отмеченных минусом, вычтем минимум из всех перевозок, означенных этим знаком, т.е. в читаем $\Delta = \min(80,90) = 80$, а ко всем вершинам с «$+$» прибавим Δ. Получим новые значения перевозок, обозначенные на рис. 5.1 в скобках.

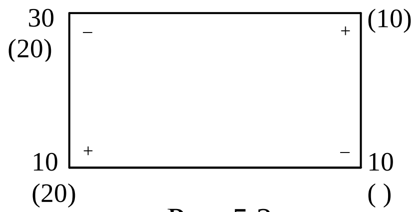


Рис. 5.2

При этом клетка A_1B_1 (обозначена знаком «()») становится свободной, и мы получаем новый опорный план (таблица 5.8). Общая стоимость перевозок равна

$$z(X) = 3\cdot 140 + 4\cdot 30 + 2\cdot 130 + 3\cdot 80 + 5\cdot 10 + 14\cdot 10 = 1230 < 1950.$$

Полученный план лучше начального, и, оценивая его оптимальность с помощью метода потенциалов, видим, что есть положительные оценки, и план не оптимален (см. таблицу 5.8). Снова выбираем свободную клетку с положительной оценкой (здесь такая клетка единственная – клетка A_2B_3) и формируем цикл с вершиной в этой клетке. Таковым является цикл, соединяющий клетки A_2B_3, A_3B_3, A_3B_2 и A_2B_2 (рис. 5.2). Так как $\Delta = \min(30,10) = 10$, то после перестановки по циклу получаем новый план $X^* = \begin{pmatrix} 0 & 0 & 140 & 0 \\ 0 & 20 & 10 & 130 \\ 80 & 20 & 0 & 10 \end{pmatrix}$, фактически уже возникавший в таблице 5.6. Его оптимальность уже была проверена.

Задачи для самостоятельного решения

Решить методом потенциалов транспортные задачи

65.

	B_1	B_2	B_3	B_4	a_i
A_1	8	6	9	2	160
A_2	7	16	12	12	60
A_3	6	15	8	3	180
b_j	80	60	60	200	

66.

	B_1	B_2	B_3	B_4	a_i
A_1	4	4	8	6	80
A_2	11	15	24	18	50
A_3	11	22	15	14	180
b_j	100	10	40	160	

67.

	B_1	B_2	B_3	B_4	a_i
A_1	10	10	4	8	90
A_2	14	25	13	23	60
A_3	12	13	6	12	140
b_j	80	40	90	80	

68.

	B_1	B_2	B_3	B_4	a_i
A_1	6	10	8	8	160
A_2	11	29	14	18	80
A_3	11	26	16	25	70
b_j	100	70	30	110	

69.

	B_1	B_2	B_3	B_4	a_i
A_1	12	5	8	6	70
A_2	12	17	13	17	60
A_3	10	18	10	14	140
b_j	90	50	100	30	

70.

	B_1	B_2	B_3	B_4	a_i
A_1	4	14	11	18	30
A_2	3	17	1	10	130
A_3	9	16	11	18	120
b_j	50	70	30	130	

71.

	B_1	B_2	B_3	B_4	a_i
A_1	5	9	14	10	120
A_2	20	15	20	20	140
A_3	12	8	14	17	70
b_j	80	90	70	90	

72.

	B_1	B_2	B_3	B_4	a_i
A_1	14	9	12	4	150
A_2	12	15	19	16	70
A_3	15	19	15	12	210
b_j	140	50	100	140	

5.4. Открытая модель транспортной задачи

Напомним, что транспортная задача $m \times n$ называется *задачей с неправильным балансом*, а ее модель – *открытой*, если $\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{n} b_j$, т.е. суммарные запасы не равны суммарным потребностям.

	B_1	B_2	B_3	B_4	a_i
A_1	1	11	3	13	150
A_2	12	4	8	2	160
A_3	3	5	14	6	120
b_j	80	40	150	130	

Таблица 5.9

Открытую задачу можно свести к замкнутой:

	B_1	B_2	B_3	B_4	B_5	a_i	Разности по строкам						
u_j \ v_j	0	3	4	2	–3								
0	11 $\boxed{-11}$	5 $\boxed{-2}$	4 **10**	2 **90**	0 $\boxed{-3}$	100	2	2	2	1	0	–	–
1	1 **70**	4 **60**	5 **70**	9 $\boxed{-6}$	0 $\boxed{-2}$	200	1	4	1	1	0	0	–
3	9 $\boxed{-6}$	8 $\boxed{-2}$	7 **100**	10 $\boxed{-5}$	0 **30**	130	7	7	1	1	0	0	0
b_j	70	60	180	90	30	400							
Разности по столбцам	8	1	1	7	0								
	–	1	1	7	0								
	–	1	1	7	–								
	–	1	1	–	–								
	–	–	1	–	–								
	–	–	2	–	–								
	–	–	0	–	–								

Таблица 5.10

1. Если $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$, то вводят фиктивного потребителя B_{n+1} с потребностью $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ и нулевыми тарифами перевозок в столбце.

2. Если $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$, то вводят фиктивного поставщика A_{m+1} с запасом $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$ и нулевыми тарифами перевозок в строке.

Пример 5.4. Рассмотрим задачу примера 5.1, но с измененным запасами, заданную таблицей 5.9. Так как сумма запасов $150+160+120=430$ больше суммы потребностей $70+60+180+90=400$, то вводим фиктивного потребителя B_5 с нулевыми тарифами перевозок и потребностями 30 (таблица 5.10).

Методом аппроксимации Фогеля построим начальный план и методом потенциалов проверим полученный план на оптимальность.

Получаем, что все оценки отрицательны, поэтому полученный план

$$X^* = \begin{pmatrix} 0 & 0 & 10 & 90 \\ 70 & 60 & 70 & 0 \\ 0 & 0 & 100 & 0 \end{pmatrix}$$

(для его записи мы отбрасываем столбец фиктивного потребителя B_5) оптимален и $F\left(X^*\right)=1580$. Как следствие неправильного баланса имеем, что от поставщика A_3 не вывезено 30 единиц груза.

	B_1	B_2	B_3	B_4	a_i
A_1	4	14	11	18	60
A_2	3	17	1	10	130
A_3	9	16	11	18	120
b_j	60	100	30	160	

Таблица 5.11

u_j \ v_j	3	8	1	10	a_i
6	4 $\boxed{5}$	14 **60**	11 $\boxed{-4}$	18 $\boxed{-2}$	60
0	3 **60**	17 $\boxed{-9}$	1 **30**	10 **40**	130
8	9 $\boxed{2}$	16 **40**	11 $\boxed{-1}$	18 **80**	120
-10	0 $\boxed{-7}$	0 $\boxed{-2}$	0 $\boxed{-9}$	0 **40**	40
b_j	60	100	30	160	

Таблица 5.12

Пример 5.5. В транспортной задаче, заданной таблицей 5.11, сумма запасов равна 310, а потребностей – 350. Поэтому необходимо ввести дополнительного поставщика A_4 с запасом 40 и тарифами перевозок 0 (см. таблицу 5.12).

Построим начальный план методом минимального тарифа и вычислим потенциалы. Легко видеть, что план не оптимален, максимальная положительная оценка равна 5, и вершинами цикла являются клетки A_1B_1, A_2B_1, A_2B_4, A_3B_4, A_3B_2 и A_1B_2. Получается цикл «с самопересечением», изображенный на рис. 5.3. Получаем, что $\Delta = \min(60{,}80) = 60$. Заметим, что после перестановки по циклу сразу в двух клетках получается нулевая перевозка.

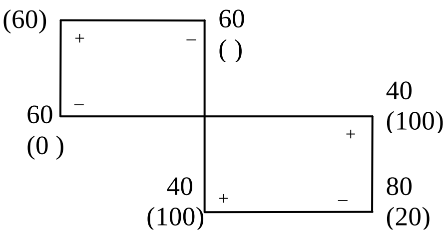


Рис. 5.3

Замечание 5.3. Если после перестановки по циклу больше чем в одной клетке образуется нулевая перевозка, то одна из них становится свободной (желательно, с максимальным тарифом), а остальные считаются занятыми с нулевой перевозкой, чтобы число занятых клеток оставалось равным $m+n-1$.

$u_j \backslash v_j$	3	8	1	10	a_i
1	4 **60**	14 $\boxed{-5}$	11 $\boxed{-9}$	18 $\boxed{-7}$	60
0	3 **0**	17 $\boxed{-9}$	1 **30**	10 **100**	130
8	9 $\boxed{2}$	16 **100**	11 $\boxed{-2}$	18 **20**	120
-10	0 $\boxed{-7}$	0 $\boxed{-2}$	0 $\boxed{-9}$	0 **40**	40
b_j	60	100	30	160	

Таблица 5.13

Таким образом, получаем новый план, указанный в таблице 5.13. Имеется единственная клетка A_2B_1 с положительной оценкой, причем в цикле, изображенном на рисунке 5.4, одна из клеток, означенных минусом, имеет нулевую перевозку.

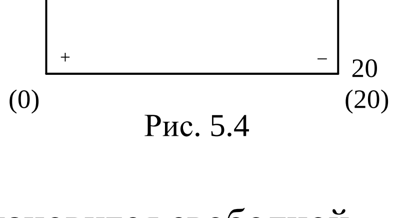


Рис. 5.4

Замечание 5.4. Если занятая клетка с нулевой перевозкой попала в цикл и соответствует знаку “–“ (при этом $\Delta = 0$), то перестановка по циклу сводится к тому, что свободная клетка объявляется занятой с нулевой перевозкой, а занятая клетка с нулевой перевозкой становится свободной.

$u_j \backslash v_j$	1	8	1	10	a_i
3	4 **60**	14 $\boxed{-3}$	11 $\boxed{-7}$	18 $\boxed{-5}$	60
0	3 $\boxed{-2}$	17 $\boxed{-9}$	1 **30**	10 **100**	130
8	9 **0**	16 **100**	11 $\boxed{-2}$	18 **20**	120
-10	0 $\boxed{-9}$	0 $\boxed{-2}$	0 $\boxed{-9}$	0 **40**	40
b_j	60	100	30	160	

Таблица 5.14

Таким образом, занятая нулевая и свободная клетки меняются местами (рис. 5.4). Проверяя план на оптимальность (таблица 5.14), убеждаемся, что все оценки отрицательны,

$$X^* = \begin{pmatrix} 60 & 0 & 0 & 0 \\ 0 & 0 & 30 & 100 \\ 0 & 100 & 0 & 20 \end{pmatrix}$$

(мы отбрасываем строку фиктивного поставщика) оптимален, и $F\left(X^*\right) = 3230$. В результате получаем, что потребности потребителя B_4 удовлетворены не полностью.

Задачи для самостоятельного решения

Решить транспортные задачи методом потенциалов

73.

	B_1	B_2	B_3	B_4	a_i
A_1	3	7	4	7	100
A_2	10	13	24	7	100
A_3	8	19	12	18	200
b_j	90	80	30	170	

74.

	B_1	B_2	B_3	B_4	a_i
A_1	1	13	12	3	60
A_2	2	16	4	6	125
A_3	13	4	17	16	75
b_j	100	100	50	50	

75.

	B_1	B_2	B_3	B_4	a_i
A_1	5	6	2	4	75
A_2	18	22	11	3	185
A_3	18	9	6	11	90
b_j	40	70	90	115	

76.

	B_1	B_2	B_3	B_4	a_i
A_1	11	3	7	4	95
A_2	12	9	8	13	65
A_3	23	14	3	8	130
b_j	40	110	85	105	

77.

	B_1	B_2	B_3	B_4	a_i
A_1	14	7	2	17	100
A_2	5	12	6	10	150
A_3	9	2	3	12	120
b_j	60	95	85	90	

78.

	B_1	B_2	B_3	B_4	a_i
A_1	6	12	7	14	120
A_2	12	3	18	3	80
A_3	23	1	3	21	70
b_j	80	110	80	50	

5.5. Определение оптимального плана транспортных задач с дополнительными ограничениями

При решении транспортных задач методом потенциалов возможно учитывать дополнительные ограничения на перевозки. Ниже перечислены варианты различных постановок транспортных задач и даны соответствующие методы сведения к закрытой транспортной задаче.

1. Если в закрытой транспортной задаче перевозки от поставщика A_i к потребителю B_j не могут быть осуществлены (*блокировка*), то для определения оптимального решения задач предполагают, что тариф перевозки единицы груза от A_i к B_j равен сколь угодно большому числу M .

5. Если дополнительным условием в транспортной задаче является обеспечение перевозки от поставщика A_i к потребителю B_j *в точности* a_{ij} единиц груза, то в клетку A_iB_j записывают указанное число a_{ij}, а эту клетку считают свободной со сколь угодно большим тарифом M .

3. Если от поставщика A_i к потребителю B_j должно быть перевезено *не менее* a_{ij} единиц груза, что запасы пункта A_i и потребности пункта B_j полагают меньше фактических на a_{ij} единиц. После нахождения оптимального плана перевозку, стоящую в клетке A_iB_j, увеличивают на a_{ij} единиц.

4. Если от поставщика A_i к потребителю B_j требуется перевезти *не более* a_{ij} единиц груза, то вводят дополнительного потребителя $B_{n+1} = B_{ij}$, которому записывают те же тарифы, что и для B_j, за исключением тарифа в i-ой строке, который считают равным сколь угодно большому числу M. Потребности пункта B_j считают равными a_{ij}, а потребности B_{ij} полагают равными $b_j - a_{ij}$.

u_i \ v_j	2	4	8	2	a_i
-5	1	11	3	13	140
0	12	4	8	2	160
1	3	5	14	6	100
b_j	80	40	150	130	400

Таблица 5.15

Пример 5.6. Найти решение транспортной задачи, заданной таблицей 5.15, если из A_3 в B_1 и из A_2 в B_3 перевозки не могут быть осуществлены, из A_1 в B_2 должно быть завезено не менее 30 ед. груза, а из A_2 в B_4 ровно 70 ед.

$u_i \backslash v_j$	1	7–M	3	-5	a_i
0	1 **80**	11 $\boxed{-M-4}$	3 **30**	13 $\boxed{-18}$	110
M–3	12 $\boxed{M-14}$	4 **10**	M **80**	2 **70**	160
11	M $\boxed{12-M}$	5 $\boxed{13-M}$	14 **40**	6 **60**	100
b_j	80	10	150	130	400

Таблица 5.16

Решение. Так как из A_2 в B_1 и из A_3 в B_4 перевозки не могут быть осуществлены, то в клетках A_3B_1 и A_2B_3 тарифы считаем равными некоторому большому числу M. В клетке A_2B_4 перевозку считаем равной 70, а тариф – равным M, и эту клетку в дальнейшем полагаем свободной. Кроме этого, запасы A_1 и потребности B_2 уменьшаем на 30.

Рис. 5.5

Получаем таблицу 5.16, строим начальный опорный план методом минимального тарифа, находим потенциалы и оценки свободных клеток (см. таблицу 5.16). Имеем единственную положительную оценку $\Delta_{12} = M - 14$ и строим цикл, соединяющий клетки A_1B_1, A_1B_3, A_2B_3 и A_2B_1 (рис. 5.5). В клетках, означенных минусом, перевозки одинаковы, поэтому в соответствии с замечанием 5.3, клетку A_1B_1 считаем занятой с нулевой перевозкой, а A_2B_3 – свободной. После перестановки по циклу получаем новый план (таблица 5.17). Так как все оценки отрицатель-

$u_i \backslash v_j$	1	-7	3	-5	a_i
0	1 **0**	11 $\boxed{-18}$	3 **110**	13 $\boxed{-18}$	110
11	12 **80**	4 **10**	M $\boxed{14-M}$	2 **70**	160
11	M $\boxed{12-M}$	5 $\boxed{-1}$	14 **40**	6 **60**	100
b_j	80	10	150	130	400

Таблица 5.17

ны, то данная таблица – заключительная. Увеличиваем перевозку в клетке A_1B_2 на 30, получим оптимальный план перевозок, удовлетворяющий всем ограничениям задачи: $X^*=\begin{pmatrix} 0 & 30 & 110 & 20 \\ 80 & 10 & 0 & 70 \\ 0 & 0 & 40 & 60 \end{pmatrix}$ и

$$F\left(X^*\right)=3\cdot110+12\cdot80+4\cdot10+2\cdot70+14\cdot40+6\cdot60=2390.$$

$u_i \backslash v_j$	1	-1	3	-3	1	a_i
0	1 **50**	11 $\boxed{-12}$	3 **20**	13 $\boxed{-16}$	1 **30**	100
5	12 $\boxed{-6}$	4 **40**	8 **30**	2 **90**	M $\boxed{6-M}$	160
M–3	7 $\boxed{M-9}$	5 $\boxed{M-9}$	M **100**	6 $\boxed{M-12}$	7 $\boxed{M-9}$	100
b_j	50	40	150	90	30	400

Таблица 5.18

Пример 5.7. Найти решение транспортной задачи, заданной таблицей 5.15, если из A_3 в B_3 перевозки запрещены, а из A_2 в B_1 должно быть завезено не более 50 ед. груза, а из A_1 в B_4 – не менее 40 ед. груза.

Решение. Так как из A_3 в B_3 перевозки запрещены, то тариф в A_3B_3 считаем равным M. Запасы A_1 и потребности B_4 уменьшаем на 40,

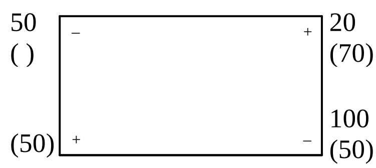


Рис. 5.6

а также вводим дополнительного потребителя B_5 с потребностями $80-50=30$. Соответственно, в клетке A_3B_5 стоимость перевозок считаем равной M, а потребности B_1 приравниваем к 50.

Получаем таблицу 5.18. Найдем начальное опорное решение методом минимального тарифа, потенциалы и оценки. Анализируя циклы, проходящие через клетки с максимальной положительной оценкой $M-9$, вибираем цикл, содержащий клетку A_3B_1, так как при этом перевозка в ячейке A_3B_3 максимально уменьшается. Итак, в результате перестановки по циклу,

$u_i \backslash v_j$	15–M	4	8	2	6	a_i
-5	1 $\boxed{9-M}$	11 $\boxed{-12}$	3 **70**	13 $\boxed{-16}$	1 **30**	100
0	12 $\boxed{3-M}$	4 **40**	8 **30**	2 **90**	M $\boxed{6-M}$	160
M-8	7 **50**	5 $\boxed{M-9}$	M **50**	6 $\boxed{M-12}$	7 $\boxed{M-9}$	100
b_j	50	40	150	90	30	400

Таблица 5.19

соединяющему клетки A_3B_1, A_1B_1, A_1B_3 и A_3B_3 (см. рис. 5.6), получаем новую таблицу 5.19.

Опять получаем две ячейки с максимальной положительной оценкой $\Delta_{32} = \Delta_{34} = M - 9$. Так как для цикла с началом в A_3B_2 перевозка в клетке A_3B_3 уменьшается на 40, а для цикла с началом в A_3B_4 эта же перевозка уменьшается лишь на 30, то для получения нового опорного плана выбираем цикл, содержащий ячейки A_3B_2, A_2B_2, A_2B_3 и A_3B_3. Перестановка по циклу изображена на рисунке 5.7, а новый план – в таблице 5.20.

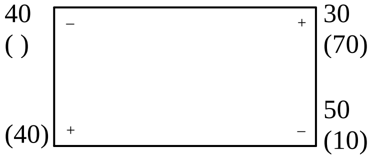


Рис. 5.7

После очередного перехода по циклу, проходящему через клетки A_3B_5, A_3B_3, A_1B_3 и A_1B_5, получим таблицу 5.21. Как легко проверить, все оценки здесь не положительны, и опорный план, определяемый таблицей 5.21, является заключительным. Данная таблица – заключительная, и, увеличивая перевозку в клетке A_1B_4 на 40, а также объединяя перевозки, записанные в соответствующих клетках столбцов B_1 и B_5, получим оптимальный план перевозок

u_i \ v_j	7	5	M	M–6	M–2	a_i
3–M	1 $\boxed{9-M}$	11 $\boxed{3-M}$	3 **70**	13 $\boxed{-16}$	1 **30**	100
8–M	12 $\boxed{3-M}$	4 $\boxed{9-M}$	8 **70**	2 **90**	M $\boxed{6-M}$	160
0	7 **50**	5 **40**	M **10**	6 $\boxed{M-12}$	7 $\boxed{M-9}$	100
b_j	50	40	150	90	30	400

Таблица 5.20

$$X^* = \begin{pmatrix} 20 & 0 & 80 & 40 \\ 0 & 0 & 70 & 90 \\ 60 & 40 & 0 & 0 \end{pmatrix},$$ удовлетворяющий всем ограничениям задачи, и

$$z(X^*) = 1 \cdot 20 + 3 \cdot 80 + 13 \cdot 40 + 8 \cdot 70 + 2 \cdot 90 + 7 \cdot 60 + 5 \cdot 40 = 2140.$$

Замечание 5.5. Поскольку решение задачи связано с разделением перевозок одному из потребителей на несколько столбцов, а затем их объединением, то число занятых клеток в оптимальном плане может

u_i \ v_j	7	5	9	3	7	a_i
-6	1 $\boxed{0}$	11 $\boxed{-12}$	3 **100**	13 $\boxed{-16}$	1 $\boxed{0}$	100

-1	12 $\boxed{-6}$	4 **20**	8 **50**	2 **90**	M $\boxed{6-M}$	160
0	7 **50**	5 **20**	M $\boxed{9-M}$	6 $\boxed{-3}$	7 **30**	100
b_j	50	40	150	90	30	400

Таблица 5.21

быть больше, чем $m+n-1$. Так, для задачи примера 5.7, в окончательном плане транспортной задачи 3×4, мы имеем 7 занятых клеток.

Замечание 5.6. Нулевая оценка одной или нескольких свободных клеток говорит о наличии альтернативных решений исходной задачи. Так, рассмотрим цикл A_2B_2, A_2B_3, A_1B_3, A_1B_5, A_3B_5 и A_3B_2, проходящий через ячейку A_2B_2 с нулевой оценкой (рис. 5.8).

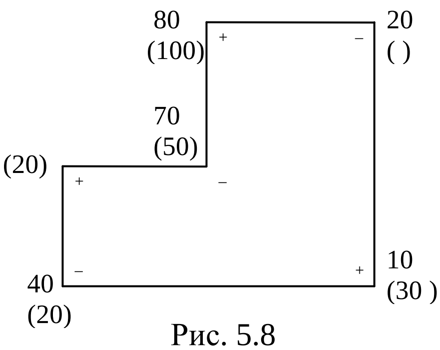


Рис. 5.8

В результате перестановки по циклу с $\Delta=20$, получим новую таблицу 5.22, которая дает новый оптимальный план

$$X_1^*=\begin{pmatrix} 0 & 0 & 100 & 40 \\ 0 & 20 & 50 & 90 \\ 80 & 20 & 0 & 0 \end{pmatrix},$$

удовлетворяющий всем ограничениям задачи. Как и следовало ожидать, суммарная стоимость перевозок

$$z\left(X_1^*\right)=3\cdot100+13\cdot40+4\cdot20+8\cdot50+2\cdot90+7\cdot80+5\cdot20=2140$$

равна $z\left(X^*\right)$.

Задачи для самостоятельного решения

79. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	3	4	7	9	100
A_2	16	5	12	4	100
A_3	8	11	12	5	200
b_j	80	110	90	120	

, если из A_1 в B_2 и из A_2 в B_3 перевозки не могут быть осуществлены.

80. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	5	14	6	21	120
A_2	20	13	17	14	90
A_3	8	21	6	7	180
b_j	95	110	80	70	

, если из A_3 в B_2 перевозки не могут быть осуществлены, а из A_1 в B_4 должно быть перевезено 70 ед. груза.

81. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	7	12	16	11	150
A_2	8	2	4	2	140
A_3	9	15	16	7	110
b_j	75	145	120	60	

, если из A_2 в B_4 перевозки запрещены, из A_1 в B_3 должно быть доставлено не менее 40 ед. груза, а из A_3 в B_1 – не более 50 ед. груза.

82. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	3	10	7	10	140
A_2	4	9	19	25	105
A_3	6	4	5	2	115
b_j	60	130	55	115	

, если из A_2 в B_1 перевозки запрещены, из A_1 в B_2 должно быть перевезено 50 ед. груза, а из A_3 в B_4 – не более 20 ед. груза.

83. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	9	6	7	11	70
A_2	3	14	25	19	170
A_3	2	8	17	10	140
b_j	90	60	160	70	

, если из A_1 в B_4 должно быть перевезено не менее 50 ед. груза, из A_3 в B_3 должно быть перевезено не менее 30 ед. груза, а из A_2 в B_2 – 40 ед. груза.

84. Найти решение транспортной задачи с таблицей

	B_1	B_2	B_3	B_4	a_i
A_1	12	14	11	20	90
A_2	3	4	5	9	155
A_3	2	18	14	12	125
b_j	100	75	75	120	

, если из A_1 в B_4 должно быть перевезено не менее 40 ед. груза, из A_2 в B_3 – не более 50 ед. груза, а из A_3 в B_1 – не менее 60 ед. груза.

5.6. Решения задач линейного программирования средствами EXCEL

Использование программы **Поиск решения** для транспортной задачи совершенно аналогично решению задач линейного программирования (см. п. 5.3). Ниже приводится описание техники решения транспортной задачи на конкретном примере.

Пример 5.8. Пусть транспортная задача задана таблицей 5.22:

	B1	**B2**	**B3**	**B4**	**a**
A1	6	8	15	4	60
A2	9	15	2	3	130
A3	1	12	7	1	90
B	30	80	60	110	

Таблица 5.22

При этом из пункта A_1 в пункт B_3 должно быть перевезено 40 единиц груза, из A_2 в B_2 - не менее 50 единиц и из A_3 в B_3 - не более 20 единиц. Необходимо составить такой план перевозок, при котором общая стоимость перевозок минимальна. В данном случае имеем закрытую модель – суммарные запасы на пунктах отправления равны суммарным заявкам.

Допустимый план перевозок представляется 3×4 матрицей X, удовлетворяющей ограничениям:

$$X \geq 0 \text{ (компоненты матрицы } X \text{ неотрицательны)}, \quad (5.4)$$

$$x_{13} = 40, \quad x_{22} \geq 50, \quad x_{31} \leq 20 \quad (5.5)$$

$$x_{1j} + x_{2j} + x_{3j} + x_{4j} = b_j\,, \quad j = 1, 2, 3, 4, \quad (5.6)$$

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} = a_i \ , \quad i = 1,2,3. \qquad (5.7)$$

Задача заключается в минимизации стоимости перевозок $\sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}x_{ij}$ на множестве допустимых перевозок.

Замечание 5.7. Отметим, что в случае открытости модели вместо ограничений (5.6) и (5.7) следует использовать следующие их модификации.

а) В случае, когда запасы покрывают потребности $\sum a_i > \sum b_j$ следует использовать ограничения $\sum_i x_{ij} = b_j$, $\sum_j x_{ij} \le a_i$.

б) В случае, когда потребности превышают запасы $\sum a_i < \sum b_j$ следует использовать ограничения $\sum_i x_{ij} \le b_j$, $\sum_j x_{ij} = a_i$.

Размещение исходных данных и необходимых формул на листе EXCEL представлено в таблице 5.23.

Таблица 5.23.

	B	C	D	E	F	G	H	I
3		*Матрица стоимости перевозок*						
4		**B1**	**B2**	**B3**	**B4**	**a**		
5	**A1**	6	8	15	4	60		
6	**A2**	9	15	2	3	130		
7	**A3**	1	12	7	1	90		
8	**b**	30	80	60	110			
9		*План*						
10		**B1**	**B2**	**B3**	**B4**	**Sum**		
11	**A1**	0	20	40	0	60	<---=СУММ(C11:F11)	
12	**A2**	10	50	20	50	130	<---=СУММ(C12:F12)	
13	**A3**	20	10	0	60	90	<---=СУММ(C13:F13)	
14	**Sum**	30	80	60	110			
15		*Стоимость плана*		1990				

План перевозок размещается в таблице C11:F13. Присоединенный столбец с именем Sum определяется суммированием соответствующей строки с использованием функции СУММ. Аналогично получается строка Sum – каждый элемент этой строки получается суммированием соответствующего столбца. Для вычисления стоимости перевозок используется функция СУММПРОИЗ (Категория математические функции).

Диалоговое окно *Поиска решения*, представлено на рис. 5.1. Настройка окна проводится с учетом приведенных выше ограничений по схеме, описанной в разделе 5.3.

Минимизации подлежит ячейка E15, в которой вычисляется минимальное значение. Окно **Изменяя ячейки** содержит адрес матрицы перевозок. В окне **Ограничения** отражены условия (5.4) – (5.7). Первая строка соответствует ограничению не отрицательности плана перевозок. Вторая, четвертая и пятая строки соответствуют ограничениям (5.5). Третья и последняя строчки строки выражают ограничения (5.6) и (5.7) соответственно.

Запуск программы поиск решения приводит к оптимальному плану, который представлен в таблице 5.23.

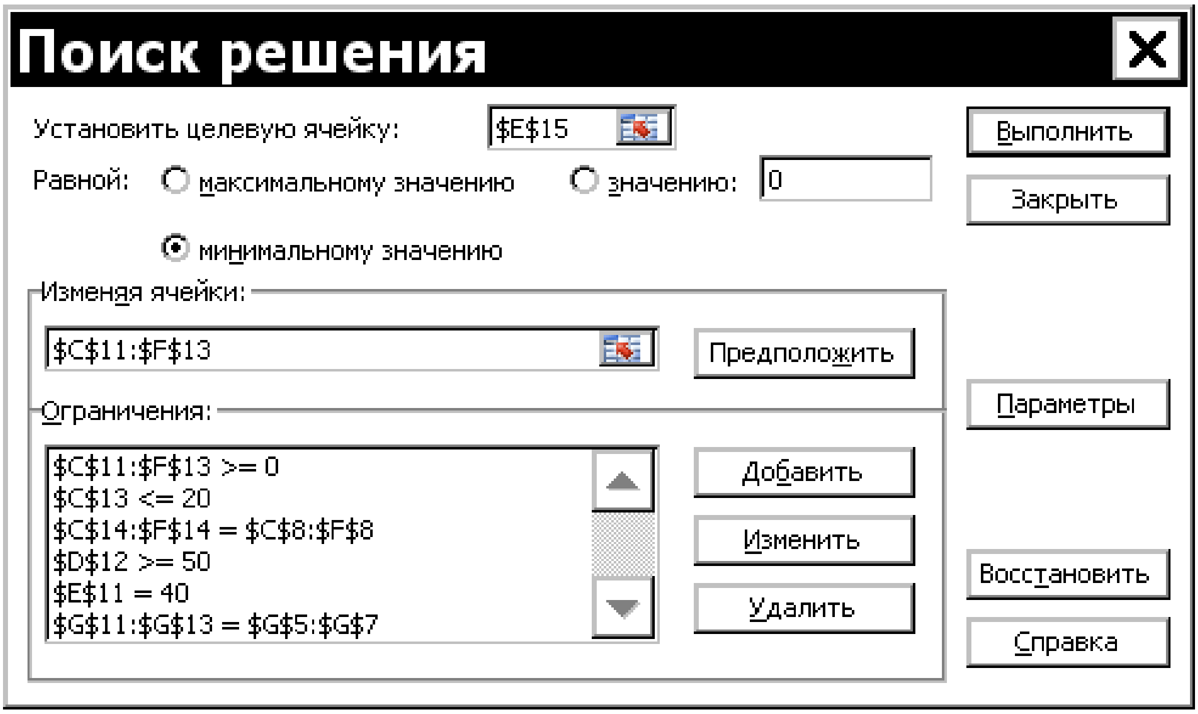


Рис. 5.1. Настроенное окно программы **Поиск Решения**

ГЛАВА 6. ОПТИМИЗАЦИОННЫЕ АЛГОРИТМЫ НА ГРАФАХ

6.1. Основные определения теории графов.

6.1.1. Понятие графа.

Определение 6.1. *Ориентированным графом* (орграфом) называется тройка <M,N,T>, где M, N - конечные множества, а T - отображение $T:N\to M\times M$. Элементы множества N называются вершинами (узлами), а элементы множества N - дугами (ребрами) графа.[1]

Отображение T сопоставляет каждой дуге $u\in N$ упорядоченную пару вершин $(i(u),j(u))$.

Первая называется *началом* дуги u, а вторая - *концом* дуги u. Иногда для обеих вершин используют термин «*граничные вершины*». В дальнейшем эти обозначения будем рассматривать как стандартные: $i(u)$ - начало дуги u, $j(u)$ - конец дуги u.

Определение 6.2. Если вершина i является граничной для дуги u, то говорят, что вершина i и дуга u *инцидентны*.

Пример 6.1. Граф может быть изображен рисунком, на котором вершинам соответствуют точки, а дугам - линии со стрелками, идущими от начал к концам (рис. 6.1).

---

[1] Основные определения и обозначения заимствованы из [13].

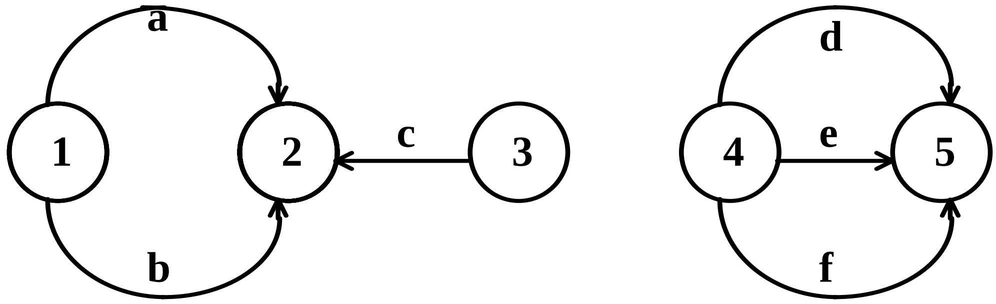


Рис. 6.1

Здесь M=1..5, N={a,b,c,d,e,f}, T(a)=(1,2), T(b)=(1,2), T(c)=(3,2), T(d)=(4,5), T(e)=(4,5), T(f)=(4,5).

Определение 6.3. Граф $<M,N',T>$, где $N'\subseteq N$, называется *частичным графом* графа $<M,N,T>$.

Из определения 6.3 следует, что частичный граф получается из исходного удалением некоторых дуг.

Пример 6.2. На рис. 6.2 приведен один из частичных графов графа, изображенного на рис 6.1.

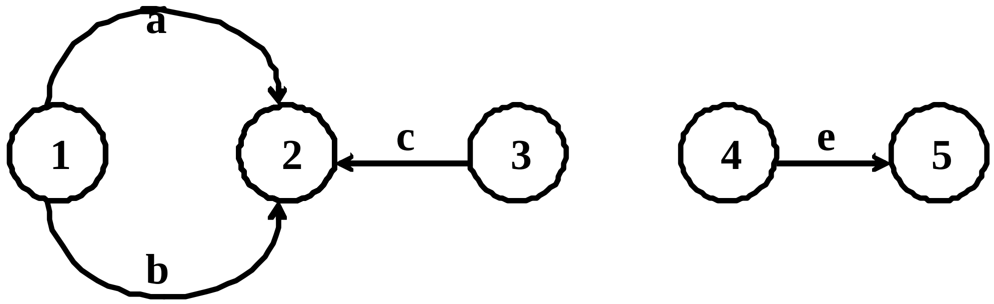


Рис. 6.2

Определение 6.4. Граф $<M',N(M'),T>$, где $M'\subseteq M$, а $N(M')=\{u\in N : i(u),j(u)\in M'\}$, называется *подграфом* графа $<M,N,T>$.

Из определения 6.4 следует, что подграф получается из исходного графа удалением некоторых вершин и всех инцидентных им дуг.

Пример 6.3. На рис 6.3 изображен один из подграфов графа, изображенного на рис 6.1.

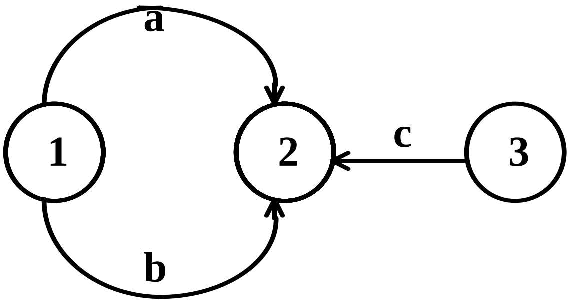


Рис. 6.3

Определение 6.5. *Частичным подграфом* называется частичный граф некоторого подграфа.

Введем некоторые обозначения, которые в дальнейшем будем активно использовать:

N_i^- - множество дуг, выходящих из вершины i.

N_i^+ - множество дуг, входящих в вершину i.

$|N|$ - число элементов множества N.

$|M|$ - число элементов множества M.

Определение 6.6. *Нуль-графом* будем называть граф, в котором множество дуг - пустое множество.

Определение 6.7. *Петлей* мы будем называть дугу, у которой начало и конец совпадают.

6.1.2. Способы задания графов.

Перечислим важнейшие способы задания графов, и рассмотрим эти способы на примере графа, приведенного на рис. 1.

1. Перечень дуг графа. Для каждой дуги указывается ее начало и конец.

Пример 6.4. Для графа, приведенного на рис. 6.1 это представление будет выглядеть так: (1,2), (1,2), (3,2), (4,5), (4,5), (4,5).

2. Матрица смежностей. Это неотрицательная целочисленная матрица $R=\left(r_{ij}\right)$ размерности $m\times m$, где $r_{ij}=\left|N_i^-\cap N_j^+\right|$, то есть каждый ее элемент r_{ij} равен числу дуг, идущих из вершины i в вершину j, если такие дуги есть, и нулю, если таких дуг нет.

Пример 6.5. В частности для примера на рис. 6.1 матрица смежностей будет следующей:

0	2	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	0	0	3
0	0	0	0	0

3. Матрица инциденций. Матрица $A=\left(a_{ij}\right)$, размерности $m \times n$, где m=|M|, а n=|N|, является матрицей инциденций, если ее элементы вычисляются следующим образом;

$$a_{iu} = \begin{cases} 1, i = i(u) \\ -1, i = j(u) \\ 0, i \neq i(u), i \neq j(u) \end{cases}$$

Пример 6.6. Для примера на рис 6.1 получаем следующую матрицу:

	a	**b**	**c**	**d**	**e**	**f**
1	1	1	0	0	0	0
2	-1	-1	-1	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	1	1
5	0	0	0	-1	-1	-1

Замечание 6.1. При рассмотрении матриц инциденций предполагается, что граф не имеет петель.

4. Сжатая форма матрицы смежностей. Матрица B=$\left(b_{ij}\right)$, размерности $m \times k$, где m=|M|, k=max{|N_i^-|,i=1..k} (максимальное число дуг, выходящих из одной вершины).

Элементами i строки этой матрицы являются номера вершин, являющихся концами дуг, выходящих из вершины i Если вершин нет, то ставятся нули.

Пример 6.7. Для примера на рис. 6.1 получаем следующую матрицу.

2	2	0
0	0	0
2	0	0

5	5	5
0	0	0

Данное представление полезно использовать, когда k<m.

5. Списковая форма представления. Существуют различные списковые формы представления графа. Здесь мы рассмотрим представление графа в виде цепного списка вершин. А для каждой вершины формируется цепной список дуг, выходящих из соответствующей вершины. На рис. 4 приведено списковое представление для рассматриваемого примера. Для каждой вершины формируется цепной список дуг, выходящих из соответствующей вершины, а ссылка на голову каждого такого списка записывается во второе поле информации о вершине (в первом поле записывается номер вершины). Третье поле информации о вершине используется для ссылки на следующую вершину. На рисунке линии указывают, ссылку на какой объект(вершину или дугу) следует записать в соответствующее поле (линия идет от середины поля к началу объекта).

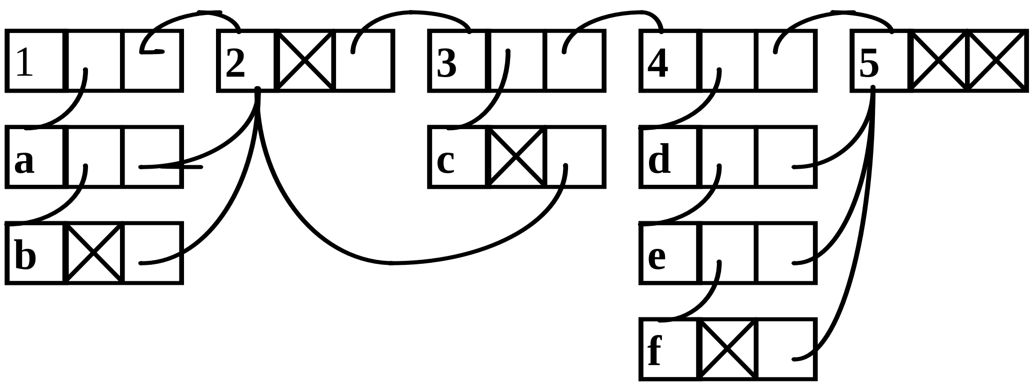


Рис. 6.4

Иногда рассматривают цепной список, сформированный по дугам, входящих в каждую вершину.

6.1.3. Связность.

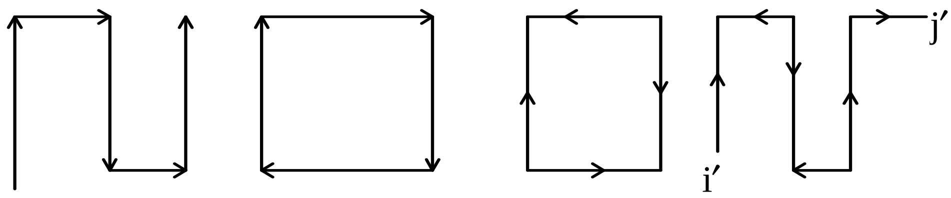
Определение 6.8. Последовательность вершин $i_1, i_2, ..., i_{k+1}$ графа <*M*,*N*,*T*> и последовательность его дуг $u_1, u_2, ..., u_k$ мы будем называть *путем*, если $\forall\ l \in 1,..,k$ выполняются равенства:

$$i(u_l) = i_l,\ j(u_l) = i_{l+1}.$$

Вершина i_1 называется *началом пути*, вершина i_{k+1} - *концом пути*, число k называется *звенностью* пути. Путь, у которого начало и конец совпадают, называется *замкнутым*.

В пути вершины и дуги занумерованы так, что конец очередной дуги является началом следующей. На рис. 6.5 изображен пример пути.

Определение 6.9. Замкнутый путь, у которого все вершины различны мы будем называть *контуром*. На рис. 6.5 изображен пример контура.



Путь Контур Цикл Цепь

Рис. 6.5.

Введем другое определение контура, эквивалентное определению 6.9.

Определение 6.10. Частичный подграф $<M',N',T>$ графа $<M,N,T>$ называется *контуром*, если:

1. $|M'|=|N'|$(число вершин равно числу дуг).

2. $\forall i \in M'$ $\left|N_i^{'-}\right|=\left|N_i^{'+}\right|=1$, т.е. из каждой вершины выходит одна дуга и одна дуга входит.

3. Никакой собственный подграф $<M',N',T>$ этими свойствами не обладает.

Термин «*собственный*» мы будем использовать, чтобы подчеркнуть, что речь идет о подмножестве именно графа $<M',N',T>$, а не графа $<M,N,T>$

Теорема 6.1. *Вершины и дуги контура можно занумеровать так, чтобы они образовывали замкнутый путь.*

Доказательство. Выберем произвольную вершину из M'. Обозначим ее через i_1. В N' (по свойству 2 определения 6.10) имеется единственная дуга, выходящая из i_1, обозначим ее через u_1, а ее второй конец - i_2. Либо i_2 совпадает с началом пути, либо имеется дуга, выходящая из i_2. Обозначим ее через u_2, а ее конец через i_3 и т.д. Так как на каждом шаге этого процесса появляется новая вершина (конец ранее не встречавшейся дуги), на некотором шаге конец пути совпадет с началом. При этом множества встретившихся вершин и дуг будут, очевидно, обладать свойствами 1 и 2 из определения 6.10, а значит, в силу 3 совпадают с M' и N'.

Введем еще ряд определений.

Определение 6.11. Частичный подграф $<M',N',T>$ графа $<M,N,T>$ называется *циклом*, если:

1. $|M'|=|N'|$ (число вершин равно числу дуг).

2. $\forall\ i\in M'$ $\left|N_i^{'-}\right|+\left|N_i^{'+}\right|=2$, т.е. каждой вершине инцидентно две дуги.

3. Никакой собственный подграф $<M',N',T>$ этими свойствами не обладает.

На рис. 6.5 приведен пример цикла.

Определение 6.12. Частичный подграф $<M',N',T>$ графа $<M,N,T>$ называется *цепью*, если:

1. $|N'|=|M'|-1$ (число дуг на единицу меньше числа вершин).

2. $\exists!\ i',j'\in M'$: $\left|N_i^{'-}\right|+\left|N_i^{'+}\right|=1$, при $i=i'$ или $i=j'$, т.е. вершинам i' и j' инцидентна одна дуга.

3. $\forall\ i\in M', i\neq i', i\neq j'$ $\left|N_i^{'-}\right|+\left|N_i^{'+}\right|=2$, т.е. каждой вершине не совпадающей с i' или j', инцидентно две дуги.

Никакой собственный подграф $<M',N',T>$ не содержит цикла.

На рис. 6.5 приведен пример цепи. Приведем еще одно определение цепи, эквивалентное определению 6.12.

Определение 6.13. Последовательность вершин $i_1,i_2,...,i_{k+1}$ графа $<M,N,T>$ и последовательность его дуг $u_1,u_2,...,u_k$ мы будем называть *цепью*, если $\forall$ $l\in 1,..,k$ либо $i(u_l)=i_l$, $j(u_l)=i_{l+1}$, либо $i(u_l)=i_{l+1}$, $j(u_l)=i_l$.

При этом все вершины $i_1,i_2,...,i_{k+1}$ должны быть различны. Дуги, для которых выполняется первое соотношение, называются *положительно ориентированными*, а второе - *отрицательно ориентированными*. Число k называется *звенностью* цепи.

Частичный подграф, состоящий из одной вершины мы тоже будем считать по определению цепью.

Введем также другое определение цикла.

Определение 6.14. *Циклом* называется замкнутая цепь.

Определение 6.15. Граф $<M,N,T>$ называется *связным,* если любые две его вершины можно соединить цепью.

Естественно, граф, сам являющийся цепью, связен. В цепи без нарушения связности нельзя удалить ни одной дуги, и в этом смысле цепь минимальна, в отличии, например, от цикла, где удаление дуги не приводит к нарушению связности. Представляет интерес нахождение такого рода минимальных связных частичных графов в произвольных графах.

Теорема 6.2. *Пусть $<M,N,T>$ - связный граф. В этом графе можно выделить частичный граф $<M,N',T>$ такой, что $|N'|=|M|-1$ и граф $<M,N',T>$ связен.*

В произвольном графе все вершины и дуги разбиваются на компоненты связности.

Определение 6.16. *Компонентой связности* называется максимально связный подграф. (Максимальность понимается в том смысле, что множество вершин в этом подграфе не может быть увеличено без нарушения связности).

Пример 6.8. Например, на рис. 6.1 изображен граф, состоящий из двух компонент связности. К первой относятся вершины 1,2,3 и все инцидентные им дуги, ко второй - относятся вершины 4 и 5 со всеми инцидентными им дугами. Подграф этого графа, состоящий из одной вершины 4 компонентой связности не является, так как он не обладает свойством максимальности (добавление вершины 5 и дуг d,e,f в этот подграф не приводит к нарушению связности).

Рассмотрим следующий вопрос: в каком графе будет минимальное число компонент связности, а в каком - максимальное?

Ответ очевиден: минимальное число компонент связности - одна - будет в связном графе, а максимальное - число вершин - в нуль графе.

Теорема 6.3. *Пусть $i \in M$, M_i - множество вершин, которые можно соединить с вершиной i цепью. Подграф $<M_i,N(M_i),T>$ является компонентой связности.*

Доказательство. Рассмотрим $\forall\ i', j' \in M_i$ и отличные от i. Существуют цепи, соединяющие i с i' и i с j'. Перенумеровав их, получим (выписывая только вершины)

$$i_1 = i, i_2, \ldots, i_k = i'$$
$$j_1 = i, j_2, \ldots, j_l = j'$$

Пусть r - наибольший номер вершины из второй цепи, содержащейся в первой цепи. Пусть p – номер этой вершины в первой цепи. Цепь

$$i' = i_k, i_{k-1}, \ldots, i_p = j_r, j_{r+1}, \ldots, j_l = j'$$

соединяет вершины i' и j' (рис. 6.6).

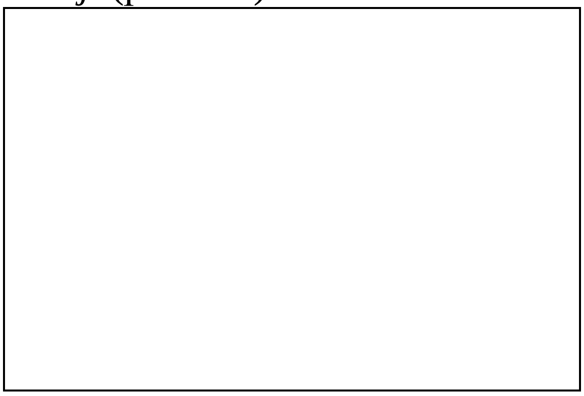


Рис. 6.6

Следующая теорема интересна тем, что связывается чисто алгебраическое свойство матрицы с наглядным геометрическим понятием.

Теорема 6.4. *Ранг матрицы инциденций произвольного графа равен $|M|$-p, где p - число компонент связности графа.*

6.1.4. Деревья

Определение 6.17. *Деревом* называется связный граф, в котором число дуг на единицу меньше числа вершин.

Если вспомнить теорему 6.2, то, учитывая введенное определение, можно ее сформулировать так: в любом связном графе можно выделить частичный граф - дерево. Свойства деревьев важны, поэтому рассмотрим следующую теорему, которая фактически перечисляет важнейшие свойства деревьев.

Теорема 6.5. *Следующие определения дерева эквивалентны:*

связный граф без циклов;

граф без циклов, в котором число дуг на единицу меньше числа вершин;

связный граф, в котором число дуг на единицу меньше числа вершин;

граф без циклов, в котором добавление дуги приводит к появлению цикла;

связный граф, который перестает быть связным после удаления любой дуги;

граф, у которого любые две вершины можно связать и цепью и притом только одной.

Определение 6.18. Дерево называется *иерархическим*, если в нем найдется вершина i_0, из которой существуют пути во все остальные вершины. Вершина i_0, если она существует, определяется этим условием однозначно. Она называется корнем дерева.

6.2. Кратчайшие пути.

6.2.1.Поиск контура в графе.

В начале данного раздела рассмотрим *алгоритм поиска контура в графе*, который носит вспомогательный характер.

Пусть имеется ориентированный граф <*M*,*N*,*T*>. Требуется проверить, есть ли в этом графе хотя бы один контур, или контуров в графе нет?

Введем дополнительно вектор *v* размерности m+1, в который будем записывать номера пройденных вершин (предполагаем, что вершины занумерованы от 1 до m).

Берем вершину 1. Пусть из нее выходит дуга u_1. Пусть $j(u_1) = i_1$. Если $i_1 \neq 1$, то рассматриваем дугу, выходящую из i_1, пусть это будет дуга u_2(иначе - петля, которую можно рассматривать как контур). Пусть $j(u_2) = i_2$. Если $i_2 = 1$, или $i_2 = i_1$, то контур найден; иначе продолжаем этот процесс, записывая в вектор *v* номера вершин $1, i_1, i_2, ...$. Здесь возможны два варианта:

1. при проверке выяснилось, что номер очередной вершины, которую нужно записывать в вектор *v*, уже встречался ранее в векторе *v*. В этом случае контур найден.

2. Из очередной вершины i_k нет выходящих дуг (тупик). Тогда возвращаемся на шаг назад в вершину i_{k-1}, выбрасываем из графа дугу (i_{k-1}, i_k), удаляем вершину i_k из вектора v и продолжаем этот процесс.

В результате, либо контур будет найден, и алгоритм закончит свою работу, либо мы вернемся в вершину 1. Во втором случае берем вершину 2 и повторяем для нее все сначала, и т.д. до тех пор пока не рассмотрим все вершины. В результате, либо мы получим контур, при этом номера вершин контура содержатся в векторе v , либо все дуги графа будут удалены, и мы получим нуль-граф.

Пример 6.9. Пусть граф задан в виде перечня дуг (1,2), (1,3), (2,3), (2,4), (3,4), (4,5), (6,5), (6,7), (7,8) (8,6). Проверим, есть ли в графе хотя бы один контур.

Решение. Берем первую вершину, записываем ее в массив v. Из вершины 1 идет дуга в 2. Так как $2\neq1$, то записываем 2 в массив v и продолжаем процесс. Из 2 идет дуга в 3: аналогично, $3\neq1$, $3\neq2$, и записываем 2 в массив v и т.д. Из 3 идет дуга в 4, $4\neq1, 4\neq2, 4\neq3$; из 4 идет дуга в 5, $5\neq1, 5\neq2, 5\neq3, 5\neq4$. Но из 5 не выходит дуг, и мы попали в тупик. Выбрасываем дугу (4,5), так как по ней мы попали в тупик, удаляем вершину 5 из вектора v. Граф будет теперь иметь вид (1,2), (1,3), (2,3), (2,4), (3,4), (6,5), (6,7), (7,8), (8,6), а вектор v - 1,2,3,4.

Но теперь вершина 4 - тупик. В этот тупик мы попали по дуге (3,4) (последние две вершины в векторе v задают дугу, по которой мы пришли в последнюю вершину). Удаляем дугу (3,4) из графа, а вершину 4 из вектора v. Приведем теперь наш граф и вектор v: (1,2), (1,3), (2,3), (2,4), (6,5), (6,7), (7,8), (8,6), v=1,2,3.

Теперь вершина 3 является тупиком. Удаляем дугу (2,3), по которой мы попали в тупик из графа, а вершину 3 из вектора v, получаем: (1,2),(1,3),(2,4),(6,5),(6,7),(7,8)(8,6); v=1,2.

Далее - из 2 идем в 4 по дуге (2,4), 4 записываем в v: $4\neq1$, $4\neq2$. Но в 4 - тупик. Удаляем дугу (2,4) из графа, а вершину 4 - из v. Получаем граф (1,2),(1,3),(6,5),(6,7),(7,8)(8,6) и v=1,2.

Теперь вершина 2 - тупик. Выбрасываем дугу (1,2) из графа, а 2 из вектора v. Получаем: (1,3),(6,5),(6,7),(7,8)(8,6), v=1

Далее, из 1 идем в 3, 3 - тупик. Удаляем (1,3) из графа, а 3 из v. Получаем: (6,5),(6,7),(7,8)(8,6), v=1

Из 1 нет выходящих дуг. Выбрасываем 1 из *v*. Записываем туда 2. Из 2 нет выходящих дуг. Выбрасываем 2 из *v*. Записываем туда 3 и т.д. Наконец записываем в вектор *v* вершину 6. Из 6 идет дуга в 5 и 5≠6. Но 5 - тупик. Выбрасываем дугу (6,5), а вершину 5 из вектора *v*, получаем: (6,7),(7,8)(8,6), *v*=6.

Из 6 идет дуга в 7, 7≠6. Записываем 7 в *v*. Из 7 идет дуга в 8, 8≠6, 8≠7. Из 8 идет дуга в 6. Записываем 6 в *v*. Теперь граф и вектор *v* будут такими: (6,7),(7,8)(8,6);*v*=6,7,8,6

Так как в векторе *v* встретилась вершина 6, которая была уже ранее, то в графе есть контур: 6→7→8→6.

6.2.2. Дерево кратчайших путей. Алгоритм Дейкстры

Постановка задачи о поиске кратчайшего пути заключается в следующем.

Пусть имеется *m* городов. Известны расстояния между ними, причем сумма расстояний из одного города в другой через промежуточный город может быть меньше прямого пути (может не выполняться неравенство треугольника). Фиксируется некоторый город. Необходимо найти кратчайшие пути от этого города до всех городов, до которых эти пути существуют.

Рассмотрим более строгую математическую постановку задачи. Рассматривается ориентированный граф $<M,N,T>$. Для каждой дуги $u\in N$ задается ее длина - $c[u]\geq 0$. Под *длиной пути* мы будем понимать сумму длин входящих в него дуг. Фиксируется некоторая вершина $i_0\in M$. Требуется найти кратчайшие пути от вершины i_0 до остальных вершин графа.

Имеются различные алгоритмы решения данной задачи. Наиболее эффективным считается алгоритм Дейкстры. Согласно этому алгоритму, множество вершин *M* разбивается на два непересекающихся подмножества: M_0 – вершины, кратчайшие расстояния до которых уже вычислены, и M_1 – остальные вершины. В начальный момент $M_0=\{i_0\}, M_1=M\setminus\{i_0\}$. В векторе *f* размерности $m=|M|$ будем хранить длины кратчайших путей от вершины i_0 до соответствующей вершины. Для того чтобы можно было найти пути в любую вершину, заведем еще один вектор размерности *m*, который обозначим через *v*. Здесь мы будем для каждой вершины запоминать номер вершины, из которой мы пришли в данную. Обозначим через i_1 текущую вершину. В начальный момент $i_1 = i_0$.

Пример 6.10. Рассмотрим пример, изображенный на рис. 6.7. Пусть в качестве i_0 выступает вершина 1. Требуется найти кратчайшие пути от 1 до остальных вершин (2,3,4 и 5).

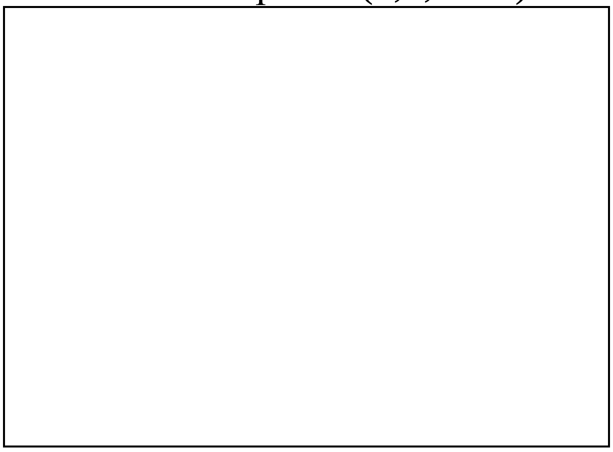


Рис. 6.7

Решение. Начинаем с 0 шага: M_0={1};M_1={2,3,4,5};i_1=1. Формируем векторы f и v: f[0,+∞,+∞,+∞,+∞]; v[0,0,0,0,0].

1-я итерация: просматриваем дуги, выходящие из текущей (в данном случае 1) вершины. Это дуги (1,2), (1,3) и (1,4). Длина пути до текущей вершины (1) от начальной вершины i_0=1 равна 0. Добавляем длину дуги (1,2), равную 6. Получаем 0+6=6, а старое значение f[2]=+∞. Новое значение меньше, чем старое, поэтому делаем пересчет: f[2]:=6;v[2]:=1 (так как в вершину 2 мы пришли из вершины 1). Аналогично считаем для концов дуг (1,3) и (1,4). Получаем, что f[0,6,2,3,+∞]; v[0,1,1,1,0].

Множество M_1≠∅, поэтому находим новую текущую вершин, то есть среди вершин из множества M_1, то есть (2,3,4,5), ищем такую, у которой значение f минимально. Очевидно, таким свойством обладает вершина 3. Она становится текущей, заносится в множество M_0 и исключается из множества M_1: i_1:=3; M_0={1,3};M_1={2,4,5}.

Заметим, что f[3]=2, то есть длина пути от начальной вершины до текущей теперь равна 2.

2-я итерация: просматриваем дуги, выходящие из вершины 3: (3,2),(3,4) и (3,5) и пересчитываем значения *f* и *v* концов этих дуг, как это было показано на 1 итерации: *f*[0,3,2,3,9];*v*[0,3,1,1,3].

Заметим, что *v*[4] осталось равным 1, так как *f*[4] не пересчитывалось, поскольку длина нового пути (2+5=7) оказалась больше старого (3).

Множество $M_1 \neq \varnothing$, поэтому находим новую текущую вершину. Теперь это будет либо вершина 2, либо вершина 4, так как минимальное значение f среди вершин из множества M_1 (2,4,5) достигается сразу в двух вершинах 2 и 4, и равно 3. Возьмем меньшую по номеру: i_1:=2; $M_0=\{1,3,2\}; M_1=\{4,5\}$.

3-я итерация: *f*[2]=3. Пересчитываем *f* и *v* так же, как и ранее: *f*[0,3,2,3,5]; *v*[0,3,1,1,2].

Множество $M_1 \neq \varnothing$, поэтому находим новую текущую вершину. Теперь это вершина 4: i_1:=4; $M_0=\{1,3,2,4\}; M_1=\{5\}$.

4-я итерация: *f*[4]=3. *f* и *v* на этой итерации не меняются, так как 3+9=12>5. Множество $M_1 \neq \varnothing$, новая текущая вершина теперь 5, так как других нет.

5-я итерация. *f* и *v* на этой итерации не меняются, так как из 5 вершины не выходят дуги. Множество $M_1 = \varnothing$.

Итак, получили итоговые векторы *f* и *v*: *f*[0,3,2,3,5]; *v*[0,3,1,1,2].

Рассмотрим, как получить дерево кратчайших путей с помощью вектора *v*. Начнем с вершины 5. По вектору *v* определяем, что в 5 мы попадаем из 2 (*v*[5]=2). 2→5. В 2 мы попадаем из 3 (*v*[2]=3). 3→2→5. В 3 мы попадаем из 1 (*v*[3]=1). 1→3→2→5. Но *v*[1]=0, значит 1 - начальная вершина. Осталась вершина 4, которая не попала в рассмотренный путь. *v*[4]=1, поэтому кратчайший путь в 4 идет из 1. В итоге получилось дерево кратчайших путей, изображенное на рис.8 .

Можно было построить это дерево по вектору *v* по-другому. Начнем с вершины 1. По вектору *v* определяем, что из вершины 1 нужно идти в вершины 3 и 4 (1 стоит в векторе *v* на 3 и 4 местах). Далее - из вершины 3 идем в 2 (3 стоит в векторе *v* на 2 позиции). Из 2 идем в 5 (2 стоит в *v* на 5 позиции). В итоге снова получается дерево кратчайших путей, изображенное на рис. 6.8.

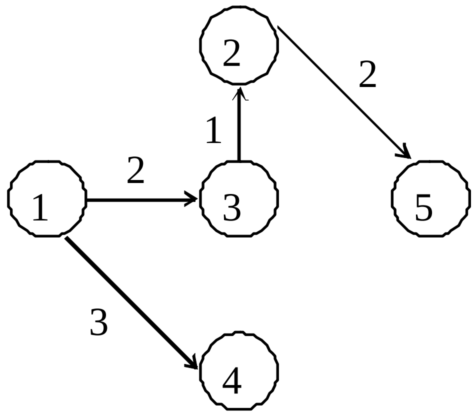


Рис. 6.8

Если не существует пути из вершины i_0 в некоторые вершины, то на соответствующих местах в векторе f останутся значение $+\infty$, а в векторе v - 0. Таким образом, алгоритм Дейкстры позволяет найти дерево кратчайших путей для вершин, до которых существуют пути из вершины i_0. В это дерево могут входить не все вершины графа.

Очевидно, что приведенный алгоритм конечен (на каждой итерации одна вершины удаляется из множества M_1 и заносится в множество M_0, а M_1 - конечное множество). Более того, ясно, что требуется всего лишь m итераций, если считать нулевой шаг за итерацию.

6.2.3. Матрица кратчайших расстояний

Рассмотрим обобщение предыдущей задачи.

Пусть, как и раньше, имеется m городов. Известны расстояния между ними, причем сумма расстояний из одного города в другой через промежуточный город может быть меньше прямого пути (может не выполняться неравенство треугольника). Необходимо найти кратчайшие пути от любого города до любого другого города, до которого эти пути существуют.

Рассмотрим более строгую математическую постановку задачи.

Имеется ориентированный граф $<M,N,T>$. Для каждой дуги $u \in N$ задается ее длина - $c[u] \geq 0$. Под *длиной пути* мы будем понимать сумму длин входящих в него дуг. Требуется найти кратчайшие пути от любой вершины до любой другой вершины графа.

По графу мы можем сформировать матрицу непосредственных переходов (кратчайших переходов за один шаг). Обозначим эту матрицу через F^1.

$$F_{ij}^1 = \begin{cases} 0, \ \ i = j, \\ \min\limits_{u \in N_i^- \bigcap N_j^+} c[u], \ \ N_i^- \bigcap N_j^+ \neq \varnothing, \\ +\infty, \ \ N_i^- \bigcap N_j^+ = \varnothing. \end{cases}$$

То есть считается, что длина пути из города в тот же город равна 0. Если из города i в город j имеется несколько дорог, то среди них выбирается дорога минимальной длины. А если прямого пути из города i в город j нет, то ставим на соответствующее место матрицы $+\infty$. Теперь рассмотрим алгоритмы, решающие нашу задачу.

6.2.4. Алгоритм Беллмана

Р. Беллман предложил вычислять кратчайшие пути по шагам. Матрица кратчайших переходов за два шага вычисляется следующим образом. Для любых городов i и j, чтобы вычислить кратчайший переход за 2 шага рассматривают все переходы из i в j через все промежуточные вершины. Очевидно, что минимальный из таких переходов (а прямой переход тоже учитывается) и будет кратчайшим переходом из i в j за два шага и менее (менее, если не было пересчета, когда прямой переход короче всех остальных). Вновь полученная таким образом матрица и будет матрицей кратчайших переходов за два шага (и менее). Используя ее и матрицу кратчайших переходов за один шаг можно получить матрицу кратчайших переходов за 3 шага (и менее). Введем следующие обозначения F^a - матрица кратчайших переходов за a шагов. Легко доказать следующую рекуррентную формулу Беллмана:

$$F_{ij}^{a+b} = \min_{k \in M} \{F_{ik}^a + F_{kj}^b\}$$

То есть, чтобы вычислить элемент i, j в матрице кратчайших переходов за $a+b$ шагов, нужно рассмотреть все промежуточные вершины k, и, взяв переход из i в k за a шагов, добавить переход из k в j за b шагов. Заметим, что при $k=i$ и $k=j$ получаются прямые переходы из i в j за b шагов и за a шагов соответственно, поскольку $F_{ii}^a = 0$ и $F_{jj}^b = 0$. Таким образом, имея рекуррентную формулу Беллмана, можно вычислить матрицу кратчайших переходов за любое количество шагов. Беллман предложил такую вычислительную схему:

$$F^1, F^2, F^3, ..., F^{m-1}$$

Очевидно, что дальше вычислять не имеет смысла, так как начиная с F^{m-1}, где $m=|M|$, матрицы меняться не будут, поскольку максимальное число вершин в пути - m, а звеньев - $m-1$.

Оценим трудоемкость такой вычислительной схемы. Для вычисления всех элементов матрицы требуется порядка m^2 операций (двойной вложенный цикл). Для вычисления матрицы F^a по формуле Беллмана требуется еще один внешний цикл, а значит, понадобится порядка m^3 операций. Для реализации вычислительной схемы Беллмана потребуется еще один внешний цикл, а значит потребуется порядка m^4 операций. Нельзя ли улучшить указанную схему?

Шимбел предложил другую вычислительную схему, реализующую рекуррентную формулу Беллмана:

$$F^1, F^2, F^4, F^8 ..., F^l$$

Вычисления проводятся по степеням двойки до тех пор, пока не будет выполнено неравенство $l \geq m-1$. Трудоемкость такой вычислительной схемы оценивается порядком $m^3 \times \log_2 m$ операций (следует округлить второй сомножитель до ближайшего целого в большую сторону).

В результате работы рассмотренных алгоритмов можно получить итоговую матрицу, где указаны длины кратчайших переходов из любой вершины в любую другую. Однако, часто требуется указать и сами пути. Для этого необходима еще одна матрица, обозначим ее через G. Каждый ее элемент G_{ij} указывает ближайшую вершину, через которую сделан переход. В начальный момент следует сформировать матрицу *G* следующим образом (для удобства будем нумеровать матрицу индексом соответствующей матрицы *F*): $G_{ij}^{1} = j, i \in 1..m$, поскольку в начальный момент ближайшей вершиной, через которую сделан переход является она сама. В дальнейшем пересчет элементов в матрице *G* ведется параллельно матрице *F*: если пересчитывается элемент матрицы *F*, то пересчитывается элемент матрицы *G*.

Пусть F_{ik}^{a} - элемент матрицы кратчайших переходов за *a* шагов, F_{kj}^{b} - элемент матрицы переходов за *b* шагов, F_{ij}^{a+b} - матрица кратчайших переходов за $a+b$ шагов, $G_{ik}^{a}, G_{kj}^{b}, G_{ij}^{a+b}$ - соответствующие элементы соответствующих матриц . Пересчет идет следующим образом: если $F_{ik}^{a} + F_{kj}^{b} < F_{ij}^{a}$, то $F_{ij}^{a+b} := F_{ik}^{a} + F_{kj}^{b}; G_{ij}^{a+b} := G_{ik}^{a}$.

Пример 6.11. Рассмотрим пример на рис. 6.9.

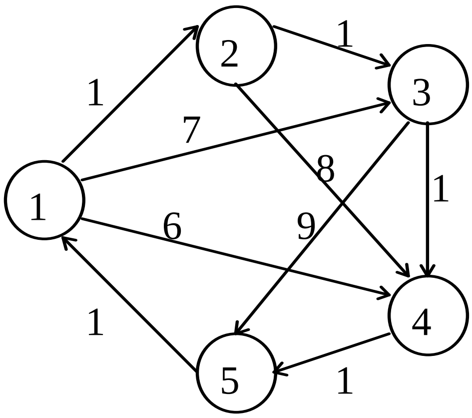


Рис. 6.9

Сформируем матрицы $F^1 = \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 7 & 6 & \infty \\ \hline \infty & 0 & 1 & 8 & \infty \\ \hline \infty & \infty & 0 & 1 & 9 \\ \hline \infty & \infty & \infty & 0 & 1 \\ \hline 1 & \infty & \infty & \infty & 0 \\ \hline \end{array}$ и

$G^1 = \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$ по данному графу. При этом ∞ в матрице F^1 означает, что прямого пути нет. Далее, по матрицам F^1и G^1 получаем F^2 и G^2. Ясно, что $F_{11}^2 = F_{11}^1 = 0$. Аналогично $F_{12}^2 = F_{12}^1 = 1$, поскольку эти элементы не могут быть уменьшены, какие бы промежуточные вершины мы не взяли. Соответствующие элементы матрицы G тоже не изменились. Рассмотрим подробно процесс вычисления элемента F_{13}^2. Чтобы вычислить этот элемент рассмотрим все промежуточные вершины:

$$1 \xrightarrow{0} 1 \xrightarrow{7} 3 \ (7)$$

$$1 \xrightarrow{1} 2 \xrightarrow{1} 3 \ (2)$$

$$1 \xrightarrow{7} 3 \xrightarrow{0} 3 \ (7)$$

$$1 \xrightarrow{6} 4 \xrightarrow{\infty} 3 \ (\infty)$$

$$1 \xrightarrow{\infty} 5 \xrightarrow{\infty} 3 \ (\infty)$$

Кратчайший путь из 1 в 3, очевидно, равен 2. Ближайшая промежуточная вершина - 2 (так как G_{12}^1=2). Поэтому F_{13}^2=2, G_{13}^2=2. Заметим, что переходы $1 \xrightarrow{0} 1 \xrightarrow{7} 3\ (7)$ и $1 \xrightarrow{7} 3 \xrightarrow{0} 3\ (7)$ фактически являются прямым переходом $1 \rightarrow 3$. Аналогично вычисляем элементы F_{14}^2 и G_{14}^2. Выясняется, что они равны F_{14}^1 и G_{14}^1 соответственно, то есть не изменились. Далее вычисляем элементы F_{15}^2 и G_{15}^2. Приведем вычисления:

$$1 \xrightarrow{\infty} 5(\infty)$$

$$1 \xrightarrow{1} 2 \xrightarrow{\infty} 5\ (\infty)$$

$$1 \xrightarrow{7} 3 \xrightarrow{9} 5\ (16)$$

$$1 \xrightarrow{6} 4 \xrightarrow{1} 5\ (7)$$

Ясно, что теперь F_{15}^2=7, а $G_{15}^2 = G_{14}^1$=4.

Теперь начинаем вычислять элементы второй строки матриц F^2 и G^2. Элементы F_{21}^2 и G_{21}^2 переносятся из F^1и G^1 соответственно, что легко проверяется. Аналогично поступаем и с элементами F_{22}^2 и G_{22}^2, а также с элементами F_{23}^2 и G_{23}^2. Все они переносятся из F^1и G^1 соответственно. Вычисляем элементы F_{24}^2 и G_{24}^2. Будем оптимизировать вычисления. Видим, что минимальный элемент (кроме 0) во второй строке матрицы F^1 - это $F_{23}^1 = 1$. Соответственно рассматриваем вариант

$$2 \xrightarrow{1} 3 \xrightarrow{1} 4\ (2)$$

Очевидно, что все остальные промежуточные вершины дают худший результат (первый переход у всех них больше двух). Поэтому F_{24}^2=2, $G_{24}^2 = G_{23}^1$. Вычисляем последние элементы во второй строке F_{25}^2 и G_{25}^2. Здесь надо просматривать 2 варианта:

$$2 \xrightarrow{1} 3 \xrightarrow{9} 5\ (10)$$

$$2 \xrightarrow{8} 4 \xrightarrow{1} 5\ (9)$$

Следовательно F_{25}^2=9, $G_{25}^2 = G_{24}^1$=4.

Аналогично проводим вычисления 3, 4 и 5 строк матриц F^2 и G^2. Получаем:

$$F^2 = \begin{array}{|c|c|c|c|c|}\hline 0 & 1 & 7 & 6 & 7 \\ \hline \infty & 0 & 1 & 2 & 9 \\ \hline 10 & \infty & 0 & 1 & 2 \\ \hline 2 & \infty & \infty & 0 & 1 \\ \hline 1 & 2 & 8 & 7 & 0 \\ \hline \end{array} \text{ и } G^2 = \begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 2 & 4 & 4 \\ \hline 1 & 2 & 3 & 3 & 5 \\ \hline 5 & 2 & 3 & 4 & 4 \\ \hline 5 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 1 & 5 \\ \hline \end{array}.$$

Если дальнейшие вычисления вести по схеме Беллмана, то нужно вычислять F^3 и G^3. Рассмотрим вычисление F_{14}^3 и G_{14}^3, поскольку предыдущие элементы переносятся без изменений из матриц F^2 и G^2 соответственно.

$$1 \xrightarrow{6} 4 \ (6)$$
$$1 \xrightarrow{1} 2 \xrightarrow{8} 4 \ (9)$$
$$1 \xrightarrow{2} 3 \xrightarrow{1} 4 \ (3)$$
$$1 \xrightarrow{7} 5 \xrightarrow{\infty} 4 \ (\infty)$$

Первый переход берется по матрице F^2, а второй - по матрице F^1(можно делать и в другом порядке, первый – по F^1, а второй - по F^2). Следовательно F_{14}^3=3, $G_{14}^3 = G_{13}^2 = 2$. Заметим, что ближайшая вершина, через которую сделан этот переход - 2, так как переход 1→3 длиной 2 есть результат вычисления на предыдущей итерации, что и записано в элементе G_{13}^2. Остальные элементы матриц F^3 и G^3 вычисляются аналогично. Однако, мы не будем так делать, поскольку можно воспользоваться вычислительной схемой Шимбела и сразу вычислить F^4 и G^4, что короче на одну итерацию.

Также как и ранее начинаем вычисления с F_{14}^4 и G_{14}^4, поскольку предыдущие элементы переносятся без изменений из матриц F^2 и G^2 соответственно.

$$1 \xrightarrow{5} 4 \ (6)$$
$$1 \xrightarrow{1} 2 \xrightarrow{2} 4 \ (3)$$
$$1 \xrightarrow{2} 3 \xrightarrow{1} 4 \ (3)$$
$$1 \xrightarrow{7} 5 \xrightarrow{8} 4 \ (15)$$

Здесь оба перехода берутся уже из матрицы F^2. Видим, что минимальное значение равно 3 и достигается в двух случаях. В принципе можно брать любое. Элемент F_{14}^4=3 и в том, и в другом случае. Но так как $G_{13}^2 = G_{12}^2 = 2$, то и G_{14}^4=2, а это означает, что независимо от выбора, ближайшей вершиной, через которую сделан переход, все равно будет 2 (в действительности двух разных путей здесь нет, есть один, найденный двумя способами. В первом случае сначала делаем 1 шаг длиной 1, а потом сразу 2 шага по единице (1+1). Во втором случае наоборот, сначала делаем 2 шага по единице (1+1), а потом один длиной 1. На рис. 9 этот путь длиной 3 легко находится). Теперь вычисляем элементы F_{15}^4 и G_{15}^4.

$$1 \xrightarrow{7} 5 \ (7)$$

$$1 \xrightarrow{1} 2 \xrightarrow{9} 5 \ (10)$$

$$1 \xrightarrow{2} 3 \xrightarrow{2} 5 \ (4)$$

Остальные варианты не просматриваем, так как первый переход у них больше, чем 4. Итак, F_{15}^4=4, $G_{15}^4 = G_{13}^2$=2. Далее аналогично вычисляем строки 2,3,4 и 5 матриц. Получаем:

$$F^4 = \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 \\ \hline 4 & 0 & 1 & 2 & 3 \\ \hline 3 & 4 & 0 & 1 & 2 \\ \hline 2 & 3 & 4 & 0 & 1 \\ \hline 1 & 2 & 3 & 4 & 0 \\ \hline \end{array} \text{ и } G^4 = \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 2 & 2 & 2 \\ \hline 3 & 2 & 3 & 3 & 3 \\ \hline 4 & 4 & 3 & 4 & 4 \\ \hline 5 & 5 & 5 & 4 & 5 \\ \hline 1 & 1 & 1 & 1 & 5 \\ \hline \end{array}.$$

Итак, получалась итоговая матрица кратчайших переходов F^4. Ей соответствует матрица ближайших вершин, через которые эти переходы сделаны - G^4. Рассмотрим, как найти произвольный путь по матрице G^4. Пусть, например, нам требуется найти кратчайший путь из города 3 в город 2. По матрице F^4 находим длину этого пути F_{32}^4=4. Чтобы найти сам путь, смотрим соответствующий элемент в матрице G^4 - G_{32}^4=4, то есть ближайшей вершиной, в этом пути является город 4. Получаем: 3→4→2. Теперь смотрим элемент G_{42}^4=5. То есть для 4 города ближайший будет 5: 3→4→5→2. Смотрим G_{52}^4=1. Для города 5 ближайшим будет 1: 3→4→5→1→2. Далее G_{12}^4=2. Процесс закончен. Таким образом нашли кратчайший путь:

$$3 \xrightarrow{1} 4 \xrightarrow{1} 5 \xrightarrow{1} 1 \xrightarrow{1} 2 \ (4)$$

Отметим, что имеется более эффективный алгоритм, который требует всего m^3 операций. Этот алгоритм принадлежит Флойду.

6.2.5. Алгоритм Флойда

Этот алгоритм состоит из $m=|M|$ шагов, выполняемых последовательно для каждой вершины $k \in M$. Эта вершина рассматривается как промежуточная. Если прямой переход больше, чем переход через промежуточную вершину k, то он изменяется, иначе - остается прежнее значение. В качестве промежуточных последовательно просматриваются все вершины от 1 до m. В результате получится матрица кратчайших расстояний. Обозначим через F матрицу расстояний, которая первоначально совпадает с F^1, и через G - матрицу ближайших переходов, которая первоначально совпадает с G^1.

Теорема 5.6. Матрица F, полученная в результате работы алгоритма Флойда, является матрицей кратчайших расстояний.

Пример 6.12. Рассмотрим тот же пример, который мы рассматривали для иллюстрации работы алгоритма Беллмана. Исходные матрицы F и G будут теми же

$$F = \begin{array}{|c|cccc|}\hline 0 & 1 & 7 & 6 & \infty \\ \hline \infty & 0 & 1 & 8 & \infty \\ \infty & \infty & 0 & 1 & 9 \\ \infty & \infty & \infty & 0 & 1 \\ 1 & \infty & \infty & \infty & 0 \\ \hline \end{array}, \quad G = \begin{array}{|c|cccc|}\hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

Пусть k=1 (первая вершина рассматривается как промежуточная). Очевидно, что в этом случае первая строка и первый столбец не изменятся (они в исходных матрицах F и G выделены), так как первая строка - переходы из первой вершины, а первый столбец - переходы в первую вершину. Остальные элементы F_{ij} в матрице F вычисляются так: складываются элемент в выделенном (первом) столбце в строке i и элемент в выделенной (первой) строке в столбце j, и сумма сравнивается со значением F_{ij}. Если новое значение меньше, то происходит пересчет, то есть $F_{ij}=F_{i1}+F_{1j}$, $G_{ij}=G_{i1}$.

Итак, начинаем расчеты. Так как F_{21}=∞, то вторая строка как в матрице F, так и в матрице G, изменяться не будет. Аналогично, так как F_{31}=∞ и F_{41}=∞, то и третьи и четвертые строки матриц F и G также не изменяются. Вычисляем пятую строку. Так как $F_{51}+F_{12}$=2<F_{52}=∞, то F_{52}=2, $G_{52}=G_{51}$=1. Аналогично вычисляем F_{53} и F_{54}. Так как $F_{51}+F_{13}$=8<F_{53}=∞, то F_{53}=8 $G_{53}=G_{51}$=1. $F_{54}=F_{51}+F_{14}$(1+6), $G_{54}=G_{51}$=1. В результате имеем

$$F=\begin{array}{c|c|ccc}
0 & 1 & 7 & 6 & \infty \\ \hline
\infty & 0 & 1 & 8 & \infty \\ \hline
\infty & \infty & 0 & 1 & 9 \\
\infty & \infty & \infty & 0 & 1 \\
1 & 2 & 8 & 7 & 0
\end{array},\quad
G=\begin{array}{c|c|ccc}
1 & 2 & 3 & 4 & 5 \\ \hline
1 & 2 & 3 & 4 & 5 \\ \hline
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 1 & 1 & 1 & 5
\end{array}$$

Теперь рассмотрим k=2 (вторая вершина теперь промежуточная). Вторая строка и второй столбец в матрицах F и G переписываются без изменений (они выделены). Остальные элементы F_{ij} в матрице F вычисляются так же, как и для k=1: складываются элемент в выделенном (первом) столбце в строке i и элемент в выделенной (первой) строке в столбце j и сумма сравнивается со значением F_{ij}. Если новое значение меньше, то происходит пересчет, то есть $F_{ij}:=F_{i2}+F_{2j}; G_{ij}:=G_{i2}$. Получаем

$$F=\begin{array}{cc|c|cc}
0 & 1 & 2 & 6 & \infty \\
\infty & 0 & 1 & 8 & \infty \\ \hline
\infty & \infty & 0 & 1 & 9 \\ \hline
\infty & \infty & \infty & 0 & 1 \\
1 & 2 & 3 & 7 & 0
\end{array},\quad
G=\begin{array}{cc|c|cc}
1 & 2 & 2 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\ \hline
1 & 2 & 3 & 4 & 5 \\ \hline
1 & 2 & 3 & 4 & 5 \\
1 & 1 & 1 & 1 & 5
\end{array}$$

Продолжим расчеты без подробных пояснений, так как дальнейшие итерации выполняются аналогично. Итак, k=3 - третья вершина теперь промежуточная. Третья строка и третий столбец (они выделены) в матрицах F и G переписываются без изменений. Получаем:

$$F=\begin{array}{ccc|c|c} 0 & 1 & 2 & 3 & 11 \\ \infty & 0 & 1 & 2 & 10 \\ \infty & \infty & 1 & 2 & 9 \\ \hline \infty & \infty & \infty & 0 & 1 \\ \hline 1 & 2 & 3 & 4 & 0 \end{array},\ G=\begin{array}{ccc|c|c} 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 1 & 5 \end{array}$$

k=4. Четвертая вершина теперь промежуточная:

$$F=\begin{array}{cccc|c|} 0 & 1 & 2 & 3 & 4 \\ \infty & 0 & 1 & 2 & 3 \\ \infty & \infty & 0 & 1 & 2 \\ \infty & \infty & \infty & 0 & 1 \\ \hline 1 & 2 & 3 & 4 & 0 \\ \hline \end{array},\ G=\begin{array}{cccc|c|} 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 1 & 5 \\ \hline \end{array}$$

И наконец, последняя, 5 вершина рассматривается как промежуточная. Пятая строка и пятый столбец (они выделены) переписываются без изменений. Получаем итоговые матрицы:

$$F=\begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 4 \\ \hline 4 & 0 & 1 & 2 & 3 \\ \hline 3 & 4 & 0 & 1 & 2 \\ \hline 2 & 3 & 4 & 0 & 1 \\ \hline 1 & 2 & 3 & 4 & 0 \\ \hline \end{array},\qquad G=\begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 2 & 2 & 2 \\ \hline 3 & 2 & 3 & 3 & 3 \\ \hline 4 & 4 & 3 & 4 & 4 \\ \hline 5 & 5 & 5 & 4 & 5 \\ \hline 1 & 1 & 1 & 1 & 5 \\ \hline \end{array}.$$

Отметим, что эти матрицы совпадают с матрицами, полученными алгоритмом Беллмана. И если матрицы F всегда совпадают, так как это матрицы кратчайших путей, то матрицы G могут различаться, если существуют несколько кратчайших путей одинаковой длины (мы рассматриваем только один из таких путей).

6.2.4.Кратчайшее дерево. Алгоритм Прима-Краскала

Рассмотрим некоторую страну, в которой m городов. Известны расстояния между городами. Требуется построить наземные скоростные линии связи так, чтобы любые два города были соединены друг с другом и при этом суммарная длина всех построенный линий была минимальна. Узлы этих линий могут находиться только в городах.

Математическая постановка задачи формулируется следующим образом. Пусть задан связный граф $<M,N,T>$. Для каждой дуги $u \in N$ задается длина дуги $c[u] \geq 0$. Требуется выделить частичный граф $<M,N',T>$ так, чтобы этот частичный граф был деревом, а $\sum_{u \in N'} c[u]$ была минимальна. Для решения этой задачи существует простой и эффективный алгоритм Прима-Краскала последовательного построения дерева.

Первоначально выбираем произвольную вершину $i_0 \in M$ и образуем $M_0 = \{i_0\}$, $M_1 = M \setminus M_0$. Вершина i_0 заносится в строящееся дерево. На каждом шаге, пока $M_1 \neq \varnothing$, из всех дуг, начала и концы которых принадлежат разным множествам, выбирается дуга минимальной длины. Эта дуга включается в строящееся дерево, а ее второй конец - в множество M_0 и исключается из множества M_1.

По окончании работы алгоритма (а требуется всего m-1 итерация) получается граф $<M,N',T>$, который по построению является деревом (N' состоит из дуг, которые мы включали на каждой итерации), поскольку он связен, а соотношение, что число дуг на единицу меньше числа вершин, поддерживается на каждой итерации. Можно доказать следующую теорему.

Теорема 6.7. *Дерево, полученное в результате работы алгоритма Прима-Краскала, является кратчайшим деревом.*

Для ускорения работы алгоритма и уменьшения просмотров следует упорядочить все дуги графа по возрастанию перед началом работы. На каждой итерации отыскивается первая дуга в этом списке, начало и конец которой принадлежат разным множествам. Эта дуга, очевидно, и будет искомой. Кроме того, при просмотре списка, следует удалить из него все дуги, начало и концы которых попали в множество M_0.

Пример 6.13. Рассмотрим пример, изображенный на рис. 6.10

Рис. 6.10

Упорядочим список дуг по возрастанию:

$$\overset{1}{(1,2)},\overset{1}{(1,3)},\overset{1}{(2,3)},\overset{2}{(3,4)},\overset{3}{(4,5)},\overset{4}{(5,1)},\overset{6}{(1,4)},\overset{8}{(2,4)},\overset{9}{(3,5)}$$

Берем произвольную вершину. Пусть это будет вершина 2. Формируем множества M_0={2}, M_1={1,3,4,5}. Включаем вершину 2 в строящееся дерево. Итак, в начальный момент в дереве одна вершина 2 и нет ни одной дуги. Просматриваем упорядоченный список дуг. Начало и конец уже первой дуги (1,2) принадлежат разным множествам. Поэтому эту дугу заносим в строящееся дерево (рис. 6.11), а ее второй конец (вершина 1) заносится в множество M_0 и исключается из множества M_1. Имеем: M_0={2,1}, M_1={3,4,5}

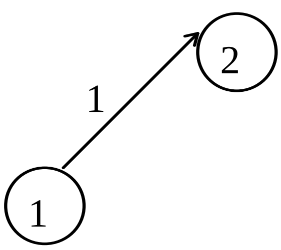


Рис. 6.11

Так как начало и конец дуги (1,2) принадлежат M_0, дуга (1,2) удаляется из списка: $\overset{1}{(1,3)},\overset{1}{(2,3)},\overset{2}{(3,4)},\overset{3}{(4,5)},\overset{4}{(5,1)},\overset{6}{(1,4)},\overset{8}{(2,4)},\overset{9}{(3,5)}$

Вторая итерация. Просматриваем упорядоченный список дуг. Начало и конец дуги (1,3) принадлежат разным множествам. Поэтому дуга заносится в строящееся дерево (рис. 6.12), а ее второй конец (вершина 3) заносится в множество M_0 и исключается из множества M_1.

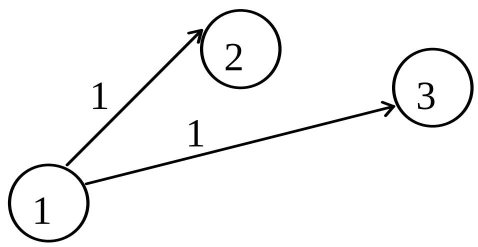


Рис. 6.12

Теперь M_0={2,1,3}, M_1={4,5}. Дуги (1,3) и (2,3) удаляются из списка упорядоченных дуг, так как оба конца этих дуг принадлежат множеству M_0: $\overset{2}{(3,4)}\overset{3}{(4,5)},\overset{4}{(5,1)},\overset{6}{(1,4)},\overset{8}{(2,4)},\overset{9}{(3,5)}$

Третья итерация. Просматриваем новый упорядоченный список дуг. Начало и конец дуги (3,4) принадлежат разным множествам. Дуга заносится в дерево (рис. 13), а ее второй конец (вершина 4) - в множество M_0 и исключается из M_1

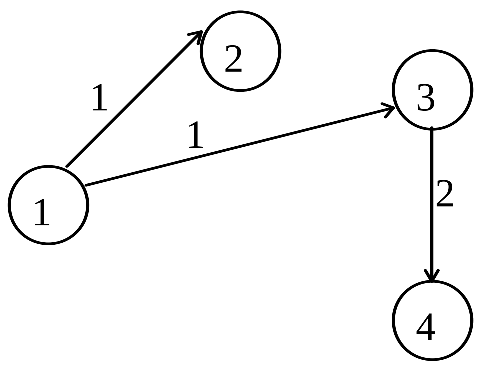


Рис. 6.13

Теперь M_0={2,1,3,4}, M_1={5}. Дуги (3,4),(1,4),(2,4) удаляются из упорядоченного списка, так как их концы принадлежат множеству M_0: $(4\overset{3}{,}5),(5\overset{4}{,}1),(3\overset{9}{,}5)$

Четвертая итерация. Просматриваем новый упорядоченный список. Начало и конец дуги (3,4) принадлежат разным множествам. Дуга заносится в дерево (рис. 6.14), а ее второй конец (вершина 5) - в множество M_0, и исключается из M_1

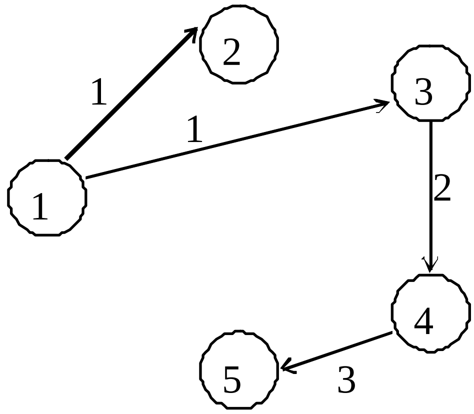


Рис. 6.14

Итак, M_0={2,1,3,4,5}, $M_1=\varnothing$. Так как $M_1=\varnothing$, то алгоритм заканчивает свою работу. На рис. 6.14 изображено полученное кратчайшее дерево. Сумма длин входящих в него дуг равна 7. Заметим, что кратчайшее дерево даже в этой задаче не единственно. Если на очередной итерации несколько дуг, начало и конец которых принадлежат разным множествам, имеют минимальную длину, то любую из таких дуг можно включать в дерево, и могут получаться различные деревья, но сумма длин входящих в них дуг одинакова. Например, в рассмотренной задаче вместо дуги (1,3) длиной 1 можно было взять дугу (2,3) такой же длины.

3.Критический путь

В некоторых моделях возникают задачи о нахождении не самого короткого, а, наоборот, самого длинного пути между двумя заданными вершинами графа. Мы рассмотрим метод решения этой задачи, а также одно из важнейших ее приложений, которое связано с вопросами управления работами над сложными объектами (стройками, проектированием и т.д.).

3.1. Поиск максимального пути в графе

Пусть задан граф <*M*,*N*,*T*>, не имеющий контуров, и каждой дуге *u* сопоставлено число $t[u]\geq 0$ - длина этой дуги. Заданы две вершины $i^-, i^+ \in M$. Требуется найти путь из i^- в i^+, имеющий наибольшую длину, где под длиной пути понимается сумма длин входящих в него дуг.

Алгоритм Дейкстры с заменой минимума на максимум здесь не работает, однако его можно немного изменить, и получится эффективный алгоритм для решения нашей задачи.

В алгоритме Дейкстры на очередной итерации мы выбирали новую текущую вершину из множества M_1 с минимальным значением пути до нее от исходной вершины. Здесь мы можем брать вершину с максимальным значением пути только в том случае, если все другие пути, проходящие через эту вершину, уже просмотрены. Поэтому для корректной работы нашего алгоритма требуется завести дополнительный вектор, обозначим его через *n*, в котором будем хранить число дуг, входящих в данную вершину, и еще не просмотренных. В начальный момент там будет число дуг, входящих в каждую вершину. Обозначим через *v* вектор самых длинных путей для вершины i^-. Так же, как и в алгоритме Дейкстры введем вектор *a*, в котором будем хранить номер вершины, из которой мы пришли. Сформируем множества: M_0- вершины, самые длинные расстояния до которых уже вычислено, и M_1- остальные вершины. В начальный момент $M_0 = \{i^-\}, M_1 = M \setminus \{i^-\}$. Обозначим через i_1 текущую вершину. В начальный момент $i_1 = i^-$. Кроме этого, $v(i^-) = 0$, $v(i) = -\infty$ для всех $i \neq i^-$; для любого $i \in M$ полагаем $a(i) = 0$, $n(i) = \left|N_i^+\right|$. Теперь для любого $u \in N_{i_1}^-$ проверяем выполнение условия $v[j(u)] < v[i_1] + t[u]$. Если оно выполнено, полагаем

$$v[j(u)] = v[i_1] + t[u],\ a[j(u)] = i_1,\ n[j(u)] = n[j(u)] - 1$$

Если теперь $M_1 = \varnothing$, то выполнение алгоритма заканчивается. Если нет, то находим $\max\limits_{i \in M_1, n(i)=0} v(i) = v(i_2)$ и считаем, что $i_1 = i_2$, $M_0 = M_0 \bigcup \{i_1\}$, $M_1 = M_1 \setminus \{i_1\}$ и переходим к следующей вершине.

Очевидно, что приведенный алгоритм конечен (на каждой итерации одна вершины удаляется из множества M_1 и заносится в множество M_0, а M_1 - конечное множество). Более того, ясно, что требуется всего лишь m итераций, если считать нулевой шаг за итерацию.

Данный алгоритм фактически позволяет найти дерево самых длинных путей для вершины i^-. Это дерево строится по вектору a - как это делалось в алгоритме Дейкстры.

6.3.2.Алгоритм поиска критических путей.

Теперь обратимся к задачам управления сложными комплексами работ. Граф, который возникает в таких задачах, трактуется следующим образом (он называется графом работ или сетевым графиком): его дуги называются работами. Для каждой работы указывается множество работ, которые должны быть завершены до того, как начнется выполнение данной работы. Указание множества предшествующих работ является заданием отношения частичного порядка на множестве N. Вершины графа мы будем трактовать как события. Будем считать, что событие наступило, если уже выполнены все непосредственно предшествующие данному событию работы.

Представляет интерес и сама задача построения графа по набору предшествований на множестве N . В силу громоздкости, анализ этой задачи мы опускаем. При желании его можно найти в [13].

Определение 6.19. Построенный граф мы будем называть *правильным* графом, если:

- существует единственная вершина $i^- \in M$, в которую не входит ни одна дуга;
- существует единственная вершина $i^+ \in M$, из которой не выходит ни одна дуга;
- в графе нет контуров;

• любая вершина $i \neq i^-$, $i \neq i^+$ лежит на каком-либо пути из i^- в i^+.

Основная задача сетевого планирования - это задача о нахождении моментов наступления различных событий. Каждая работа ограничена условием выполнения предшествовавших ей работ. В терминах графа можно сказать, что работа начинается не раньше, чем наступит событие, являющееся ее началом.

Таким образом, чтобы найти момент самого *раннего наступления события*, нужно вычислить максимальную длину путей, ведущих из вершины i^- в соответствующую вершину. Следовательно, чтобы вычислить ранние моменты наступления событий надо воспользоваться приведенным выше алгоритмом. При этом вектор a здесь не нужен.

Определение 6.20. Путь максимальной длины от начала проекта i^- до конца проекта i^+ называется *критическим путем*.

По аналогии с ранними наступлениями событий определяются *поздние моменты наступления событий* - самые поздние сроки наступления событий, при которых суммарная продолжительность проекта не увеличивается.

Обозначим вектор поздних моментов наступления событий через w.

Вектор поздних наступлений событий w может быть найден алгоритмом, аналогичным приведенному выше, но делающий все наоборот. Рассмотрим такой алгоритм. Поскольку мы требуем, чтобы суммарная продолжительность проекта не увеличивалась, то $w[i^+]=v[i^+]$. Итак, изначально $i_1 := i^+$, $v(i^-)=0$, $w(i)=+\infty$ для всех $i \neq i^+$, $n(i)=\left|N_i^-\right|$. Теперь для любого $u \in N_{i_1}^+$ проверяем условие $w[i(u)] > w[i_1]-t[u]$. Если оно выполнено, то полагаем $w[i(u)] = w[i_1]-t[u]$ и $n[i(u)] = n[i(u)]-1$. Затем, если $M_1 = \varnothing$, то работа алгоритма заканчивается. Если $M_1 \neq \varnothing$, то находим $\min\limits_{i \in M_1, n(i)=0} w(i) = w(i_2)$ и полагаем $i_1 = i_2$, $M_0 = M_0 \bigcup \{i_1\}$, $M_1 = M_1 \setminus \{i_1\}$ и переходим к началу алгоритма.

Определение 6.22. Величина $\rho[u]=w[j(u)]-v[i(u)]-t[u]$ называется *резервом времени* работы u.

Определение 6.23. Дуги, для которых резерв времени равен нулю, называются *критическими*.

Теорема 6.8. *Для того, чтобы дуга была критической, необходимо и достаточно, чтобы она принадлежала какому-нибудь критическому пути.*

Доказательство. Пусть рассматривается дуга u с началом i и концом j. Рассмотрим самый длинный путь из i^- в i. Его длина равна $v[i]$. Рассмотрим также самый длинный путь из j в i^+. Его длина равна $w[i^+]$-$w[j]$. Очевидно, что для того, чтобы $\rho[u]=0$, необходимо и достаточно, чтобы

$$v[i]+t[u]+w[i^+]-w[j]=v[i^+],$$

т.е. чтобы путь, составленный из u и этих дух путей был критическим.

Следствие 6.1. В графе $<M,N_c,T>$, составленным из критических дуг, любая дуга лежит на пути из i^-в i^+.

Таким образом граф $<M,N_c,T>$ может рассматриваться как самостоятельный сетевой график. Мы будем называть этот граф критическим графом.

Пример 6.14. Рассмотрим пример поиска критического графа на рис. 6.15. Для дуг, не входящих в эти критические пути найдем резервы времени.

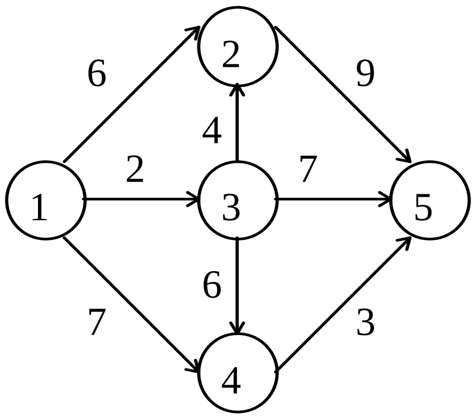


Рис. 6.15

Рассмотрев граф, изображенный на рис. 6.15 видим, что $i^-=1$, а $i^+=5$. Легко проверяется правильность графа. Очевидно, что контуров в графе нет, а любая вершина, не совпадающая с i^-и i^+(это вершины 2, 3 и 4), лежит на пути из i^-в i^+. Делаем начальные присваивания: $M_0=\{1\}$, $M_1=\{2,3,4,5\}$, $i_1=1$, $v[0,-\infty,-\infty,-\infty,-\infty]$, $n[0,2,1,2,3]$.

1 итерация. Просматриваем дуги, выходящие из текущей вершины (1). Это дуги (1,2), (1,3), (1,4). Длина пути до текущей вершины пока равна 0. Проверяем 0+6>-∞, 0+2>-∞, 0+7>-∞, и, значит, пересчитываем *v*[2], *v*[3] и *v*[4]: *v*[0,6,2,7,-∞]. Так как мы просмотрели дуги (1,2), (1,3), (1,4), то значения *n*[2], *n*[3], *n*[4] уменьшается на 1: *n*[0,1,0,1,3].

Выбираем новую текущую вершину. Она должна быть из множества M_1, а значение *n* должна быть равна 0. Такими свойствами обладает только вершина 3. Полагаем i_1=3, M_0={1,3}, M_1={2,4,5}≠∅.

2 итерация. Просматриваем дуги, выходящие из текущей вершины (3). Это дуги (3,2), (3,4), (3,5). Длина пути до текущей вершины равна 2. Проверяем 2+4=6, 2+6>7, 2+7>-∞. Поэтому пересчитываем только *v*[4] и *v*[5], а *v*[2] не меняется: *v*[0,6,2,8,9]. Но дуги (3,2), (3,4) и (3,5) просмотрены. Поэтому значения *n*[2], *n*[4] и *n*[5] уменьшаются на единицу: *n*[0,0,0,0,2].

Выбираем новую текущую вершину. Имеем: *n*[2]=*n*[4]=0, но, так как *v*[4]>*v*[2], то новой текущей вершиной становится вершина 4, т.е. i_1=4. Таким образом, M_0={1,3,4}, M_1={2,5}≠∅.

3 итерация: *v*[4]=8, пересчитываем *v* и *n*: *v*[0,6,2,8,11], *n*[0,0,0,0,1]. Поэтому i_1=2, M_0={1,3,4,2}, M_1={5}≠∅.

4 итерация: *v*[2]=6, пересчитываем *v* и *n*: *v*[0,6,2,8,15], *n*[0,0,0,0,0] и i_1:=5, M_0={1,3,4,2,5}, M_1=∅.

Заметим, что критерием окончания работы алгоритма может быть нулевой вектор *n* (нет непросмотренных дуг). Формируем данные для работы алгоритма в обратную сторону: i_1=5, M_0={5}, M_1={1,2,3,4}, *w*[+∞,+∞,+∞,+∞,15], *n*[3,1,3,1,0]

1 итерация. Просматриваем дуги, входящие в текущую вершину (5). Это дуги (2,5), (3,5), (4,5). Длина пути до текущей вершины равна 15. Проверяем 15-9<+∞, 15-7<+∞, 15-3<+∞, значит пересчитываем *w*[2], *w*[3] и *w*[4]: *w*[+∞,6,8,12,15]. Так как мы просмотрели дуги (2,5), (3,5), (4,5), то значения *n*[2], *n*[3], *n*[4] уменьшается на 1: *n*[3,0,2,0,0]. Выбираем новую текущую вершину. Она должна быть из множества M_1, а значение *n* должна быть равна 0. Такими свойствами обладают вершины 2 и 4. Но *w*[2]<*w*[4], поэтому i_1=2, M_0={5,2}, M_1={1,3,4}≠∅.

2 итерация: $w[2]=6$, пересчитываем v и n: $v[0,6,2,12,15]$, $n(2,0,1,0,0)$, $i_1=4$; $M_0=\{5,2,4\}; M_1=\{1,3\}\neq\varnothing$.

3 итерация: $w[4]=12$, w на этой итерации не меняется: $w[0,6,2,12,15]$, $n[1,0,0,0,0]$, $i_1=3$, $M_0=\{5,2,4,3\}$, $M_1=\{1\}\neq\varnothing$.

4 итерация: $w[3]=6$, v на этой итерации не меняется: $w[0,6,2,12,15]$, $n[0,0,0,0,0]$, $i_1=1$, $M_0=\{5,2,4,3,1\}; M_1\neq\varnothing$.

Итак, мы рассчитали ранние наступление событий $v[0,6,2,8,15]$ и поздние наступления событий $w[0,6,2,12,15]$. Теперь можно вычислить резервы времени для каждой работы:

$1\rightarrow2(0)$ $w[2]$-$v[1]$-$t[(1,2)]$=6-0-6=0

$1\rightarrow3(0)$ 2-0-2=0

$1\rightarrow4(5)$ 12-0-7=5

$2\rightarrow5(0)$ 15-6-9=0

$3\rightarrow2(0)$ 6-2-4=0

$3\rightarrow4(4)$ 12-2-6=4

$3\rightarrow5(6)$ 15-2-7=6

$4\rightarrow5(4)$ 15-8-3=4

Видим, что работы (1,2), (1,3), (3,2) и (2,5) являются критическими, так как их резервы времени равны нулю. Они образуют критический граф, изображенный на рис. 6.16

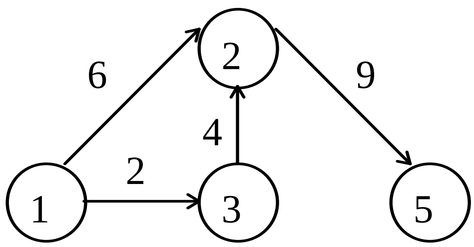


Рис. 6.16

Остальные работы имеют некоторые резервы времени, то есть любая из таких работ может выполняться на столько единиц времени больше, чем ей положено.

Мы рассмотрели только одну из задач, связанных с организацией работ. Имеется большое число дополнительных проблем, относящихся к той же области, - вероятностные модели, оценки продолжительности работ, составление расписаний при ограниченности числа исполнителей, распределения финансирования по работам и др.

6.4.Эйлеровы и Гамильтоновы пути, циклы и контуры

Интересные задачи связаны с построением полных обходов графа. В термин «полный обход» можно вкладывать различное содержание. Более простой из них является задача построения обхода, содержащего все дуги графа.

6.4.1.Эйлеровы пути, циклы и контуры

Для примера сначала рассмотрим задачу о Кенигсбергских мостах.

В городе Кенигсберге в 18 веке была схема мостов приведенная на рис. 6.17,

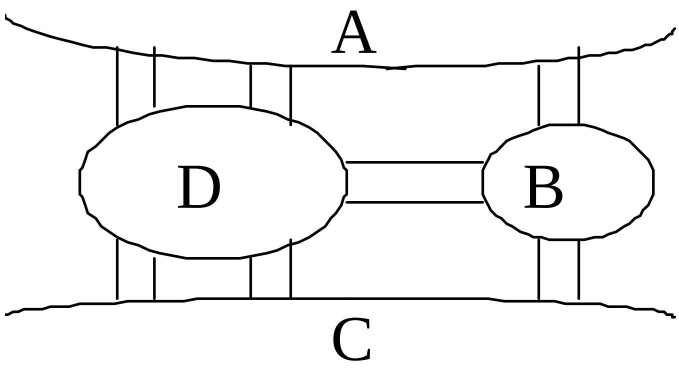


Рис. 6.17

где *A*, *B*, *C* и *D* - различные районы города. Районы *B* и *D* расположены на островах. Задача ставится так: выйдя из дома в одном из районов города обойти все мосты, пройдя по каждому только один раз, и вернуться домой.

Очевидно, задача имеет эквивалентную постановку в терминах теории графов. Рассмотрим граф, приведенный на рис. 6.18.

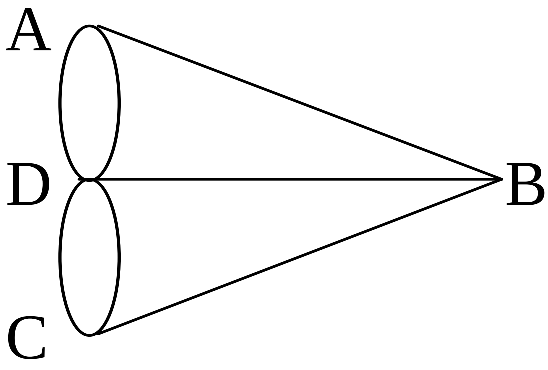


Рис. 6.18

Итак, начав с любой из вершин, требуется пройти по каждой дуге графа ровно один раз и вернуться в исходную вершину.

Постановка задачи принадлежит Л.Эйлеру. Однако, он не только поставил задачу, но и доказал невозможность такого маршрута. Кроме того, он получил результат, когда такого рода обход графа существует.

Имеется много задач подобого типа. Наиболее известны из них задачи рисования некоторой фигуры, не отрывая карандаша от бумаги. В качестве примера подобного рода задач приведем так называемые «сабли Магомеда», изображенные на рис. 6.19

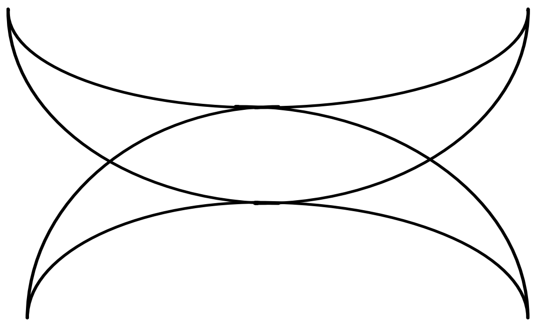


Рис. 6.19

Требуется нарисовать этот рисунок, не отрывая карандаша от бумаги так, чтобы карандаш вернулся в исходную точку.

Прежде, чем приводит результат, полученный Л.Эйлером введем ряд определений.

Определение 6.24. Замкнутая цепь, содержащая по одному разу каждую дугу графа называется *эйлеровым циклом*.

Замечание 6.2. В определении 6.24 мы снимаем предположение о том, что в цепь каждая вершина графа входит не более одного раза.

Определение 6.25. Граф, содержащий эйлеров цикл, называется *эйлеровым графом*.

Определение 6.26. Путь, содержащий по одному разу каждую дугу графа, называется *эйлеровым путем*, а если он замкнут, - то *эйлеровым контуром*.

Теорема 6.9. *Связный граф $<M,N,T>$ обладает эйлеровым циклом тогда и только тогда, когда каждой его вершине инцидентно четное число дуг.*

Доказательство. Необходимость очевидна. Если в графе есть вершина, которой инцидентно нечетное число дуг, все дуги не смогут войти в эйлеров цикл, поскольку каждое прохождение этой вершины использует по две инцидентных этой вершине дуги.

Достаточность. Пусть в связном графе каждой вершине инцидентно четное число дуг. Построим какой-нибудь цикл, и множество его вершин обозначим через M_0, а множество его дуг через N_0. Положим $N_1 = N \setminus N_0$. В графе $< M, N_1, T >$ нет вершин, которым инцидентно нечетное число дуг. Кроме того, существуют вершины из M_0, которым инцидентны дуги из N_1 (если бы таких дуг не было, то графы $< M_0, N_0, T >$ и $< M \setminus M_0, N_1, T >$ были бы не связаны между собой вопреки, предположению о связности исходного графа).

Выберем вершину $i \in M_0$, которой инцидентны дуги из N_1 и построим из дуг N_1 цикл, содержащий вершину $i \in M_0$. Встроим дуги их этого цикла в начальный цикл, соответственно пополнив M_0 и N_0. Снова положим $N_1 = N \setminus N_0$. Продолжаем этот процесс пока $N_1 \neq \varnothing$. Построенная в результате замкнутая цепь и будет искомым эйлеровым циклом.

Заметим, что доказательство теоремы конструктивно и фактически дает алгоритм построения эйлерова цикла.

На основании доказанной теоремы можно сделать вывод, что задача о Кенигсбергских мостах решения не имеет, так как в графе на рис. 6.18 есть вершины, которым инцидентно нечетное число дуг. Также на основании теоремы можно сделать вывод, что задача о «саблях Магомеда» решение имеет.

Сформулируем теперь аналогичную теорему, дающую условия существования эйлерова контура.

Теорема 6.10. *Для того, чтобы в связном графе <M,N,T> существовал эйлеров контур необходимо и достаточно, чтобы для любого* $i \in M$ $\left|N_i^-\right| = \left|N_i^+\right|$.

Замечание 6.3. Доказательство теоремы аналогично доказательству предыдущей и здесь мы его приводить не будем. Его можно найти в [13].

Более трудными являются задачи обхода графа по всем вершинам.

6.4.2. ГАМИЛЬТОНОВЫ ПУТИ, ЦИКЛЫ И КОНТУРЫ

Определение 6.27. Цикл (путь, контур) называется *гамильтоновым*, если через каждую вершину графа он проходит ровно по одному разу.

В общем случае задача нахождения гамильтонова пути, цикла или контура достаточно сложна, однако, для некоторых классов графов она решается просто. К таким графам относятся полные графы.

Определение 6.28. Граф называется *полным*, если любая его пара вершин соединена дугой хотя бы в одном направлении.

Теорема 6.11 (Кёнига). *В полном графе всегда существует гамильтонов путь.*

Доказательство. Построим произвольный путь в графе: $i_1, i_2, ..., i_k$. Возьмем произвольную вершину j, не входящую в этот путь (если такой вершины не найдется то $i_1, i_2, ..., i_k$ - гамильтонов путь). Если из вершины j в вершину i_1 идет дуга, то вершину j вставляем в начало пути. Если такой дуги нет, то, в силу полноты графа, существует дуга из i_1 в j. Если из j в i_2 идет дуга, то вставляем вершину j в путь: $i_1, j, i_2, ..., i_k$. Если такой дуги нет, то из i_2 в j идет дуга. Продолжая этот процесс, мы либо вставим вершину j между какими-либо двумя вершинами пути, либо убедимся в том, что из вершины i_k в вершину j идет дуга. В последнем случае мы вставим вершину j в конец путь. Аналогично включаются в этот путь все остальные вершины, не вошедшие в него.

Теперь мы рассмотрим одну более сложную задачу - задачу нахождения гамильтонова контура минимальной длины. Она известна как задача о бродячем торговце или задача о коммивояжере (в оригинале - traveling salesman problem).

Задача о коммивояжере. Имеется n городов заданы расстояния между любыми двумя городами. Коммивояжер выходит из некоторого города. Ему нужно обойти все города, побывав в каждом ровно по одному разу, и вернуться в исходный город так, чтобы суммарная длина маршрута была минимальна.

6.4.3. Метод ветвей и границ

Когда мы решаем задачи типа задачи о коммивояжере, надо помнить что один метод решения такой задачи всегда есть - это метод, заключающийся в полном переборе всех вариантов. Проблема заключается в том, что число этих вариантов слишком велико. Так, для задачи о коммивояжере число различных маршрутов равно $(n-1)!$.

Большая группа методов улучшенного перебора объединяется под названием «метод ветвей и границ». Общая идея этого метода достаточно проста. Предположим, что мы решаем задачу:

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

где A - некоторое конечное множество. Пусть имеется разбиение множества A на непересекающиеся подмножества

$$A = A_0 \cup A_1 \cup \ldots \cup A_k$$

и известно, что в множестве A_0 оптимальное решение не содержится. Тогда

$$\min_{x\in A} f(x) = \min_{i\in 1..k}\{\min_{x\in A_i} f(x)\}$$

Разбиение множества A на подмножества преследует несколько целей. Прежде всего, выделение множества A_0 сокращает задачу, и, поэтому, мы стремимся как можно большую часть множества A передать в A_0. Разбиение на подмножества A_i оставшейся части A может упростить задачу нахождения минимума $f(x)$. Рассмотрим способы пополнения множества A_0.

Если мы знаем, что для двух элементов разбиения A_i и A_j имеет место соотношение: $\min\limits_{x\in A_i} f(x) > \min\limits_{x\in A_j} f(x)$, то на множестве A_i заведомо оптимальных решений нет, и $A_0 := A_0 \bigcup A_i$.

Определение 6.29. Наименьшее значение функции *f*, которое мы имеем на данный момент мы будем называть *рекордом*.

Если мы имеем некоторый рекорд $f(x_0)$ в точке x_0 и: $\min\limits_{x\in A_i} f(x) > f(x_0)$, то и в этом случае в A_i оптимальных решений нет, и $A_0 := A_0 \bigcup A_i$. Заметим, что в этом случае нам достаточно чтобы оценка снизу всех значение на множестве A_i была больше рекорда.

Имея некоторое разбиение множества *A*, можно проводить его дальнейшее измельчение. При этом последовательно уточняются оценки и находятся новые допустимые решения, а некоторые из элементов разбиения переводятся в множество A_0. Реализация этого общего принципа имеет много сложностей, часто меняющихся от задаче к задаче. Поэтому рассмотрим действие этих принципов на задаче о коммивояжере, иллюстрируя рассмотрение конкретным примером. В этой задачи все расстояния можно задавать с помощью квадратной матрицы C размерности $n \times n$. Если пути из города *i* город *j* нет, то $C_{ij} = \infty$. Диагональные элементы этой матрицы $C_{ii} = \infty$, так как любые петли запрещены по смыслу задачи.

Определение 6.30. Две матрицы *C* и *D* мы будем называть *эквивалентными*, если одна из них получается из другой прибавлением к элементам каждой строки одного и того же числа (для разных строк эти числа могут быть различны) и к элементам каждого столбца одного и того же числа (для разных столбцов эти числа могут быть различны). Например,

$$D_{ij} = C_{ij} + e_i + f_j,\ i = 1,\ldots,n,\ j = 1,\ldots,n.$$

Ясно, что тогда и вторая матрица получается из первой прибавлением к ее строкам и столбцам постоянных чисел ($-e_i$ и $-f_j$ соответственно), так что свойство эквивалентности взаимно.

Теорема 6.12. *Оптимальные решения двух задач о коммивояжере с эквивалентными матрицами совпадают (без доказательства).*

Заметим, что целевые функции в таких задачах отличаются на величину равную $\sum_{i=1}^{n} e_i + \sum_{j=1}^{n} f_j$.

При неотрицательной матрице C нуль будет оценкой снизу для минимума. Поэтому, если в матрице для задачи о коммивояжере мы вычитаем минимальный элемент в каждой строке, а затем минимальный элемент в каждом столбце, то, очевидно, сумма всех таких минимальных элементов будет оценкой снизу для целевой функции такой задачи.

Такой способ построения оценки снизу универсален в данной задаче и будет применяться при вычислении всех оценок.

Пример 6.15. Рассмотрим задачу с 6 городами, приведенную в таблице:

	1	**2**	**3**	**4**	**5**	**6**
1	∞	8	7	1	6	4
2	9	∞	2	7	5	6
3	7	1	∞	8	4	5
4	2	8	6	∞	3	5
5	8	9	5	2	∞	2
6	9	5	3	1	5	∞

Так как в задаче 6 городов, то множество A содержит 5! вариантов. Это 120 вариантов. Перебрать все 120 вариантов вручную - слишком трудоемкая задача. Поэтому применим метод «ветвей и границ» для этой задачи. На каждом шаге одно из множеств будет делиться на два подмножества. Принцип деления для каждого множества будет один: либо мы разрешаем переход из одного города в другой, либо мы его запрещаем. При этом в дереве вариантов, которое мы будем строить, начиная с множества A - всех вариантов задачи, ветвь, где мы разрешаем переход, будем рисовать слева и называть левой, а ветвь, где запрещаем переход, будем рисовать справа, и называть правой.

Итак, в исходной матрице вычитаем, сначала минимальный элемент в каждой строке из всех элементов этой строки, а затем в полученной эквивалентной матрице то же проделываем со столбцами. Получаем таблицу:

	1	**2**	**3**	**4**	**5**	**6**	
1	∞	7	6	0	4	3	**1**
2	7	∞	0	5	2	4	**2**
3	6	0	∞	7	2	4	**1**
4	0	6	4	∞	0	3	**2**
5	6	7	3	0	∞	0	**2**
6	8	4	2	0	3	∞	**1**
					1		

В последнем столбце записаны минимальные элементы, которые мы вычли из остальных элементов каждой строки. Аналогично в последней строке - минимальные элементы столбцов. Сложив все эти элементы получаем оценку снизу множества всех вариантов - 10.

Теперь надо выбрать переход для ветвления. Если мы выбрали переход из города *i* в город *j*, то, очевидно, что при вычислении оценки левой ветви надо увеличить оценку на C_{ij}, где C_{ij} элемент преобразованной матрицы. Поскольку, переход из *i* в *j* зафиксирован, то остальные элементы строки *i* нам не нужны, аналогично остальные элементы столбца *j* нам не нужны (можно считать, что вершины *i* и *j* «склеились» при этом в одну). Таким образом, по левой ветви в матрице вычеркивается строка *i* и столбец *j*. По правой ветви полагаем $C_{ij}=\infty$. Число вариантов по левой ветви будет (n-2)!, а по правой ветви остальные - (n-1)!-(n-2)!. Ясно, что в левой ветви вариантов существенно меньше, чем в правой. Так, в нашем примере при первом ветвлении в левой ветви будет 4!=24 варианта, а в правой - 5!-4!=120-24=96. Поэтому нам желательно, чтобы при ветвлении оценка по левой ветви возрастала как можно меньше, а по правой - как можно больше. В этом случае, есть надежда, что она не будет просматриваться вообще. Итак, чтобы оценка по левой ветви не возрастала надо рассматривать нули в преобразованной матрице *C*. Среди них нужно выбрать такой, чтобы оценка по правой верви была максимальной. Очевидно, что после присваивания C_{ij}:=∞, может измениться минимальный элемент в строке *i* и минимальный элемент в столбце *j*, в остальных строках и столбцах минимальные элементы не изменяются. Следовательно, мы можем подсчитать на сколько изменится оценка по правой ветви и выбрать тот нуль, где оценка меняется максимально:

запрет 1→4(+3); запрет 2→3(+2); запрет 3→2(+6);
запрет 4→1(+6); запрет 4→5(+2); запрет 5→4(0);
запрет 5→6(+3); запрет 6→4(+2);

Видим, что максимальное увеличение правой ветви будет, если запретить переход 3→2(+6) или 4→1(+6). Выберем для ветвления переход 3→2. Варианты левой ветви обозначим через A_1 а варианты правой ветви - A_2. Итак, после подсчета оценок (а они считаются также как и ранее: вычитаем минимальный элемент в каждой строке и минимальный элемент в каждом столбце), получаем две матрицы:

A_1(14) 3→2

	1	**3**	**4**	**5**	**6**	
1	∞	4	0	4	3	
2	5	∞	3	0	2	**2**
4	0	2	∞	0	3	
5	6	1	0	∞	0	
6	8	0	0	3	∞	
		2				

A_2(16) (запрет 3→2)

	1	**2**	**3**	**4**	**5**	**6**	
1	∞	3	6	0	4	3	
2	7	∞	0	5	2	4	
3	4	∞	∞	5	0	2	**2**
4	0	2	4	∞	0	3	
5	6	3	3	0	∞	0	
6	8	0	2	0	3	∞	
		4					

Заметим, что в левой ветви в матрице переход 2→3 запрещен. Это сделано для того, чтобы исключить контуры, не содержащих всех городов. В частности, в данном случае исключается такой контур 3→2→3.

Итак, $A = A_1 \bigcup A_2$. Теперь надо продолжить ветвление, рассматривая либо варианты из множества A_1, либо варианты из множества A_2. Дальнейшее зависит от политики ветвления, которую мы будем применять. Рассмотрим 3 политики ветвления:

- ветвление побегами,
- левосторонний обход дерева вариантов,
- ветвление полным фронтом.

При ветвлении побегами продолжается то множество вариантов, которое имеет наименьшую оценку.

Левосторонний обход дерева вариантов заключается в том, что мы всегда продолжаем левую ветвь до тех пор, пока это возможно. Когда это невозможно, возвращаемся назад, делаем один шаг вправо, а затем снова идем влево до тех пор пока это возможно и т.д.

Ветвление полным фронтом заключается в том, что сначала рассматривается каждое подмножество первого уровня, затем рассматривается каждое подмножество второго уровня и т.д., пока мы не дойдем до последнего уровня. Для применения этой политики необходимо иметь достаточно хороший рекорд с тем, чтобы сразу отбрасывать все варианты, у которых оценка выше рекорда. Если отбрасывать ничего не удается, то получается полный перебор.

Это же относится и к двум другим политикам. Наличие хорошего рекорда позволяет отбросить многие неперспективные варианты, однако при применении этих политик есть шанс получить такой рекорд при ветвлении. Если же применяется политика ветвления полным фронтом, то там такой возможности нет.

Итак, вернемся к нашему примеру. Поскольку рекорда пока у нас пока нет, то политику ветвления полным фронтом мы применять не будем. Применим политику ветвления побегами (в данном случае она пока совпадает с левосторонним обходом дерева вариантов). Ветвим множество A_1. Опять рассматриваем для ветвления нули матрицы вариантов A_1, с тем, чтобы определить тот нуль, который дает максимальное увеличение оценки по правой ветви:

запрет 1→4(+3); запрет 2→4(+2);запрет 4→1(+5);

запрет 4→5(0); запрет 5→4(0); запрет 6→3(+1);
запрет 6→4(0);

Видим, что максимальное увеличение правой ветви будет, если запретить переход 4→1(+5). Варианты левой ветви обозначим через A_{11}, а варианты правой ветви - A_{12} (так как это подмножества множества A_1). В матрице, соответствующей левой ветви необходимо $C_{14}=\infty$, так как иначе может быть контур 4→1→4. После подсчета оценок получаем две матрицы:

A_{11}(17) 4→1

	3	**4**	**5**	**6**	
1	1	∞	1	0	**3**
2	∞	3	0	2	
5	1	0	∞	0	
6	0	0	3	∞	

A_{12}(19) запрет 4→1

	1	**3**	**4**	**5**	**6**
1	∞	4	0	4	3
2	0	∞	3	0	2
4	∞	2	∞	0	3
5	1	1	0	∞	0
6	3	0	0	3	∞

В результате получаем дерево вариантов, изображенное на рис. 6.20.

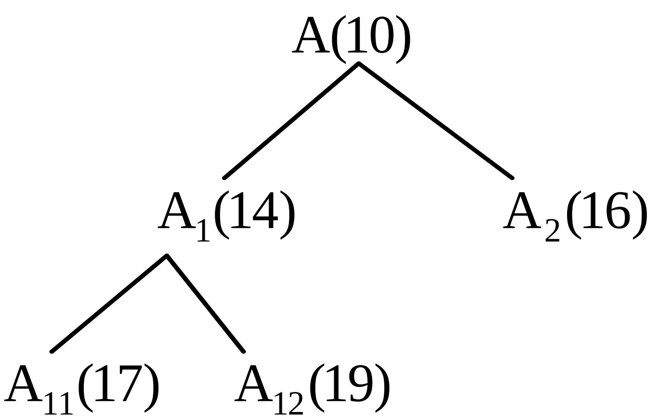


Рис. 6.20

Так как мы используем политику ветвления побегами, надо продолжать ветвление множества A_2 (при левостороннем обходе надо было бы для ветвления брать множество A_{11}).

Ветвим множество A_2. Опять рассматриваем для ветвления нули матрицы вариантов A_2, с тем, чтобы определить тот нуль, который дает максимальное увеличение оценки по правой ветви:

запрет 1→4(+3); запрет 2→3(+4); запрет 3→5(+2);

запрет 4→1(+4); запрет 4→5(0); запрет 5→4(0);

запрет 5→6(+3); запрет 6→4(+0); запрет 6→2(+2);

Для ветвления берем переход 4→1. После подсчета оценок, получаем:

A_{21}(19) 4→1

	2	**3**	**4**	**5**	**6**	
1	0	3	0	1	0	**3**
2	∞	0	5	2	4	
3	∞	∞	5	0	2	
5	3	3	0	∞	0	
6	2	0	0	3	∞	

A_{21}(20) (запрет 4→1)

	1	**2**	**3**	**4**	**5**	**6**
1	∞	3	6	0	4	3
2	3	∞	0	5	2	4
3	0	∞	∞	5	0	2
4	∞	2	4	∞	0	3
5	2	3	3	0	∞	0
6	4	0	2	0	3	∞

В результате получаем дерево вариантов, изображенное на рис. 21.

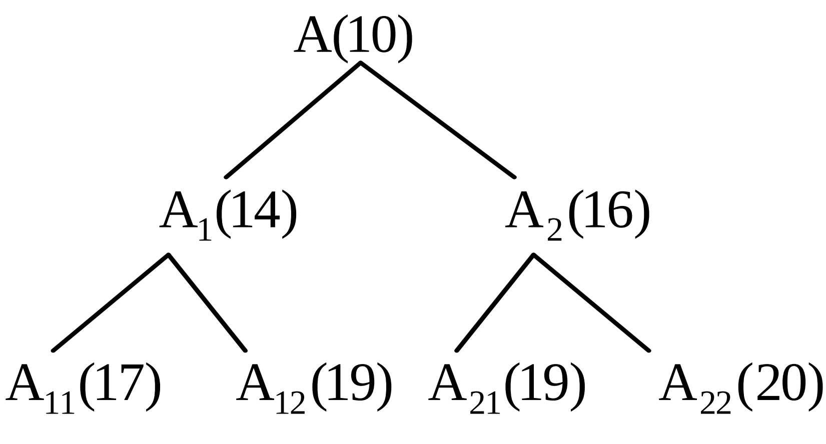


рис. 6.21

В данном случае $A = A_{11} \bigcup A_{12} \bigcup A_{21} \bigcup A_{22}$. Ясно, что оценка снизу всех вариантов теперь равна 17. Отсеять ни один из вариантов пока мы не можем. Продолжаем ветвление. Теперь минимальной оценкой обладает множество A_{11}. Рассматриваем матрицу, соответствующую этому множеству. Опять ищем нуль, замена которого на ∞ приводит к максимальному возрастанию оценки правой ветви:

запрет 1→6(+1); запрет 2→5(+3); запрет 5→4(0);

запрет 5→6(0); запрет 6→3(+1);запрет 6→4(0);

Для ветвления берем переход 2→5. В матрице, соответствующей левой ветви надо $C_{53} = \infty$, так как иначе может быть контур 3→2→5→3. После подсчета оценок получаем две матрицы:

A_{111}(17) 2→5

	3	**4**	**6**
1	1	∞	0
5	∞	0	0
6	0	0	∞

A_{112}(20) запрет 2→5

	3	**4**	**5**	**6**	
1	1	∞	0	0	
2	∞	1	∞	0	**2**
5	1	0	∞	0	
6	0	0	2	∞	

Итак, получили дерево вариантов, изображенное на рис. 6.22

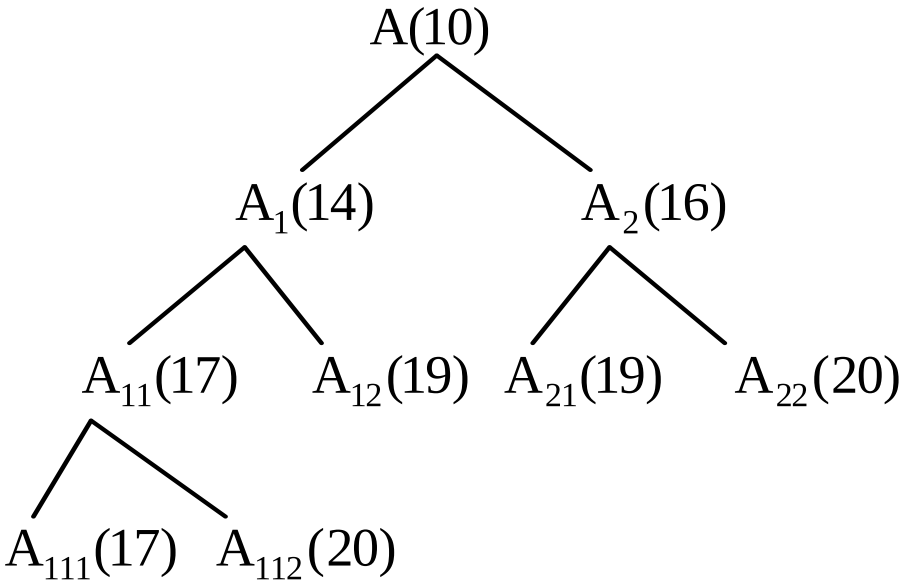


Рис. 6.22

Минимальной оценкой обладает теперь множество A_{111}. Однако, соответствующая матрица имеет размерность 3, а значит, с учетом уже фиксированных переходов, имеется всего 2 варианта. Поэтому проще выписать сразу эти варианты:

$$A_{1111}: 1 \xrightarrow{1} 3 \longrightarrow 2 \longrightarrow 5 \xrightarrow{0} 6 \xrightarrow{0} 4 \longrightarrow 1 \text{ и}$$

$$A_{1112}: 1 \xrightarrow{0} 6 \xrightarrow{0} 3 \longrightarrow 2 \longrightarrow 5 \xrightarrow{0} 4 \longrightarrow 1.$$

Соответственно имеем значение целевой функции для варианта A_{1111} - 18, а для варианта A_{1112} - 17. В отличие от всех остальных вершин дерева эти отличаются тем, что дальнейшее ветвление для них невозможно. Такие вершины мы будем называть листьями дерева. Поскольку 17 и 18 это длины соответствующих гамильтоновых контуров, то 17 - это рекорд, который мы имеем на данный момент времени. Просматривая полученное дерево вариантов (рис. 6.23), делаем выводы, что

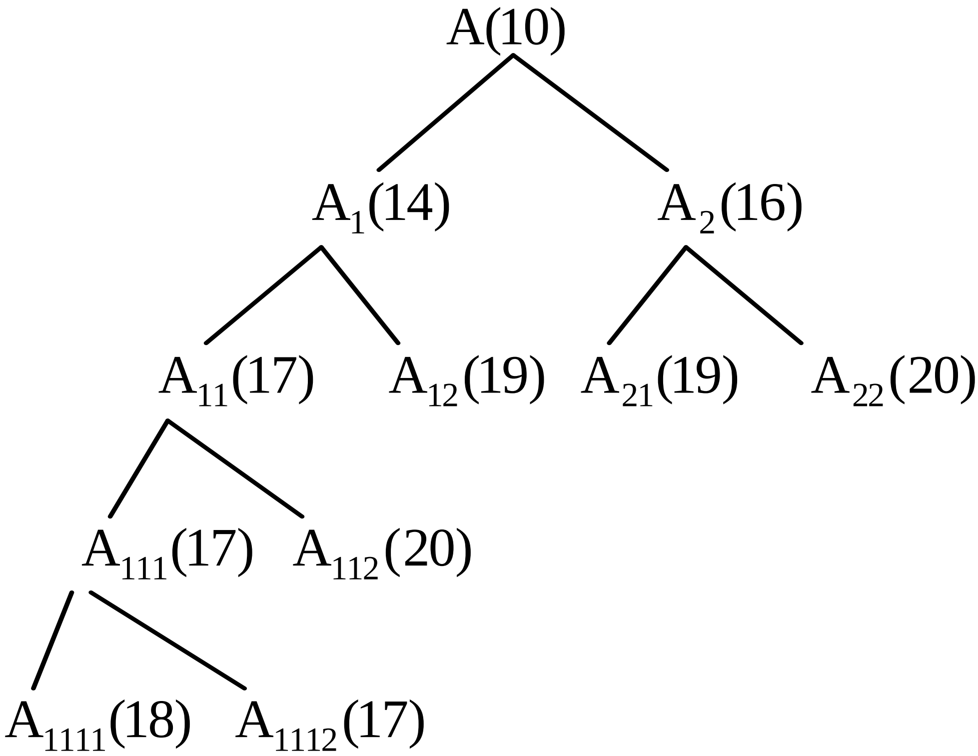


Рис. 6.23

у вариантов $A_{12}, A_{21}, A_{22}, A_{112}$ оценки снизу больше рекорда, а значит там оптимального решения быть не может, и эти ветви можно не рассматривать. Но, таким образом, все дерево вариантов у нас просмотрено, а это значит, что оптимальным решением будет гамильтонов контур:

$$1 \xrightarrow{4} 6 \xrightarrow{3} 3 \xrightarrow{1} 2 \xrightarrow{5} 5 \xrightarrow{2} 4 \xrightarrow{2} 1 \ (17)$$

Длины путей здесь взяты из исходной матрицы. Итак, в данной задаче метод ветвей и границ существенно уменьшил объем перебора.

Если бы применяли политику левостороннего обхода дерева вариантов, то порядок просмотра был бы иным. После ветвления A_1 на A_{11} и A_{12}, мы, вместо того, чтобы ветвить A_2, независимо от полученных оценок произвели бы ветвление множества вариантов A_{11} на A_{111} и A_{112}. После этого произвели ветвление A_{111} на A_{1111} и A_{1112}, получили бы рекорд (тот же что и ранее). После этого в дереве вариантов закрыли бы ветви A_{12} и A_{112}. И только после этого продолжили бы ветвление множества вариантов A_2. Очередное ветвление множества A_2 на A_{21} и A_{22} привело бы к закрытию ветвей A_{21} и A_{22}, так как их оценки больше чем рекорд. После этого все дерево вариантов было бы просмотрено.

Заметим, что при применении метода ветвей и границ нет гарантии, что отсеется большая часть вариантов, и не придется делать почти полного перебора. Поэтому данный метод иногда называют «метод улучшенного перебора». Рост числа вариантов при увеличении размерности идет по экспоненте, в связи с этим решать задачи о коммивояжере большой размерности весьма сложно. Поэтому на практике часто в качестве «приближенного» решения рассматривают рекорд, который имеется на момент прекращения вычислений.

Напомним, что описанный метод называется методом ветвей и границ, так как в нем поиск оптимального решения идет по ветвям дерева вариантов, и при этом ветвлении используются границы - оценки.

6.5. Задача о куче камней

В этом параграфе мы рассмотрим одну задачу построения наилучшего разбиения множества. Интересен метод, которым будет решаться эта задача - в нем метод ветвей и границ будет сочетаться с наивным и на первый взгляд неперспективным подходом. Однако алгоритм получается эффективный.

Итак, пусть имеется множество камней N. Для каждого камня $j \in N$ задано $w[j]>0$ - вес камня. Требуется разделить это множество на заданное число куч m таким образом, чтобы вес самой тяжелой из куч был минимален.

Математическая постановка задачи. Требуется найти такое разбиение $N = \bigcup\limits_{i=1..m} N_i$, $N_i \cap N_j = \varnothing$ при $i \neq j$, которое является решением задачи

$$\min_{i=1..m}\{\max \sum_{j \in N_i} w[j]\}$$

Рассмотрим один общий подход к решению подобных комбинаторных задач. Пусть нам требуется найти $\min_{x \in A} f(x)$. Предположим, что мы имеем два числа *e* и *r*, для которых известно, что $e < f(x) \le r$. Выберем какое-либо число $z \in (e, r)$ и рассмотрим так называемую z-задачу: найти

$$\{x \in A : f(x) \le z\}.$$

Если нам удалось найти решение этой задачи, то *r*=*z*, после чего можно повторить процесс. Если мы покажем, что z-задача решения не имеет, то *e*=*z*, и также повторяем процесс. В конце концов полуинтервал (*e*,*r*] стягивается в точку, которая и будет оптимальным решением нашей задачи.

Главная трудность здесь, разумеется, лежит в способе решения получающейся z-задачи. Оказывается, в некоторых случаях, в том числе и в задаче о куче камней, можно искать решение z-задачи методом ветвей и границ.

z-задача, соответствующая задаче о куче камней, выглядит следующим образом:

Найти разбиение $N = \bigcup_{i=1..m} N_i$, $N_i \cap N_j = \varnothing$ при $i \ne j$, при котором $\sum_{j \in N_i} w[j] \le z$ для любого $i = 1, \dots, m$.

Будем решать эту задачу методом ветвей и границ с односторонним обходом дерева вариантов. Считаем, что у нас имеется *m* ящиков для камней, каждый из которых имеет вместимость (по весу) *z*. Тогда, в ящиках должно оставаться свободное место $s = z \times m - \sum_{j \in N} w[j]$ (от общей вместимости всех ящиков отнимается вес всех камней).

Будем укладывать камни в ящики, последовательно один за другим, стремясь каждый раз уложить самый большой из оставшихся (и помещающихся) камней. Остающееся в ящике свободное место будет вычитаться из остатка свободного места *s*, и этот остаток в течение всего вычислительного процесса должен оставаться неотрицательным. Если *s* становится отрицательным (камней, которые мы могли бы положить в ящик нет, а оставшееся свободное место в ящике больше, чем *s*), то возвращаемся на шаг назад, отменяем последнее назначение, берем следующий по весу камень и т.д., так как это и полагается делать при одностороннем обходе дерева вариантов. Либо мы уложим все камни по кучам, либо, возвращаясь назад, дойдем до первого камня первой кучи. Если его надо отменять, то ясно, что укладка невозможна.

Пример 6.16. Пусть исходная куча состоит из 16 камней, которые нужно разложить на 4 кучи (m=4):

80,74,70,65,62,57,55,52,43,41,40,39,37,35,33,19

Мы сразу упорядочиваем наши камни по убыванию веса. Суммарный вес камней равен 802. Вес максимальной кучи будет больше, чем $\frac{\sum_{j \in N} w[j]}{m}$. Поэтому в нашей задаче в качестве e можно взять 200, причем эта оценка снизу недостижима, так как 802/4=200.5, все веса камней целые, а значит, вес максимальной кучи должен быть целым. В качестве оценки сверху для минимума максимального веса r можно взять 802 (наихудший вариант, когда все камни складываются в одну кучу). Теперь нам надо выбрать z. Поскольку оценка сверху явно завышена, возьмем z ближе к 200. Попробуем решить задачу для z=205. Подсчитываем свободное место s=205×4-802=18. Начинаем заполнять первый ящик. Будем записывать наши действия двумя строками, в верхней остаток места в выбранном ящике, а в нижней - вес выбранного камня:

1 ящик:	205	125	51	8	
	80	74	43		

Остаток свободного места s=18-8=10

2 ящик:	205	135	70	8	s=10-8=2
	70	65	62		
3 ящик:	205	148	93	41	s=2
	57	55	52	41	

Четвертый ящик можно не заполнять: в него уместится все остальное со свободным местом 2.

Теперь мы знаем, что оптимальное решение задачи лежит между 200 и 205. Попробуем решить z-задачу с z=203 (середина между 201 и 205). Если нам удастся укладка, то останется просмотреть

201 и 202. Если нет - то останется просмотреть 204. Подсчитываем s=203×4-802=10.

1 ящик:	203	123	49	6			s=10-6=4
	80	74	43				

2 ящик:	203	133	68	6			s=4-6=-2
	70	65	62				

Остаток слишком большой. У нас всего 4 единицы свободного места, а требуется 6. Поэтому отменяем последнее назначение и ставим следующий по порядку камень - 57. Остаток - 11. Тоже не походит. Отменяем камень весом 57, берем следующий - 55. Остаток -13. Не походит. Берем 52. Остаток 16. Не походит. Берем 41 (43 уже в первой куче). Остаток 27. Получаем:

2 ящик.	203	133	68	27	8		s=4-8=-4
	70	65	41	19			

Опять не подходит. Отменяем 2 последних назначения, бер ем следующий 40, получаем

2 ящик:	203	133	68	28	9		
	70	65	40	19			

Опять не подходит. Отменяем 2 последних назначения, берем следующий 39. Проделываем то же, что и выше, убеждаемся, укладка не удается. Аналогично 37. Берем теперь 35:

2 ящик:	203	133	68	33			s=4-0=4
	70	65	35	33			

Второй ящик уложили. Остаток свободного места 4. Укладываем 3 ящик.

3 ящик:	203	141	100	60	21	2	s=4-2=2
	62	41	40	39	19		

Четвертый ящик можно не заполнять: в него уместится все остальное со свободным местом 2.

Теперь мы знаем, что оптимальное решение задачи лежит между 200 и 203. Попробуем решить z-задачу с z=201. Если нам удастся укладка, то это и будет оптимальное решение. Если нет - то останется просмотреть 202. Подсчитываем s=201×4-802=2.

1 ящик:	201	121	47	4	
	80	74	43		

Укладка не удалась, так как остаток больше, чем у нас имеется свободного места. Отменяем последнее назначение. Пробуем камень весом 41. Он тоже не подходит. Далее 40, 39, 37, 35,33 и 19. Укладка не удается. Возвращаемся на 2 шага назад. Отменяем 74. Берем следующий. Итак,

1 ящик:	201	121	51	8	
	80	70	43		

Укладка не удалась, так как остаток больше, чем у нас имеется свободного места. Отменяем последнее назначение. Пробуем камни весом 41, 40, 39, 37, 35, 33 и 19. Укладка не удается. Возвращаемся на 2 шага назад. Отменяем 70. Берем следующий. Итак,

1 ящик:	201	121	56	1	s=2-1=1
	80	65	55		

Первый ящик уложили. Остаток свободного места 1. Укладываем 2 ящик.

2 ящик:	201	127	57		s=1-0=1
	74	70	57		

Второй ящик уложили. Остаток свободного места 1. Укладываем 3 ящик.

3 ящик:	201	139	87	45	4
	62	52	43	41	

Укладка не удалась, так как остаток больше, чем у нас имеется свободного места. Отменяем последнее назначение. Пробуем камни весом 40, 39, 37, 35, 33 и 19. Укладка не удается. Возвращаемся на 2 шага назад. Отменяем 43. Берем следующий. Итак,

3 ящик:	201	139	87	46	6
	62	52	41	40	

Укладка не удалась, так как остаток больше, чем у нас имеется свободного места. Отменяем последнее назначение. Пробуем камни весом 40, 39, 37, 35, 33 и 19. Укладка не удается. Возвращаемся на

2 шага назад. Отменяем 41. Берем следующий. Аналогично проверяем, что укладка не удается. Далее пробуем 40, 37 и, наконец, 35:

3 ящик:	201	139	87	52	19	s=1-0=1
	62	52	35	33	19	

Четвертый ящик можно не заполнять: в него уместится все остальное со свободным местом 1. Тем не менее приведем заполнение четвертого ящика, так как мы получили оптимальное решение.

4 ящик:	201	158	117	77	38	1	s=1-0=1
	43	41	40	39	37		

Итак, поскольку решение z-задачи с z=200 невозможно, то указанная выше укладка с z=201 и является искомым оптимальным решением. Вес максимальной кучи (а это вторая и третья) равен 201.

Ответы

1. а) $\lambda_A = 7,\ x_A = (0,0,t),\ t > 0$; б) $\lambda_A = 8,\ x_A = (t,t,t),\ t > 0$;

2. а) $\lambda_A = 3,\ x_A = (t,0,0),\ t > 0$; б) $\lambda_A = 2,\ x_A = (t,t,0),\ t > 0$;

в) $\lambda_A = 8,\ x_A = (3t,2t,2t),\ t > 0$; г) $\lambda_A = 4,\ x_A = (3t,0,2t),\ t > 0$;

3. $A = \begin{pmatrix} 0,15 & 1,2 \\ 0,25 & 0,1 \end{pmatrix}$, $X = (103,23; 56,45)$.

4. $X = (135,7; 69,38; 81,38)$. **5.** $A = \begin{pmatrix} 0,9 & 0,1 \\ 0,2 & 0,4 \end{pmatrix}$.

6. $X = (190; 120)$; объемы валового выпуска изменятся на 15,79% и 33,33%.

7. **а)** $\vec{p} = (17,9; 10,2; 16,58)$; **б)** 8,72%, 3,82%, 3,32%.

8. **а)** A продуктивна; **б)** A не продуктивна; **в)** A продуктивна.

9. *Указание*. **а)** Сумма *элементов* каждого столбца меньше единицы; **б)** Сумма элементов в каждой строке меньше 1. $\lambda_A = 0,9$; $\alpha = 1/9$.

10. $a < \dfrac{1}{6}$. **14.** $\begin{cases} z = x_3 - 6x_4 \to \max \\ 7x_3 - 9x_4 \le -6, \\ 5x_3 - 6x_4 \le -3, \\ x_3 \ge 0, x_4 \ge 0. \end{cases}$ **15.** $\begin{cases} 27x_3 - 17x_4 + 30 \to \min \\ -6x_3 + 5x_4 \le 7, \\ -9x_3 + 6x_4 \le 10, \\ x_3 \ge 0, x_4 \ge 0. \end{cases}$

16. $\begin{cases} -28x_3 - 45x_4 - 42 \to \max \\ 7x_3 + 11x_4 \le -10 \\ -8x_3 - 13x_4 \le 12 \\ x_3 \ge 0, x_4 \ge 0. \end{cases}$ **17.** Если x_i – количество изделий i-го вида $(i = 1,2)$,

то $\begin{cases} f = 3000x_1 + 2000x_2 \to \max \\ 2x_1 + 4x_2 \le 100, \\ x_1 \le 40, x_2 \le 20, \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$ **18.** Если x_1, x_2 – суточное потребление

яблок и абрикосов (в кг), то $\begin{cases} f = 30x_1 + 60x_2 \to \min \\ 70x_1 + 74x_2 \ge 70, \\ x_1 + 24x_2 \ge 2, \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$.

20. Если x_1, x_2 – количество произведенного чая сорта А и Б соответственно, то $\begin{cases} f = 320x_1 + 290x_2 \to \max \\ 0,5x_1 + 0,2x_2 \le 600, \\ 0,2x_1 + 0,6x_2 \le 870, \\ 0,3x_1 + 0,2x_2 \le 430, \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$

21. Если x_1, x_2 – количество машин типа А и Б, отремонтированных за год, то $f = 10^3 \cdot \left(20x_1 + 24x_2\right) \to \max$ при $\begin{cases} \dfrac{x_1}{80} + \dfrac{x_2}{320} \le 1, \\ \dfrac{x_1}{110} + \dfrac{x_2}{110} \le 1, \\ \dfrac{x_1}{240} + \dfrac{x_2}{120} \le 1, \\ \dfrac{x_1}{160} + \dfrac{x_2}{80} \le 1 \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$

22. Если x_1, x_2 – число скорых и пассажирских поездов, то

$$f = 626x_1 + 656x_2 \to \max \text{ при } \begin{cases} x_1 + x_2 \le 12, \\ x_1 \le 8, \\ 5x_1 + 8x_2 \le 81, \\ 6x_1 + 4x_2 \le 70, \\ 3x_1 + x_2 \le 26, \\ x_1 \ge 0, x_2 \ge 0. \end{cases}$$

23. $z_{\max} = z(5,10) = 50$, $z_{\min} = z(4,1) = 19$. **24.** $z_{\max} = z(3,8) = 7$, $z_{\min} = z(8,3) = -18$.

25. $z_{\max} = z(3,3) = -21$, $z_{\min} = z(5,13) = -59$. **26.** $z_{\max} = z(10,1) = 7$, $z_{\min} = z(12,10) = -18$.

27. $z_{\max} = +\infty$, $z_{\min} = z(4,6) = 16$. **28.** $z_{\max} = z(11,8) = -57$, $z_{\min} = -\infty$.

29. $z_{\max} = +\infty$, $z_{\min} = z(0,1,3,0,7) = -2$. **30.** $z_{\max} = z(3,0,1,0) = 27$, $z_{\min} = z(2,1,0,0) = 25$.

31. $z_{\max} = z(2,0,8,0,1) = 5$, $z_{\min} = z(X^*) = -4$, где $X^* = (4 - 4t, 2t + 3, 0,5 - 2t, 10t), t \in [0,1]$

32. $z_{\max} = z(0,9,0,12) = 40$, $z_{\min} = z(X^*) = 16$, где $X^* = (6 + 2t, t, 3 - 3t, 0), t \in [0,1]$.

33. $z_{\max} = z(X^*) = 40$, где $X^* = (6 + 3t, 4 + t, 0, 5t), t \ge 0$, $z_{\min} = -\infty$.

34. $z_{\max} = z(8,13,0,0,65) = 8$, $z_{\min} = z(11,8,65,0,0) = -187$. **35.** $z_{\max} = z(1,0,3,0,1) = 26$.
36. $z_{\min} = z(2,6,33,0,0) = 11$. **37.** $z_{\min} = z(2,5,0,0,17) = 10$. **38.** $z_{\max} = z(6,7,3,0) = 29$.
39. $z_{\min} = z(X^*) = -7$, где $X^* = (3t,0,3-3t,5-3t),\ t \in [0,1]$.
40. $z_{\max} = z(X^*) = 12$, где $X^* = \left(\frac{3}{2} - \frac{3}{2}t, 2+2t, 4t, 4+8t\right),\ t \in [0,1]$. **41.** $z_{\min} = z(1,0,0,4) = -4$.
42. $z_{\max} = z(0,1,1,1,0) = 19$. **43.** $z_{\max} = z(0,0,2,2,2) = 4$. **44.** $z_{\min} = z(0,0,12,1,3) = -34$.
45. $z_{\max} = z(1,0,3,0,3) = 40$. **46.** $z_{\min} = z(0,0,2,12,6) = 1$. **47.** $z_{\max} = z(2,1,0,5,0) = 36$.
53. $z_{\min} = z(2,1,0) = 156$. **54.** $z_{\min} = z(3,2,0) = 11$. **55.** $z_{\min} = z(2,1,0) = 148$.
56. $z_{\min} = z(1,1,0) = 34$. **57.** $z_{\min} = z(3,1,0) = 67$. **58.** $z_{\min} = z(3,2,0) = 12$.
59. $z_{\max} = z(0,0,7,0,1) = T(3,\frac{7}{2}) = 29$. **60.** $z_{\max} = z(1,0,7,0,0) = T(1,0) = 17$.
61. $z_{\max} = z(0,7,0,10,0) = T(2,1) = 26$. **62.** $z_{\max} = z(6,0,0,0,1) = T(1,\frac{6}{5}) = 33$.
63. $z_{\max} = z(10,0,2,0,0) = T(11,10) = 642$. **64.** $z_{\max} = z(0,24,0,3,0) = T(13,\frac{98}{5}) = 335$.

65. $X^* = \begin{pmatrix} 0 & 60 & 0 & 100 \\ 60 & 0 & 0 & 0 \\ 20 & 0 & 60 & 100 \end{pmatrix},\ z(X^*) = 1880$. **66.** $X^* = \begin{pmatrix} 0 & 10 & 0 & 70 \\ 50 & 0 & 0 & 0 \\ 50 & 0 & 40 & 90 \end{pmatrix},\ z(X^*) = 3420$.

67. $X^* = \begin{pmatrix} 0 & 10 & 0 & 80 \\ 60 & 0 & 0 & 0 \\ 20 & 30 & 90 & 0 \end{pmatrix},\ z(X^*) = 2750$. **68.** $X^* = \begin{pmatrix} 0 & 70 & 0 & 90 \\ 30 & 0 & 30 & 20 \\ 70 & 0 & 0 & 0 \end{pmatrix},\ z(X^*) = 3300$.

69. $X^* = \begin{pmatrix} 0 & 50 & 0 & 20 \\ 60 & 0 & 0 & 0 \\ 30 & 0 & 100 & 10 \end{pmatrix},\ z(X^*) = 2530$. **70.** $X^* = \begin{pmatrix} 30 & 0 & 0 & 0 \\ 0 & 0 & 30 & 100 \\ 20 & 70 & 0 & 30 \end{pmatrix},\ z(X^*) = 2990$.

71. $X^* = \begin{pmatrix} 80 & 0 & 0 & 40 \\ 0 & 20 & 70 & 50 \\ 0 & 70 & 0 & 0 \end{pmatrix},\ z(X^*) = 4060$. **72.** $X^* = \begin{pmatrix} 0 & 50 & 0 & 100 \\ 70 & 0 & 0 & 0 \\ 70 & 0 & 100 & 40 \end{pmatrix},\ z(X^*) = 4720$.

73. $X^* = \begin{pmatrix} 0 & 80 & 0 & 20 \\ 0 & 0 & 0 & 100 \\ 90 & 0 & 30 & 50 \end{pmatrix},\ z(X^*) = 3380$. **74.** $X^* = \begin{pmatrix} 25 & 0 & 0 & 35 \\ 75 & 0 & 50 & 0 \\ 0 & 75 & 0 & 0 \end{pmatrix},\ z(X^*) = 780$.

75. $X^* = \begin{pmatrix} 40 & 0 & 35 & 0 \\ 0 & 0 & 35 & 115 \\ 0 & 70 & 20 & 0 \end{pmatrix},\ z(X^*) = 1750$. **76.** $X^* = \begin{pmatrix} 0 & 45 & 0 & 50 \\ 0 & 65 & 0 & 0 \\ 0 & 0 & 85 & 45 \end{pmatrix},\ z(X^*) = 1535$.

77. $X^* = \begin{pmatrix} 0 & 0 & 85 & 0 \\ 60 & 0 & 0 & 90 \\ 0 & 95 & 0 & 0 \end{pmatrix},\ z(X^*) = 1560$. **78.** $X^* = \begin{pmatrix} 80 & 0 & 40 & 0 \\ 0 & 30 & 0 & 50 \\ 0 & 70 & 0 & 0 \end{pmatrix},\ z(X^*) = 1040$.

79. $X^* = \begin{pmatrix} 10 & 0 & 90 & 0 \\ 0 & 100 & 0 & 0 \\ 70 & 10 & 0 & 120 \end{pmatrix},\ z(X^*) = 2430$. **80.** $X^* = \begin{pmatrix} 30 & 20 & 0 & 70 \\ 0 & 90 & 0 & 0 \\ 65 & 0 & 80 & 35 \end{pmatrix},\ z(X^*) = 4315$.

81. $X^* = \begin{pmatrix} 75 & 35 & 40 & 0 \\ 0 & 110 & 30 & 0 \\ 0 & 0 & 50 & 60 \end{pmatrix},\ z(X^*) = 3145$. **82.** $X^* = \begin{pmatrix} 20 & 50 & 0 & 70 \\ 0 & 80 & 0 & 25 \\ 40 & 0 & 55 & 20 \end{pmatrix},\ z(X^*) = 3160$.

83. $X^* = \begin{pmatrix} 0 & 0 & 20 & 50 \\ 20 & 40 & 110 & 0 \\ 70 & 20 & 30 & 20 \end{pmatrix}, z\left(X^*\right) = 5070$. **84.** $X^* = \begin{pmatrix} 0 & 0 & 50 & 40 \\ 40 & 75 & 25 & 15 \\ 60 & 0 & 0 & 65 \end{pmatrix}, z\left(X^*\right) = 2995$.

Заключение

Авторы издания надеются, что после чтения данного учебного пособия и решения предлагаемых задач, студенты смогут не только успешно пройти все элементы текущего и промежуточного контроля, предусмотренные рабочей программой изучаемой дисциплины – контрольные и самостоятельные работы, курсовые и лабораторные работы, экзамен или зачет и т.д.

Исключительно важным, по нашему мнению, является то, что представленные здесь модели и методы необходимы не только при изучении последующих разделов прикладной математики, но и широко востребованы при применении количественных методов во многих разделах экономики и финансов при решении практических задач.

Безусловно, представленный в учебном пособии материал является вполне доступным для студентов первого курса экономических и математических направлений подготовки бакалавров. Поэтому студент, освоивший методику построения простейших экономических задач оптимизации, основные методы решения естественно возникающих при этом задач линейного программирования (графический метод, симплекс-метод) и ключевые идеи теории двойственности может считать себя овладевшим базовыми идеями применения количественных методов.. Их дальнейшее развитие находит свое отражение в последующих прикладных дисциплинах, читаемых в Финансовом университете, а также в практической деятельности в различных областях экономики, финансов и менеджмента.

Рекомендуемая литература

1. **Акулич И.Л.** *Математическое программирования в примерах и задачах*. М.: Высшая школа, 1993.

2. *Методы оптимальных решений в экономике и финансах*. Учебник/ И.А. Александрова [и др.]; под ред. В.М. Гончаренко, В.Ю. Попова. М.: Кнорус, 2013 .

3. *Сборник задач по курсу «Математика в экономике». Часть 1. Линейная алгебра и линейное программирование.* Учебное пособие/ Е.Н. Орел [и др.]; под редакцией В.А. Бабайцева и В.Б. Гисина. М.: Финансы и статистика, 2010.

4. **Васильев А.Н.** *Финансовое моделирование и оптимизация средствами EXCEL 2007*. Спб. ПИТЕР, 2009.

5. **Винюков И.А., Попов В.Ю., Пчелинцев С.В.** *Многочлены и комплексные числа. Собственные значения и собственные векторы. Модель Леонтьева. Учебное пособие для подготовки бакалавров*. М.: Финансовая академия при Правительстве РФ, 2009.

6. **Винюков И.А., Попов В.Ю., Пчелинцев С.В.** *Линейное программирование. Учебное пособие для подготовки бакалавров.* М.: Финансовая академия при Правительстве РФ, 2009.

7. **Грешилов А.А.** *Как принять наилучшее решение в реальных условиях*. М.: Радио и связь, 1991.

8. **Зайченко Ю.П.** *Исследование операций.* К.: Вища школа, 1979.

9. **Интрилигатор М.** *Математические методы оптимизации и экономическая теория.* М.: Айрис-Пресс, 2002.

10. **Колемаев В.А.** *Математические методы и модели исследования операций. Учебник.* М.: ЮНИТИ, 2008.

11. **Красс М.С., Чупрынов Б.П.** *Основы математики и ее приложения в экономическом образовании.* М.: Дело, 2002.

12. **Кремер Н.Ш.** *Исследование операций в экономике.* М.: ЮНИТИ, 1996.

13. **Романовский И.В.** *Алгоритмы решения экстремальных задач*. М.: Наука, 1977

14. **Солодовников А.С., Бабайцев В.А., Браилов А.В., Шандра И.Г.** *Математика в экономике. Ч.1*. М.: Финансы и статистика, 2007.

15. **Ху Т.** *Целочисленное программирование и потоки в сетях*. М.: Мир, 1974.

16. **Харари Ф.** *Теория графов*. М.: Мир, 1973

17. **Цисарь И.Ф., В.Г. Нейман** *Компьютерное моделирование экономики*. М.ДИАЛОГ МИФИ, 2002.

ОГЛАВЛЕНИЕ