Министерство образования
Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра высшей математики № 2

ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Методические указания и контрольные задания для студентов экономических специальностей БНТУ

Минск 2006

Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра высшей математики №2

ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Методические указания и контрольные задания для студентов экономических специальностей БНТУ

Под редакцией

А.Д. КОРЗНИКОВА

Минск 2006

УДК 519.85 (075.8)
~~ББК 18я7~~
Э 45

Составители:
А.Д. Корзников, Л.Д. Матвеева, М.Б. Смирнов

Рецензенты:
В.Ф. Бубнов, А.Н. Рудый

Методические указания включают в себя изложение теоретических методов решения основных задач математического программирования, а также контрольные задания по каждой теме для самостоятельного решения. Указания состоят из восьми разделов, по каждому из которых предлагается 10 задач. В каждом разделе описывается теоретическое обоснование метода, формальный алгоритм и пример решения типовой задачи.

Указания предназначены для студентов экономических специальностей заочного отделения БНТУ, они могут быть также полезны преподавателям, ведущим практические занятия по курсу математического программирования.

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ ЖОРДАНА-ГАУССА

Системой линейных алгебраических уравнений называется совокупность формальных равенств вида:

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

где a_{ij}, $b_i \in R$ - заданные числа, x_j - неизвестные, $1 \le i \le m$, $1 \le j \le n$.

Матрицы $A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$ и $\overline{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{pmatrix}$,

называются соответственно *матрицей системы* и *расширенной матрицей системы.*

Решением системы (1) называется упорядоченная совокупность чисел $X = (c_1, c_2, \ldots, c_n)$, которые при подстановке $c_j \leftrightarrow x_j$ ($j = 1, \ldots, n$) обращают каждое уравнение системы (1) в верное равенство. Система, имеющая хотя бы одно решение, называется *совместной*, иначе – *несовместной*. Решить систему – означает найти все ее решения. Две системы называются *эквивалентными* или *равносильными*, если они имеют одинаковые множества решений. Аналогично, расширенные матрицы эквивалентных систем будем называть эквивалентными.

Например, системы

$$S: \begin{cases} 2x_1 + x_2 = 4, \\ 5x_1 - 2x_2 = 1, \end{cases} \quad S_1: \begin{cases} 2x_1 + x_2 = 4, \\ 9x_1 = 9, \end{cases} \quad S_2: \begin{cases} x_2 = 2, \\ x_1 = 1, \end{cases}$$

с расширенными матрицами

$$\overline{A}=\begin{pmatrix}2 & 1 & 4\\ 5 & -2 & 1\end{pmatrix},\qquad \overline{A_1}=\begin{pmatrix}2 & 1 & 4\\ 9 & 0 & 9\end{pmatrix},\qquad \overline{A_2}=\begin{pmatrix}0 & 1 & 2\\ 1 & 0 & 1\end{pmatrix},$$

являются эквивалентными, так как все они имеют единственное решение $X = (1, 2)$.

Элементарными преобразованиями матрицы называются: перестановка местами любых двух строк; умножение строки на любое, отличное от нуля число; прибавление к одной строке матрицы любой другой строки, умноженной на любое число; удаление нулевой строки.

Решение системы методом Гаусса и его модификацией – методом Жордана-Гаусса основано на следующем утверждении: матрица, полученная элементарными преобразованиями расширенной матрицы системы эквивалентна исходной матрице, т.е. элементарные преобразования расширенной матрицы системы не изменяют множества решений системы.

Суть обоих методов состоит в том, чтобы при помощи элементарных преобразований привести расширенную матрицу системы к наиболее простому виду, т.е. к такому виду, когда решение найти достаточно легко. Например, ясно, что систему S_1 с матрицей $\overline{A_1}$ решить легче, чем исходную систему S с матрицей $\overline{A}$, а решение системы S_2 вообще очевидно. Переход от матрицы $\overline{A}$ к матрице $\overline{A_1}$ можно осуществить, например, прибавляя ко второй строке матрицы $\overline{A}$, первой строки, умноженной на 2. Чтобы из матрицы $\overline{A_1}$ получить $\overline{A_2}$, можно поступить следующим образом: сначала вторую строку $\overline{A_1}$ умножим на 1/9, а затем к первой строке прибавим вторую, умноженную на -2.

Переменная x_j называется **базисной** в i-м уравнении системы (1) если

$$a_{ij} = 1 \text{ и } a_{kj} = 0 \text{ при } k \neq i,\ k = 1, 2, \ldots, m.$$

Другими словами, переменная x_j вляется базисной в i-м уравнении, если коэффициент при ней в этом уравнении равен 1, а в остальных уравнениях - 0, т.е. в других уравнениях этой переменной нет.

Говорят, что матрица системы приведена к базисному виду (или имеет базис) если в каждом ее уравнении имеется базисная переменная. Например, матрица $\overline{A}$ системы S не имеет ни одной базисной переменной, матрица $\overline{A_1}$ имеет базисную переменную x_2 в первом уравнении, а матрица $\overline{A_2}$ приведена к базисному виду.

Справедливо следующее утверждение: При помощи элементарных преобразований расширенную матрицу любой совместной системы можно привести к базисному виду.

Если матрица системы приведена к базисному виду, то переменные, не являющиеся базисными, называются **свободными.** Например, в матрице $\overline{A_2}$ все переменные – базисные, свободных нет.

Решение системы, полученное после приравнивания нулю всех свободных переменных, называется **базисным.**

Алгоритм приведения матрицы к базисному виду

Каждая итерация алгоритма состоит из трех шагов:

Шаг 1. В первой строке матрицы находим ненулевой элемент $a_{1j} \neq 0$, (желательно, $a_{1j} = 1$) . Если таких нет, то в случае $b_1 = 0$ вычеркиваем нулевую строку; если $b_1 \neq 0$, то, очевидно, система несовместна. Найденный элемент назовем разрешающим (или ведущим).

Если $a_{1j} = 1$, то переходим к шагу 3, иначе к шагу 2.

Шаг 2. Делим первую строку на разрешающий элемент $a_{1j} \neq 0$.

(После этого шага коэффициент при x_j в первом уравнении будет $a_{1j} = 1$).

Шаг 3. Ко всем остальным строкам, кроме первой, прибавляем первую строку, умноженную на ($-a_{ij}$), где i - номер изменяемой строки ($i = 2,3,\dots, m$). После этого шага коэффициент при x_j в остальных уравнениях будет 0 , и переменная x_j станет базисной в первом уравнении. Затем применяем шаги 1 , 2 и 3 ко второму уравнению полученной матрицы и т.д. Так как число уравнений системы конечно, то этот процесс завершится не более чем за m итераций.

Решение системы по этому алгоритму называется *методом Жордана-Гаусса.*

После того, как система приведена к базисному виду, находят базисное решение, соответствующее выбранному базису. Для этого переменные, не вошедшие в базис, приравнивают нулю, а остальные переменные (базисные) находят по правым частям соответствующих уравнений. Приведем решение типового примера задания 1:

Найти базисное решение системы с расширенной матрицей

$$\left(\begin{array}{cccc|c} 2 & 1 & 3 & 1 & 4 \\ 2 & 2 & 1 & 5 & 12 \\ 4 & 5 & 0 & 14 & 32 \end{array}\right).$$

Применим алгоритм приведения матрицы к базисному виду: В первой строке элемент $a_{12}=1$, поэтому выберем его в качестве разрешающего. Теперь изменяем вторую и третью строки следующим образом: ко второй строке прибавляем первую, умноженную на (-2), к третьей прибавляем первую, умноженную на (-5). В результате получим матрицу

$$\left(\begin{array}{cccc|c} 2 & 1 & 3 & 1 & 4 \\ -2 & 0 & -5 & 3 & 4 \\ -6 & 0 & -15 & 9 & 12 \end{array}\right),$$

в которой переменная x_2 стала базисной в первом уравнении. Теперь применяем шаги 1-3 ко второй строке полученной матрицы. Находим ненулевой элемент, например, $a_{24} = 3$, и делим вторую строку на этот элемент. Получим матрицу

$$\left(\begin{array}{cccc|c} 2 & 1 & 3 & 1 & 4 \\ -2/3 & 0 & -5/3 & 1 & 4/3 \\ -6 & 0 & -15 & 9 & 12 \end{array}\right).$$

Теперь делаем нули в остальных строках четвертого столбца этой матрицы, для чего к первой строке прибавляем вторую, умноженную на -1, к третьей прибавляем вторую, умноженную на -9. В результате расширенная матрица системы примет вид:

$$\left(\begin{array}{cccc|c} 8/3 & 1 & 14/3 & 0 & 8/3 \\ -2/3 & 0 & -5/3 & 1 & 4/3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Вычеркивая нулевую третью строку, получим матрицу, в базисном виде:

$$\left(\begin{array}{cccc|c} \left[8/3\right] & 1 & 14/3 & 0 & 8/3 \\ -2/3 & 0 & -5/3 & 1 & 4/3 \end{array}\right).$$

В первой строке базисной является переменная x_2 , а во второй – переменная x_4. Переменные x_1 и x_3 являются свободными. Приравнивая их нулю, получаем базисное решение, соответствующее этому базису: $x_1 = x_3 = 0$, $x_2 = 8/3$, $x_4 = 4/3$ или $X_1 = (0, 8/3, 0, 4/3)$. Найдем другое базисное решение, т.е. решение, в котором базисными являются другие переменные. В базис можно включить переменные x_1 или x_3 , которые сейчас являются свободными. Выберем, например, переменную x_1 для включения в базис. Ее можно сделать базисной в первой строке, т.к. элемент $a_{11} = 8/3 \neq 0$ (при этом из базиса выйдет переменная x_2), или во второй строке $a_{21} = -2/3 \neq 0$ (при этом из базиса выйдет x_4). Будем делать x_1 базисной, например, в первой строке, т.е. в качестве разрешающего выберем элемент $a_{11} = 8/3 \neq 0$ (помечен в последней матрице). Как и раньше, разделив первую строку на разрешающий элемент и прибавив ко второй строке полученную первую, умноженную на 2/3, приведем матрицу к новому базису:

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

Полагая свободные переменные x_2 и x_3 равными нулю, получим новое базисное решение $X_2 = (1, 0, 0, 2)$.

Контрольные задания для самостоятельного решения

Задание 1. Найти целочисленное базисное решение системы с заданной расширенной матрицей:

Варианты:

1. $\left(\begin{array}{cccc:c} 0 & 1 & 0 & -1 & -1 \\ -3 & 3 & -2 & 4 & 1 \\ 0 & 1 & 0 & 3 & -1 \end{array}\right)$,

2. $\left(\begin{array}{cccc:c} 2 & 2 & 0 & 1 & 8 \\ 3 & 0 & 3 & -2 & 12 \\ -3 & 4 & 0 & -1 & 2 \end{array}\right)$,

3. $\left(\begin{array}{cccc:c} 4 & -3 & 3 & 2 & 9 \\ 3 & 3 & -4 & 1 & 12 \\ 11 & -3 & 2 & 5 & 30 \end{array}\right)$,

4. $\left(\begin{array}{cccc:c} 1 & 3 & 1 & 3 & -2 \\ 0 & 3 & 3 & 1 & 0 \\ 2 & 9 & 5 & 7 & -4 \end{array}\right)$,

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

6. $\left(\begin{array}{cccc:c} 4 & -3 & 1 & 1 & 3 \\ 1 & 1 & 1 & -1 & -3 \\ 3 & 1 & -2 & -1 & -13 \end{array}\right)$,

7. $\left(\begin{array}{cccc:c} -2 & 2 & -3 & 4 & 12 \\ 3 & -1 & -1 & 1 & -7 \\ 2 & -2 & -3 & 2 & 0 \end{array}\right)$,

8. $\left(\begin{array}{cccc:c} 0 & 3 & 4 & 2 & 10 \\ 0 & 2 & -3 & 2 & 1 \\ 3 & 4 & -1 & 3 & 10 \end{array}\right)$,

9. $\left(\begin{array}{cccc:c} 0 & 2 & 0 & -1 & -6 \\ -3 & 4 & 4 & 0 & -20 \\ 2 & -3 & 1 & -4 & 7 \end{array}\right)$,

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

РЕШЕНИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ГЕОМЕТРИЧЕСКИМ МЕТОДОМ

Общей задачей линейного программирования называется задача нахождения минимума (максимума) линейной функции

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \to \min (\max) \tag{1}$$

при ограничениях :

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

$$x_j \ge 0 \ , j = 1, \ldots , n, \tag{3}$$

где c_j, a_{ij}, b_i - заданные числа, x_j - неизвестные, $i = 1,...,m, j = 1,...,n$ и в любом из ограничений вида (2) может встречаться любой из знаков $\le$, $=$ или $\ge$.

Если число неизвестных n = 2, то задача (1) – (3) примет вид

$$Z = c_1 x + c_2 y \to \min (\max), \tag{4}$$

$$\begin{cases} a_{11} x + a_{12} y \le (=, \ge)\ b_1 , \\ a_{21} x + a_{22} y \le (=, \ge)\ b_2 , \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1} x + a_{m2} y \le (=, \ge)\ b_m , \end{cases} \tag{5}$$

$$x \ge 0 \ , y \ge 0, \tag{6}$$

и ее можно решить геометрическим методом, так как каждая пара неизвестных (x , y) может быть представлена точкой на координатной плоскости xOy.

При решении задачи (4) – (6) сначала строят так называемую допустимую область, т.е. множество точек (x, y) плоскости, координаты которых удовлетворяют всем ограничениям (5) и лежат в первой четверти координатной плоскости (ограничение (6)). Поскольку все ограничения (5) – линейные, то допустимая область будет представлять собой выпуклый многоугольник (конечный или бесконечный) или пустое множество.

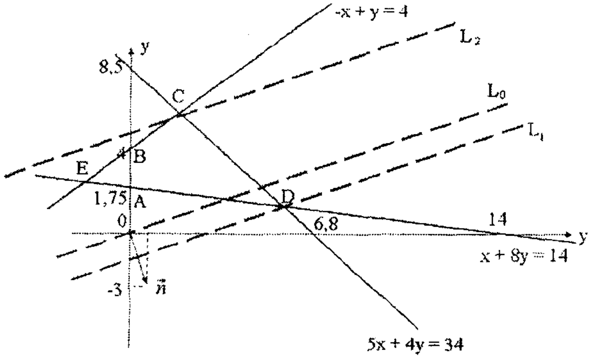
Затем среди точек допустимой области находят оптимальную, т.е. такую точку M_0 координаты которой (x_0 , y_0) доставляют минимум (максимум) целевой функции Z. Для этого по виду целевой функции (4) строят линию уровня функции Z, соответствующую Z=0, т.е. прямую L_0: $c_1\ x + c_2\ y = 0$ и находят градиент функции Z – вектор $\overrightarrow{gradZ}=\overrightarrow{\nabla Z}=\left(\frac{\partial Z}{\partial x},\frac{\partial Z}{\partial y}\right)=(c_1,c_2)$, который показывает направление наибыстрейшего возрастания функции Z. Вектор антиградиента $(-c_1,\ -c_2)$ будет показывать направление наибыстрейшего убывания целевой функции Z. Вектор градиента перпендикулярен линии уровня L_0 .

Перемещая линию уровня L_0 параллельно самой себе в направлении градиента $(c_1,\ c_2)$, находим последнюю точку допустимой области, которую она пересекает при таком движении. Очевидно, что это будет точка максимума. Перемещая линию уровня в противоположном направлении $(-c_1,\ -c_2)$, находим точку минимума. Поясним этот метод на конкретном примере.

Геометрическим методом найти максимум и минимум функции Z для $x,\ y \geq 0$ при заданных ограничениях

$$\begin{cases} Z = x - 3y, \\ -x + y \leq 4, \\ 5x + 4y \leq 34, \\ x + 8y \geq 14. \end{cases}$$

Р е ш е н и е. Построим допустимую область. Для этого в системе координат $x0y$ строим прямую $-x + y = 4$ - границу первого ограничения. Затем определяем, в какой полуплоскости находятся точки $(x,\ y)$, для которых $-x + y < 4$. Для этого выбираем любую точку, например (0, 0), и проверяем, удовлетворяет ли она этому неравенству. Поскольку $-0 + 0 = 0 < 4$, то точки, для которых $-x+y\leq 4$ лежат в той же полуплоскости, что и точка O(0, 0), т.е. справа (ниже) от границы $-x + y = 4$.



Аналогично строятся остальные полуплоскости, соответствующие ограничениям $5x + 4y \leq 34$ и $x + 8y \geq 14$ (ниже прямой $5x + 4y = 34$ и выше прямой $x + 8y = 14$). Множество точек, удовлетворяющих всем трем неравенствам, образуют треугольник ECD. Учитывая ограничение $x \geq 0$ и $y \geq 0$, получаем допустимую область – четырехугольник $ABCD$.

Проведем линию уровня L_0, соответствующую значению $Z=0$, т.е. прямую $x - 3y = 0$ (для точек, лежащих на этой прямой, значение $Z=0$). Она будет проходить через точку $O(0, 0)$ перпендикулярно вектору $\vec{n} = \overrightarrow{\nabla Z} = (1,-3)$. Перемещая линию уровня в направлении градиента $\vec{n}$, т.е. в направлении возрастания Z, находим последнюю точку допустимой области, которую линия уровня пересекает при этом движении (линия L_1). Это будет точка максимума. В нашем случае – это точка $D(6, 1)$, координаты которой можно найти, решив систему линейных уравнений $5x + 4y = 34$ и $x + 8y = 14$. Аналогично, двигая линию уровня в противоположном направлении до линии L_2, находим точку минимума – точку $C(2, 6)$. Таким образом, $Z_{max} = Z(6,1) = 6 - 3 \cdot 1 = 3$, $Z_{min} = Z(2,6) = 2 - 3 \cdot 6 = -16$.

Задача решена полностью.

Контрольные задания для самостоятельного решения

Задание 2. Геометрическим методом найти максимум и минимум функции Z для x, $y \geq 0$ при заданных ограничениях.

Варианты

1. $Z = 2x + y$,
$$\begin{cases} 3x - 4y \le 4, \\ -x + 6y \le 8, \\ x + y \ge 1, \end{cases}$$

2. $Z = 3x + y$,
$$\begin{cases} y \le 1, \\ x - 2y \le 3, \\ 2x + y \ge 1, \end{cases}$$

3. $Z = 2x - y$,
$$\begin{cases} 2x + y \le 3, \\ x + y \ge 6, \\ x - 3y \le 3, \end{cases}$$

4. $Z = x + y$,
$$\begin{cases} y \ge 2, \\ x + y \le 7, \\ -x + 2y \le 2, \end{cases}$$

5. $Z = 4x + y$,
$$\begin{cases} x - 2y \ge 0, \\ 4x - y \le 14, \\ 3x + y \ge 7, \end{cases}$$

6. $Z = 3x - y$,
$$\begin{cases} 2x + 3y \le 13, \\ x \ge 2, \\ 5x - 3y \le 22, \end{cases}$$

7. $Z = 2x + y$,
$$\begin{cases} x \ge 2, \\ 4x - y \le 8, \\ x - y \ge -1, \end{cases}$$

8. $Z = x + 5y$,
$$\begin{cases} 3x - y \ge 3, \\ x - y \le 4, \\ x + y \ge 6, \end{cases}$$

9. $Z = 5x + y$,
$$\begin{cases} x - 4y \ge -3, \\ 4x - 3y \le 14, \\ 3x + y \ge 4, \end{cases}$$

10. $Z = 3x$,
$$\begin{cases} x + y \le 7, \\ 2x - y \le 11, \\ 4x + y \ge 19. \end{cases}$$

РЕШЕНИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ СИМПЛЕКС-МЕТОДОМ

Задача линейного программирования (ЗЛП) (1) - (3) (см. задание 2) называется *канонической*, если все ограничения вида (2) являются уравнениями (равенствами), т.е. задачей линейного программирования в канонической форме называется задача:

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \to \min (\max) \qquad (1)$$

при ограничениях :

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

$$x_j \geq 0,\ j = 1, \ldots, n. \qquad (3)$$

Симплекс-метод является вычислительной процедурой, которая позволяет решить любую каноническую ЗЛП алгебраическим методом. Теоретической основой метода являются следующие два утверждения:

Теорема 1. Если ЗЛП имеет допустимые решения, то существует хотя бы одно базисное допустимое решение задачи.

Теорема 2. Если ЗЛП имеет конечное оптимальное решение, то хотя бы одно из оптимальных решений является базисным.

Напомним, что решение системы (2) называется допустимым, если оно удовлетворяет ограничениям (3). Таким образом, из теоремы 1 следует, что если не существует ни одного базисного допустимого решения системы (2), то эта система вообще не имеет допустимых решений, т.е. ограничения (2) – (3) являются несовместными. В случае если имеются допустимые решения системы 2, то теорема 2 утверждает, что *оптимальное* решение ЗЛП можно найти среди *базисных* допустимых решений этой системы, которых, как мы знаем, конечное число. Суть симплекс-метода и состоит в последовательном переборе базисных допустимых решений системы (2), начиная с некоторого начального, которое еще называют первоначальным опорным планом. Перебор осуществляется таким образом, чтобы каждое новое решение было лучше предыдущего (в смысле приближения к максимуму или минимуму целевой функции Z).

Итак, для применения метода необходимо, чтобы задача была в канонической форме, и чтобы существовало начальное базисное допустимое решение (опорный план), наличие которого гарантирует непротиворечивость ограничений задачи.

Чтобы преобразовать ЗЛП к каноническому виду к правой части каждого ограничения-неравенства прибавляют или вычитают *неотрицательную* дополнительную переменную, например, ограничение $3x_1 + 5x_2 - 2x_3 \leq 7$ преобразуется в уравнение $3x_1 + 5x_2 - 2x_3 + x_4 = 7$ прибавлением дополнительной переменной $x_4 \geq 0$, а неравенство вида $x_1 - 2x_2 + x_3 \geq 5$ заменяется на уравнение $x_1 - 2x_2 + x_3 - x_5 = 5$, где $x_5 \geq 0$.

Заметим, что знак неравенства $\geq$ можно заменить на $\leq$ умножением всего неравенства на -1, например, неравенство $x_1 - 3x_2 \geq -6$ эквивалентно неравенству $-x_1 + 3x_2 \leq 6$. Отметим также, что максимизацию целевой функции Z можно заменить на минимизацию функции $-Z$ и наоборот, так как максимум функции

$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$ достигается в тех же точках, что и минимум функции $-Z = -c_1 x_1 - c_2 x_2 - \ldots - c_n x_n$.

Мы приведем алгоритм симплекс-метода для минимизации целевой функции (1) $Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$. Поясним суть метода на следующем примере:

Пусть требуется решить следующую ЗЛП: Найти максимум функции

$$Z = 3x_1 + 4x_2 \to \max,$$

при ограничениях:

$$\begin{cases} 3x_1 + x_2 \leq 9, \\ x_1 + 2x_2 \leq 8, \end{cases}$$
$$x_1, x_2 \geq 0.$$

Приведем задачу к каноническому виду и заменим максимизацию целевой функции Z на минимизацию функции $Z' = -Z$. Получим следующую задачу:

$$Z' = -3x_1 - 4x_2 + 0x_3 + 0x_4 \rightarrow \min, \qquad (4)$$

$$\begin{cases} 3x_1 + x_2 + x_3 = 9, \\ x_1 + 2x_2 + x_4 = 8, \end{cases} \qquad (5)$$

$$x_1, x_2, x_3, x_4 \geq 0. \qquad (6)$$

Ясно, что переменные x_3 и x_4 являются базисными в системе (5) и соответствующее базисное решение $X_0 = (0, 0, 9, 8)$ является допустимым, т.к. все $x_j \geq 0$. При этом $Z_0 = 0$.

Начнем с того, что проверим опорный план X_0 на оптимальность. Поскольку целевая функция $Z' = -3x_1 - 4x_2 + 0x_3 + 0x_4$ выражена через свободные переменные x_1 и x_2, коэффициенты при которых отрицательны, то, очевидно, увеличение этих переменных приведет к уменьшению значения целевой функции на 3 ед. при увеличении x_1 на одну единицу и на 4 ед. при увеличении на одну единицу x_2 (сейчас x_1 и x_2 равны 0). Поэтому делаем вывод, что опорный план X_0 не является оптимальным (т.к. введение в базис переменных x_1 и x_2 приведёт к уменьшению значения целевой функции).

Итак, для того, чтобы получить лучшее базисное решение, мы должны включить в базис переменные x_1 и x_2 . Будем вводить их в базис по очереди. Поскольку увеличение переменной x_2 быстрее уменьшает Z, то выбираем переменную x_2 для включения ее в базис (это разумно, но не обязательно). Теперь надо выяснить в какой строке системы (5) переменная x_2 будет базисной, чтобы полученное новое базисное решение было допустимым.

Анализируя первое уравнение системы $3x_1 + x_2 + x_3 = 9$, замечаем, что поскольку $x_1 = 0$, то увеличение x_2 влечет уменьшение x_3. Так как $x_3 \geq 0$ ввиду (6), то увеличение x_2 возможно лишь до тех пор, пока x_3 не уменьшится до нуля, т.е. до значения $9/1 = 9$. Поскольку переменная x_2 есть также и во втором уравнении, то рассуждая аналогично, заключаем, что в этом случае переменную x_2 можно увеличивать максимум до значения $8/2 = 4$ при котором переменная x_4 уменьшится до нуля.

Новое базисное решение будет допустимым, если мы будем увеличивать переменную x_2 до значения, равного $\min\{9/1,\ 8/2\} = 4$, которое достигается во второй строке системы. Поэтому переменная x_2

должна быть базисной во втором уравнении, элемент $a_{22} = 4$ системы будет разрешающим и, проводя преобразования Жордана-Гаусса, получаем новое (улучшенное) базисное допустимое решение:

$$\left(\begin{array}{cccc|c} 3 & 1 & 1 & 0 & 9 \\ 1 & \boxed{2} & 0 & 1 & 8 \end{array}\right) \cong \left(\begin{array}{cccc|c} 3 & 1 & 1 & 0 & 9 \\ \frac{1}{2} & 1 & 0 & \frac{1}{2} & 4 \end{array}\right) \cong \left(\begin{array}{cccc|c} \frac{5}{2} & 0 & 1 & -\frac{1}{2} & 5 \\ \frac{1}{2} & 1 & 0 & \frac{1}{2} & 4 \end{array}\right),$$

$$X_1 = (\,0,\ 4,\ 5,\ 0\,),\ \ Z_1 = -3\cdot 0 - 4\cdot 4 + 0\cdot 5 + 0\cdot 0 = -16 < Z_0.$$

Теперь повторим эту процедуру для нового опорного плана X_1. Для того, чтобы проверить его на оптимальность, нужно выразить целевую функцию через *свободные* переменные x_1 и x_4 этого плана, поскольку их нужно будет изменять, чтобы получить другое (лучшее) базисное решение. Для этого из второго уравнения последней системы выражаем базисную переменную $x_2 = 4 - ½\, x_1 - ½\, x_4$ и подставляем ее в целевую функцию $Z = -3x_1 - 4\, x_2 + 0x_3 + 0x_4 = -3x_1 - 4\,(4 - ½\, x_1 - ½\, x_4) + 0x_3 + 0x_4 = -16 - x_1 + 0x_2 + 0x_3 + 2x_4$.

Проводимые вычисления удобно оформлять в виде так называемых симплекс-таблиц, которые являются, фактически, расширенными матрицами системы ограничений (5) с добавленной Z – строкой.

Исходная симплекс-таблица.

Базис	x_1	x_2	x_3	x_4	*Значение* (b_i)
x_3	3	1	1	0	9
x_4	1	$\boxed{2}$	0	1	8
- Z	-3	-4	0	0	0

Опорный план $X_0 = (\,0,\ 0,\ 9,\ 8)$, значение Z равно $Z_0 = 0$.

Таблица 1

Базис	x_1	x_2	x_3	x_4	*Значение* (b_i)
x_3	$\boxed{5/2}$	0	1	-½	5
x_2	½	1	0	½	4
- Z	-1	0	0	2	16

План $X_1 = (0, 4, 5, 0)$, значение Z равно $Z_1 = -16$.

Последняя строка симплекс-таблицы называется Z – строкой поскольку в ней расположены коэффициенты целевой функции Z. Заметим, что для того, чтобы выразить коэффициенты целевой функции через свободные переменные, достаточно преобразованиями Жордана - Гаусса сделать нули в базисных столбцах Z – строки. При этом новое значение Z автоматически появится в столбце "Значение" с противоположным знаком.

Поскольку в Z – строке есть отрицательный коэффициент в первом столбце, то план X_1 не оптимален, так как введение в базис свободной переменной x_1 уменьшает Z (введение в базис свободной переменной x_4 увеличивает Z). Поэтому вводим в базис x_1, и первый столбец таблицы будет разрешающим. Чтобы выбрать разрешающую строку, как и раньше делим элементы столбца "Значение" на положительные элементы разрешающего столбца и находим минимальное частное: $min\left\{\frac{5}{5/2}, \frac{4}{1/2}\right\} = 2$ которое достигается в первой строке, которая и будет разрешающей. Значит разрешающим элементом табл. 1 будет элемент $a_{11} = 5/2$ (выделим его прямоугольником). Теперь проведем обычные преобразования Жордана – Гаусса относительно этого элемента, т.е. сначала делим разрешающую строку на разрешающий элемент,

Базис	x_1	x_2	x_3	x_4	*Значение* (b_i)
	$\boxed{1}$	0	2/5	-1/5	2
x_2	½	1	0	½	4
- Z	-1	0	0	2	16

а затем делаем нули на месте всех остальных элементов разрешающего столбца, для чего: ко второй строке прибавляем первую, умноженную на -½ , к Z – строке прибавляем первую. В результате получим новую таблицу

Таблица 2

Базис	x_1	x_2	x_3	x_4	*Значение* (b_i)
x_1	1	0	2/5	-1/5	2
x_2	0	1	-1/5	3/5	3
- Z	0	0	2/5	9/5	18

Новое базисное решение (план) $X_2 = (2, 3, 0, 0)$, значение Z равно $Z_2 = -18$.

Просматривая Z – строку, замечаем, что в ней нет отрицательных элементов. Это означает, что при попытке ввести в базис свободные переменные x_3 или x_4 целевая функция будет увеличиваться (на 2/5 и 9/5 единиц при увеличении на 1 единицу переменных x_3 и x_4 соответственно). Таким образом, других базисных решений, лучших чем X_2, (т.е. с меньшим, чем –18 значением Z) не существует.

Решение $X_2 = (2, 3, 0, 0)$ является оптимальным и $Z_{min} = Z(X_2) = -18$.

Решение исходной задачи: $Z_{max} = 18$ при $x_1 = 2,\ x_2 = 3,\ x_3,\ x_4 = 0$.

Обобщая приведенные выше рассуждения, сформулируем

Алгоритм Симплекс-метода

Исходные данные: задача в канонической форме; целевая функция минимизируется; найдено начальное базисное допустимое решение (опорный план), то есть система уравнений (2) имеет базис и все правые части уравнений b_i 0 - неотрицательны; целевая функция выражена через свободные переменные.

При выполнении этих условий каждая итерация метода состоит из трех шагов:

Шаг 1. Имеющийся план проверяется на оптимальность. Если в Z – строке нет отрицательных элементов, то имеющийся план оптимален и задача решена. Если отрицательные элементы есть, то план не оптимален. Выбираем любой отрицательный элемент Z – строки (как правило, максимальный по модулю) и считаем столбец, в котором он находится в качестве разрешающего. Пусть для определенности это столбец переменной x_s.

Шаг 2. Выбор разрешающей строки. Пусть разрешающий столбец, выбранный на предыдущем шаге, это столбец переменной x_s. Для каждой i-й строки $(i = 1,...,m)$ делим элементы столбца свободных членов "Значение" (напомним, что все они неотрицательные) на *положительные* элементы разрешающего столбца, стоящие в этой строке, и находим минимальное из полученных частных, т.е. находим $\min\limits_{i, a_{is}>0} \frac{b_i}{a_{is}} = \frac{b_r}{a_{rs}}$.

Пусть этот минимум достигается в строке r. Тогда r-ая строка является разрешающей, элемент a_{rs} - разрешающий элемент таблицы.

Ш а г 3. Пересчет таблицы. Преобразованиями Жордана-Гаусса пересчитываем таблицу относительно разрешающего элемента a_{rs}, найденного на предыдущем шаге, для чего:

3.1. Делим разрешающую строку на разрешающий элемент. В результате, на месте элемента a_{rs} будет стоять $a'_{rs} = 1$.

3.2. Ко всем остальным строкам таблицы (включая Z - строку) прибавляем полученную разрешающую, умноженную на элемент $(-a_{is})$, где i – номер изменяемой строки $i = 1,2,...,r-1,r+1,...,m$. К Z – строке прибавляем разрешающую строку, умноженную на $(-c_s)$. Иначе говоря, элементы новой таблицы (со штрихом) вычисляются по формулам:

$a'_{rj} = a_{rj}/a_{rs}$; - новые элементы разрешающей строки $(j = 1, 2, ..., n)$;

$b'_r = b_r/a_{rs}$;

$a'_{ij} = a_{ij} - a'_{rj} \cdot a_{is}$; - новые элементы i-й строки $(i = 1, 2, ..., m;\ j = 1, 2, ..., n)$;

$b'_i = b_i - b'_r \cdot a_{is}$;

$c'_j = c_j - a'_{rj} \cdot c_s$; - новые элементы Z - строки $(j = 1, 2, ..., n)$;

$Z' = Z - b'_r \cdot c_s$.

В результате этих преобразований в столбце x_s везде будут стоять нули кроме r-строки, где будет 1, т.е. x_s будет новой базисной переменной, целевая функция будет выражена через новые свободные переменные, новый план X' находится в столбце "Значение" и лучше предыдущего, так как значение целевой функции для нового плана равно $-Z' < Z$. Переходим к ш а г у 1 и повторяем всю процедуру для нового плана X'.

Поскольку базисных решений системы (2) конечное число, а каждое новое базисное решение лучше предыдущего, то этот процесс завершится за конечное число шагов.

Решение задачи линейного программирования в общем случае. Рассмотрим следующую задачу ЛП:

$$Z = -3x_1 + x_2 \to min,$$

$$\begin{cases} 2x_1 - x_2 \ge 2, \\ -x_1 + 2x_2 \ge 5, \\ x_1 + x_2 \le 10, \end{cases}$$
$$x_1, x_2 \ge 0.$$

Приведем ее к каноническому виду введением трех неотрицательных переменных x_3, x_4, x_5. Получим задачу:

$$Z = -3x_1 + x_2 \to min,$$

$$\begin{cases} 2x_1 - x_2 - x_3 = 2, \\ -x_1 + 2x_2 - x_4 = 5, \\ x_1 + x_2 + x_5 = 10, \end{cases} \tag{7}$$

$$x_1, x_2, x_3, x_4, x_5 \ge 0.$$

При попытке решить эту задачу симплекс-методом возникает определенная трудность, связанная с тем, что нет очевидного начального базисного допустимого решения (опорного плана), так как переменные x_3 и x_4 не являются базисными. Умножение первых двух уравнений на -1 также ничего не дает, поскольку соответствующее базисное решение (0, 0, -2, -5, 10) не будет допустимым. Пытаться просто перебирать базисные решения в попытке отыскать допустимое, нецелесообразно, так как неясно, имеет ли эта задача вообще допустимые решения.

Для решения проблемы применим *метод искусственного базиса*. Введем в первые два уравнения (третье не создает проблем) *искусственные* переменные $x_6 \ge 0$ и $x_7 \ge 0$. В результате получим базис из переменных x_6, x_7, x_5.

$$\begin{cases} 2x_1 - x_2 - x_3 + x_6 = 2, \\ -x_1 + 2x_2 - x_4 + x_7 = 5, \\ x_1 + x_2 + x_5 = 10. \end{cases} \qquad (8)$$

Соответствующее базисное решение $X_0 = (0, 0, 0, 0, 10, 2, 5)$ является допустимым для ограничений (8). Однако ограничения (7) и (8) не являются эквивалентными в том смысле, что любому допустимому решению системы ограничений (8), в котором хотя бы одна искусственная переменная отлична от нуля, нельзя поставить в соответствие допустимое решение системы ограничений (7). С целью исключения искусственных переменных из базисного решения системы ограничений (8), т.е. для получения допустимого базисного решения системы ограничений (7), используем алгоритм симплекс-метода. Для того, чтобы искусственные переменные стали свободными, необходимо, чтобы они были равны нулю (сейчас $x_6 = 2$, $x_7 = 5$). Поэтому введем искусственную целевую функцию

$$W = x_6 + x_7$$

и будем ее минимизировать при ограничениях (8). Если удастся найти базисное допустимое решение при котором $W = 0$ (сейчас $W = 7$), то тогда x_6 и x_7 будут равны нулю, поскольку они неотрицательны, и мы получим базис из основных и дополнительных переменных.

Таким образом, задача разбивается на два этапа. На первом этапе минимизируется искусственная целевая функция W. Этот этап закончится либо нахождением опорного плана исходной задачи (7), либо тем, что минимизировать функцию W до нуля не удастся, т.е. допустимых планов нет, а значит (ввиду **теоремы** 1) и нет вообще допустимых решений задачи.

Если опорный план найдется, то на втором этапе решаем задачу симплекс-методом. Проиллюстрируем описанный выше метод искусственного базиса, применив его к решению задачи (7).

Этап 1. Составим исходную симплекс-таблицу для задачи (8). Все вычисления будем проводить также и для целевой функции Z. Тогда после завершения первого этапа получим целевую функцию Z , выраженную через свободные переменные.

Базис	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Значение (b_i)
x_6	2	-1	-1	0	0	1	0	2
x_7	-1	2	0	-1	0	0	1	5
x_5	1	1	0	0	1	0	0	10
- Z	-3	1	0	0	0	0	0	0
- W	0	0	0	0	0	1	1	0

Для того, чтобы начать минимизацию функции W, ее надо выразить через свободные переменные. Для этого, из W - строки вычтем первую и вторую строки. Получим

Таблица 1

Базис	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Значение (b_i)
x_6	$\boxed{2}$	-1	-1	0	0	1	0	2
x_7	-1	2	0	-1	0	0	1	5
x_5	1	1	0	0	1	0	0	10
- Z	-3	1	0	0	0	0	0	0
- W	-1	-1	1	1	0	0	0	-7

Так как в W – строке есть отрицательные элементы, то выбираем в качестве разрешающего любой из первых двух столбцов, например первый. Поскольку $min\{2/2,\ 10/1\} = 1$ достигается в первой строке, то разрешающая строка – первая и разрешающий элемент $a_{11} = 2$. Пересчитывая таблицу относительно этого элемента, получим новую таблицу:

Таблица 2

Базис	x_1	x_2	x_3	x_4	x_5	x_7	Значение (b_i)
x_1	1	-½	-½	0	0	0	1
x_7	0	$\boxed{3/2}$	-½	-1	0	1	6
x_5	0	3/2	½	0	1	0	9
- Z	0	-½	-3/2	0	0	0	3
- W	0	-3/2	½	1	0	0	-6

Так как переменная x_6 вышла из базиса, то в дальнейшем ее не используем (вычеркиваем столбец x_6). Теперь разрешающим столбцом будет второй, разрешающей строкой – вторая и после пересчета получим (поскольку искусственная переменная x_7 выйдет из базиса, столбец x_7 также вычеркиваем):

Таблица 3

Базис	x_1	x_2	x_3	x_4	x_5	*Значение* (b_i)
x_1	1	0	-2/3	-1/3	0	3
x_2	0	1	1/3	-2/3	0	4
x_5	0	0	1	1	1	3
- *Z*	0	0	-5/3	-1/3	0	5

Этап 1 успешно завершен, так как искусственные переменные выведены из базиса и мы получили опорный план $X_0 = (3, 4, 0, 0, 3)$ исходной задачи (7), к которому можно применить алгоритм симплекс-метода. На втором этапе *W* - строка уже не нужна и мы ее вычеркиваем.

Этап 2.

Таблица 3

Базис	x_1	x_2	x_3	x_4	x_5	*Значение*(b_i)
x_1	1	0	-2/3	-1/3	0	3
x_2	0	1	-1/3	-2/3	0	4
x_5	0	0	$\boxed{1}$	1	1	3
- *Z*	0	0	-5/3	-1/3	0	5

Целевую функцию *Z* можно уменьшить (сейчас $Z = -5$) если ввести в базис x_3 или x_4.

Выбираем третий столбец в качестве разрешающего, т.к. $-5/3 < -1/3$. Разрешающей строкой будет третья, т.к. только в ней есть положительный элемент разрешающего столбца. Разрешающий элемент $a_{33} = 1$. Пересчитывая таблицу, получим:

Базис	x_1	x_2	x_3	x_4	x_5	*Значение*(b_i)
x_1	1	0	0	1/3	2/3	5
x_2	0	1	0	-1/3	1/3	5
x_3	0	0	1	1	1	3
- Z	0	0	0	4/3	5/3	10

Поскольку в Z – строке таблицы 4 нет отрицательных элементов, то новый план $X_1 = (5,\ 5,\ 3,\ 0,\ 0)$ является оптимальным и $Z_{min} = Z(5,5) = -10$.

Задача решена полностью.

Контрольные задания для самостоятельного решения

Задание 3. Решить задачу линейного программирования симплекс-методом (x, y ≥ 0).

Варианты

1. $Z = x - 2y \to \min,$
$$\begin{cases} 3x + y \geq 8, \\ 4x + 5y \leq 29, \\ x + 4y \geq 10, \end{cases}$$

2. $Z = -x + y \to \min,$
$$\begin{cases} 3x - 2y \geq 7, \\ x + 2y \leq 13, \\ x - 2y \leq 1, \end{cases}$$

3. $Z = 2x + y \to \max,$
$$\begin{cases} -x + 2y \geq 3, \\ -x + y \leq 1, \\ 3x - 2y \leq 3, \end{cases}$$

4. $Z = x - y \to \max,$
$$\begin{cases} -2x + 3y \leq 5, \\ x + 2y \geq 8, \\ 3x - y \leq 10, \end{cases}$$

5. $Z = 2x + y \to \max,$
$$\begin{cases} x - y \leq 0, \\ 3x - 2y \leq 3, \\ 5x - 4y \geq 1, \end{cases}$$

6. $Z = x + y \to \max,$
$$\begin{cases} x - y \geq 0, \\ -x + 2y \leq 4, \\ -3x + 4y \geq 2, \end{cases}$$

7. $Z = 5x + 2y \to \max,$

$$\begin{cases} 3x + y \geq 9, \\ 2x + y \geq 7, \\ 5x + 2y \leq 17, \end{cases}$$

8. $Z = 2x + 4y \to \max,$

$$\begin{cases} 3x + y \geq 8, \\ -x + 3y \geq 4, \\ x + 2y \leq 11, \end{cases}$$

9. $Z = 2x + y \to \max,$

$$\begin{cases} -x + 5y \leq 4, \\ x - y \leq 4, \\ x + y \geq 2, \end{cases}$$

10. $Z = x - y \to \min,$

$$\begin{cases} x + 3y \leq 7, \\ 2x - y \leq 7, \\ 5x + y \geq 7. \end{cases}$$

ДВОЙСТВЕННОСТЬ В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ. ДВОЙСТВЕННЫЙ СИМПЛЕКС-МЕТОД

Рассмотрим задачу линейного программирования в общей форме:

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \to \max \qquad (1)$$

при ограничениях :

$$\begin{array}{ll} y_1 \geq 0, & a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n, \quad \leq b_1, \\ y_2 \geq 0, & a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n, \quad \leq b_2, \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_k \geq 0, & a_{k1} x_1 + a_{k2} x_2 + \ldots + a_{kn} x_n, \quad \leq b_k, \\ y_{k+1}, & a_{k+1\,1} x_1 + a_{k+1\,2} x_2 + \ldots + a_{k+1\,n} x_n = b_{k+1}, \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m, & a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \quad = b_m, \end{array} \qquad (2)$$

$$x_j \geq 0,\ j = 1, \ldots, l; \qquad l \leq n. \qquad (3)$$

Каждому i-му ограничению из (2) соответствует переменная y_i так называемой двойственной задачи к задаче (1) – (3) (показана слева от соответствующего ограничения).

Двойственная задача имеет вид:

$$W = b_1 y_1 + b_2 y_2 + \ldots + b_m x_m \to min \qquad (4)$$

при ограничениях :

$$\begin{array}{ll} x_1 \geq 0, & a_{11} y_1 + a_{21} y_2 + \ldots + a_{m1} y_m, \quad \geq c_1, \\ x_2 \geq 0, & a_{12} y_1 + a_{22} y_2 + \ldots + a_{m2} y_m, \quad \geq c_2, \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_l \geq 0, & a_{1l} y_1 + a_{2l} y_2 + \ldots + a_{ml} y_m, \quad \geq c_l, \\ x_{l+1}, & a_{1\,l+1} y_1 + a_{2\,l+1} y_2 + \ldots + a_{m\,l+1} y_m = c_{l+1}, \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n, & a_{1n} y_1 + a_{2n} y_2 + \ldots + a_{mn} y_m = c_n, \end{array} \qquad (5)$$

$$y_i \geq 0,\ i = 1, \ldots, k;\ k \leq n. \qquad (6)$$

Задачи (1) – (3) и (4) – (6) образуют пару задач, называемую в линейном программировании *двойственной парой.*

Сравнивая две задачи, видим, что двойственная задача по отношению к исходной (прямой) содержит те же самые коэффициенты a_{ij}, b_i, c_j и составляется согласно следующим правилам:

1. Целевая функция (1) исходной задачи задается на максимум, а целевая функция (4) двойственной задачи – на минимум.

2. Матрица A^T, составленная из коэффициентов при неизвестных в системе ограничений (5) двойственной задачи получается из матрицы A прямой задачи транспонированием (т.е. заменой строк столбцами):

$$A = \begin{pmatrix} a_{11}\ a_{12} \ldots a_{1n} \\ a_{21}\ a_{22} \ldots a_{2n} \\ \ldots\ldots\ldots\ldots \\ a_{m1}\ a_{m2} \ldots a_{mn} \end{pmatrix}. \qquad A^T = \begin{pmatrix} a_{11}\ a_{21} \ldots a_{m1} \\ a_{12}\ a_{22} \ldots a_{m2} \\ \ldots\ldots\ldots\ldots \\ a_{1n}\ a_{2n} \ldots a_{mn} \end{pmatrix}.$$

3. Число переменных в двойственной задаче (4)–(6) равно числу ограничений в системе (2) прямой задачи, а число ограничений в системе (5) двойственной задачи равно числу переменных в прямой задаче.

4. Коэффициентами при неизвестных в целевой функции (4) двойственной задачи являются свободные члены в системе (2) прямой задачи и наоборот.

5. Если переменная $x_j \geq 0$, то j-е ограничение в системе (5) двойственной задачи является неравенством " $\geq$ ". Если же переменная x_j может иметь любой знак, то j-е ограничение в системе (5) представляет собой уравнение.

Каждая из задач двойственной пары (1) – (3) и (4) – (6) фактически является самостоятельной задачей линейного программирования и может быть решена независимо одна от другой. Однако при определении симплексным методом оптимального плана одной из задач, тем самым находится решение и другой задачи. Существующие зависимости между решениями прямой и двойственной задач характеризуются сформулированными ниже теоремами двойственности.

Теорема 1. (*Основная теорема двойственности*). Если одна из пары двойственных задач имеет оптимальный план, то и другая имеет оптимальный план причем значения целевых функций задач при их оптимальных планах совпадают, т.е. $Z_{max} = W_{min}$.

Если же целевая функция одной из задач не ограничена (для исходной (1) – (3) – сверху, для двойственной (4) – (6) – снизу), то другая задача вообще не имеет допустимых решений.

Теорема 2. (*Вторая теорема двойственности*). Для того, чтобы два допустимых решения $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ и $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$ пары двойственных задач были их оптимальными решениями, необходимо и достаточно, чтобы они удовлетворяли системам уравнений:

$$\left(\sum_{i=1}^{m} a_{ij} y_i^* - c_j\right) \cdot x_j^* = 0 \quad j = 1,2,\dots,n,$$

$$\left(\sum_{j=1}^{n} a_{ij} x_j^* - b_i\right) \cdot y_i^* = 0 \quad i = 1,2,\dots,m. \tag{7}$$

Замечание. Соотношения (7) верны только для ограничений в виде неравенств и для неотрицательных переменных.

Двойственный симплекс-метод, как и симплекс-метод, используется при нахождении решения задачи линейного программирования, записанной в канонической форме. Вместе с тем двойственный симплекс-метод можно применять при решении задачи ЛП, свободные члены системы ограничений которой могут быть любыми числами (при решении задачи симплексным методом эти числа предполагались неотрицательными).

Опишем алгоритм двойственного симплекс-метода. Рассмотрим задачу линейного программирования в канонической форме.

$$Z = c_1 x_1 + \dots + c_n x_n \to min, \tag{8}$$

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m, \end{cases} \tag{9}$$

$$x_1 \ge 0,\ x_2 \ge 0, \dots\ x_n \ge 0. \tag{10}$$

Пусть матрица ограничений A содержит единичную подматрицу порядка m и первые m переменных $x_1,\ldots, x_m$ являются базисными. Среди чисел $b_i\left(i=\overline{1,m}\right)$ есть отрицательные. Вектор $X=\left(b_i, \ldots, b_m, 0, \ldots 0\right)$ есть решение системы (9). Однако, это решение не является планом задачи (8) – (10), поскольку среди его компонент есть отрицательные числе (т.е. не выполняются ограничения (10)). Исключим базисные переменные из целевой функции z:

$$z = d_0 + d_{m+1} \cdot x_{m+1} + \ldots + d_n x_n .$$

Предположим, что $d_{m+1} \geq 0,\ldots,d_n \geq 0$.

Ш а г 1. Составить исходную симплексную таблицу.

№ строки	Базис	x_1	…	x_m	x_{m+1}	…	x_k	…	x_n	b
1	x_1	1	…	0	a_{1m+1}	…	a_k	…	a_{1n}	b_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
l	x_l	0	…	0	a_{lm+1}	…	a_{lk}^{*}	…	a_{ln}	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m	x_m	0	…	1	a_{mm+1}	…	a_{mk}	…	a_{mn}	b_m
$m+1$	$-z$	0	…	0	d_{m+1}	…	d_k	…	d_n	d_o

Ш а г 2. Выяснить, имеется ли хотя бы одно отрицательное число среди элементов столбца b. Если нет, то перейти к шагу 8. Иначе – к шагу 3.

Ш а г 3. Если отрицательных чисел в столбце b несколько, то выбрать наименьшее. Пусть, для определенности, это число $b_l \left(b_l = \min\limits_{i:\, b_i<0} b_i \right)$. Строка с номером l – ведущая.

Ш а г 4. Среди элементов a_{lj}, $j=\overline{1,n}$ ведущей строки l находят отрицательные. Если таковых нет, то исходная задача не имеет решения. В противном случае перейти к шагу 5.

Ш а г 5. Вычислить $\min\limits_{j:\, a_{lj}<0}\left\{-\dfrac{d_j}{a_{lj}}\right\} = -\dfrac{d_k}{a_{lk}}$. Столбец с номером k – ведущий, a_{lk} - ведущий элемент.

Ш а г 6. С помощью ведущего элемента a_{lk} провести одну итерацию метода Жордана-Гаусса.

Ш а г 7. Построить новую симплексную таблицу и перейти к шагу 2.

Ш а г 8. Задача линейного программирования (8) – (10) решена. По последней симплексной таблице выписать оптимальный план и минимальное значение целевой функции задачи (8) – (10).

Замечание. Если среди чисел $d_{m+1},\dots,d_n$ есть отрицательные, то следует в системе ограничений (9) преобразовать свободные члены b_j в неотрицательные, умножив на (-1) строки, содержащие отрицательные свободные члены и решать задачу (8) – (10) методом искусственного базиса.

Пример. Решить задачу линейного программирования:

$$Z = 6x_1 + 9x_2 + 3x_3 \to \min,$$

$$\begin{cases} -x_1 + 2x_2 + x_3 \geq 2, \\ 3x_1 + x_2 - x_3 \geq 1, \\ x_j \geq 0,\ j = 1,\ 2,\ 3. \end{cases} \tag{11}$$

Решение. Составим для задачи (11) двойственную:

$$W = 2y_1 + y_2 \to \max,$$

$$\begin{cases} -y_1 + 3y_2 \leq 6, \\ 2y_1 + y_2 \leq 9, \\ y_1 - y_2 \leq 3, \end{cases} \tag{12}$$

$$y_1,\ y_2 \geq 0.$$

Для решения задачи (11) двойственным симплекс-методом приведем ее к каноническому виду. Для этого умножим первое и второе ограничения на (-1) и добавим соответственно неотрицательные дополнительные переменные $x_4 \geq 0$, $x_5 \geq 0$:

$$Z = 6x_1 + 9x_2 + 3x_3 \rightarrow min,$$

$$\begin{cases} x_1 - 2x_2 - x_3 + x_4 = -2, \\ -3x_1 - x_2 + x_3 + x_5 = -1, \end{cases} \tag{13}$$
$$x_j \geq 0, j = 1, 2, ..., 5.$$

Базисными переменными здесь являются переменные x_4 и x_5 . Поскольку все коэффициенты $c_j \geq 0$, то критерий оптимальности для этого базисного решения выполнен, однако само решение $X = (0,\ 0,\ 0,\ -2,\ -1)$ содержит отрицательные переменные, то есть не является допустимым. Естественно попытаться вывести отрицательные (не являющиеся допустимыми) переменные из базиса, сохранив при этом неотрицательность коэффициентов целевой функции, так как в этом случае полученное допустимое решение будет являться и оптимальным. Такой подход является содержанием двойственного симплекс-метода. Проиллюстрируем его на примере решения задачи (13).

Составим исходную симплекс-таблицу.

Базис	x_1	x_2	x_3	x_4	x_5	*Значение* (b_i)
x_4	1	-2	-1^*	1	0	-2
x_5	-3	-1	1	0	1	-1
$-Z$	6	9	3	0	0	0

Вычисляем $\min\limits_{i:b_i<0}\{b_i\} = \min\{-2;-1\} = b_1 = -2.$ Из базиса будем выводить переменную x_4 . Следовательно, первая строка таблицы является разрешающей. Среди элементов разрешающей строки находим отрицательные $a_{12} = -2$; $a_{13} = -1$.

Определяем $\min\left\{-\dfrac{c_2}{a_{12}}; -\dfrac{c_3}{a_{13}}\right\} = \min\left\{\dfrac{9}{2}; \dfrac{3}{1}\right\} = -\dfrac{c_3}{a_{13}} = 3.$ Столбец, в котором достигается этот минимум, соответствует переменной x_3.

Этот столбец является разрешающим и разрешающим элементом является элемент a_{13} = -1. Это делается для того, чтобы элементы последней строки остались неотрицательными. Проводим одну итерацию метода Жордана-Гаусса относительно этого элемента, т.е. из базиса исключаем переменную x_4 и включаем в базис переменную x_3. Новая симплекс-таблица имеет вид:

Базис	x_1	x_2	x_3	x_4	x_5	Значение (b_i)
x_3	-1	2	1	-1	0	2
x_5	-2	-3^*	0	1	1	-3
- Z	9	3	0	3	0	-6

Элемент $b_2 = -3 < 0$. Следовательно, разрешающей является вторая строка таблицы. Как и ранее, находим

$$min\left\{-\frac{c_1}{a_{21}}; -\frac{c_2}{a_{22}}\right\} = \ = min\left\{\frac{9}{2}; \frac{3}{3}\right\} = 1.$$

Следовательно второй столбец – разрешающий, переменную x_2 включаем в базис, переменную x_5 исключаем из базиса. Пересчитывая таблицу относительно элемента a_{22} = -3, получаем новую таблицу:

Базис	x_1	x_2	x_3	x_4	x_5	*Значение* (b_i)
x_3	-7/3	0	1	-1/3	2/3	0
x_2	2/3	1	0	-1/3	-1/3	1
- Z	7	0	0	4	1	-9

Среди элементов столбца "Значение" нет отрицательных чисел. В Z – строке также нет отрицательных чисел. Следовательно, найден оптимальный план: $X^* = (0; 1; 0)$, при этом $Z^* = Z_{min} = 9$. По последней симплекс-таблице находим решение двойственной задачи (12). Для этого выясняем, какие переменные задачи (11) входили в исходный базис. В первоначальной таблице это – x_4, x_5. В последней симплекс-таблице находим элементы Z – строки, соответствующие этим базисным переменным ($c_4 = 4$, $c_5 = 1$) и прибавляем к ним соответствующие коэффициенты *исходной* целевой функции ($c_4 = c_5 = 0$). В результате получаем $Y^* = (4; 1)$, $W^* = W_{\max} = 9$.

Контрольные задания
для самостоятельного решения

Задание 4. Решить задачу линейного программирования двойственным симплекс-методом.

Варианты

1. $z = x_1 + 2x_2 \to min,$
$2x_1 + 3x_2 + x_3 = 11,$
$3x_1 + 2x_2 \geq 11,$
$2x_1 - 2x_2 \leq 7,$
$x_j \geq 0, j = \overline{1,3}.$

2. $z = 2x_1 + 5x_2 \to min,$
$3x_1 + 4x_2 \leq 25,$
$2x_1 - x_2 \geq 1,$
$x_1 + x_3 = 5,$
$x_j \geq 0, j = \overline{1,3}.$

3. $z = x_1 + x_2 \to min,$
$x_1 + 2x_2 \geq 1,$
$3x_2 - x_1 \leq 14,$
$2x_1 + 2x_2 + x_3 = 6,$
$x_j \geq 0, j = \overline{1,3}.$

4. $z = 3x_1 + x_2 \to min,$
$3x_1 + x_2 \geq 1,$
$4x_2 - x_1 \leq 15,$
$3x_1 + x_2 + x_3 = 16,$
$x_j \geq 0, j = \overline{1,3}.$

5. $z = 2x_1 + x_2 \to min,$
$3x_1 + 2x_2 \geq 1,$
$-x_1 + x_2 + x_3 = 2,$
$2x_1 + 2x_2 \leq 5,$
$x_j \geq 0, j = \overline{1,3}.$

6. $z = x_1 + 2x_2 \to min,$
$x_1 + x_2 \leq 5,$
$-x_1 + 2x_2 \leq 3,$
$-x_1 + 2x_2 - x_3 = -3,$
$x_j \geq 0, j = \overline{1,3}.$

7. $z = 2x_1 + x_2 \to min,$
$3x_1 + 2x_2 + x_3 \leq 11,$
$2x_1 + 3x_2 \geq 1,$
$2x_2 - 2x_1 \leq 7,$
$x_j \geq 0, j = \overline{1,3}.$

8. $z = 2x_1 + 3x_2 \to min,$
$2x_1 + 2x_2 \geq 1,$
$4x_2 - x_1 \leq 15,$
$x_1 + 3x_2 + x_3 = 16,$
$x_j \geq 0, j = \overline{1,3}.$

9. $z = x_1 + 5x_2 \to min,$

$$3x_1 + 5x_2 \geq 1,$$
$$2x_2 - x_1 \leq 9,$$
$$x_1 + x_2 + x_3 = 7,$$
$$x_j \geq 0, j = \overline{1,3}.$$

10. $z = 8x_1 + 6x_2 + 7x_3 \to min,$

$$x_1 - x_2 + 3x_3 \geq 1,$$
$$x_1 - 2x_2 - 3x_3 \leq 1,$$
$$x_j \geq 0, j = \overline{1,3}.$$

ТРАНСПОРТНАЯ ЗАДАЧА

Общая постановка транспортной задачи состоит в определении оптимального плана перевозок некоторого однородного груза из m пунктов отправления A_1, A_2, ... , A_m в n пунктов назначения B_1, B_2, ... , B_n. При этом в качестве критерия оптимальности берется либо минимальная стоимость перевозок всего груза, либо минимальное время его доставки.

Математическая модель транспортной задачи имеет вид:

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \to min, \tag{14}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1,\dots,n), \tag{15}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1,\dots,m), \tag{16}$$

$$x_{ij} \geq 0, \quad (i = 1,\dots,m;\ j = 1,\dots,n). \tag{17}$$

Здесь c_{ij} - тарифы перевозки единицы груза из i-го пункта отправления A_i в j-й пункт назначения B_j , b_j - потребность в грузе в j-м пункте назначения, a_i – запасы груза в i-м пункте отправления.

Наличие линейных уравнений (15) и (16) обеспечивает доставку необходимого количества груза в каждый из пунктов назначения и вывоз всего имеющегося груза из всех пунктов отправления. При этом, ввиду (17), исключаются обратные перевозки. Задача (14) – (17) является частным случаем задачи линейного программирования, однако, в силу своей специфики решается специальным методом.

Если выполняется так называемое *условие баланса* то такая транспортная задача называется *закрытой*. Если условие баланса (18) не выполняется, то задача называется *открытой*.

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \,, \tag{18}$$

Теорема 1. Для разрешимости транспортной задачи необходимо и достаточно, чтобы выполнялось условие баланса (18).

Если $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$, то вводится фиктивный $(n + 1)$ - й пункт назначения с потребностью $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ и соответствующие тарифы полагают равными нулю, т.е. $c_{i\,n+1} = 0$ $(i = 1,2,\ldots, m)$. Аналогично, при $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ вводится фиктивный, $(m+1)$ - й пункт отправления с запасами груза $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$, при этом тарифы на перевозку из этого пункта также полагают равными нулю, $c_{m+1\,j} = 0$.

Для решения задачи (14) – (17) применяется *метод потенциалов*, который по существу, является другой формой симплекс-метода.

Опишем алгоритм метода. Исходные данные транспортной задачи запишем в таблице

a_i \ b_j	b_1	b_2	...	b_n
a_1	c_{11}	c_{12}	...	c_{1n}
a_2	c_{21}	c_{22}	...	c_{2n}
⋮	...	...	...	...
a_m	c_{m1}	c_{m2}	...	c_{mn}

1. Построение начального опорного плана. Система ограничений (15)-(16) содержит $m{\cdot}n$ неизвестных и $m+ n$ уравнений, связанных отношением (18). Невырожденный опорный план задачи содержит $m + n - 1$ положительных перевозок или компонент. Таким образом, если каким-либо способом получен невырожденный опорный план задачи, то в матрице (x_{ij}) значений его компонент положительными являются только $m + n - 1$, а остальные равны нулю.

Клетки, в которых находятся отличные от нуля перевозки, называются *занятыми*, остальные – *незанятыми*. Занятые клетки соответствуют базисным неизвестным, и для невырожденного опорного плана их количество равно $m + n - 1$.

Для построения начального опорного плана применим *метод минимальной стоимости.*

Выбираем клетку с минимальной стоимостью (если их несколько, возьмем любую из них). Пусть, например, $c_{lk} = \min_{i,j} c_{ij}$. Тогда в клетку ($l$, k) записывают число $x_{lk} = \min(a_l, b_k)$. При этом, если $\min(a_l, b_k) = a_l$, то значение b_k уменьшают на a_l и «закрывают» строку с номером l, так как ресурсы l-го поставщика исчерпаны. Если $\min(a_l, b_k) = b_k$, то значение a_l уменьшают на b_k и «закрывают» столбец с номером k, что означает удовлетворение спроса k-го потребителя. Если же $a_l = b_k$, то «закрывают» строку или столбец по выбору.

Вышеописанную процедуру повторяют для оставшейся транспортной таблицы до тех пор, пока не будут закрыты все строки и столбцы.

2. *Построение системы потенциалов*. Система потенциалов строится для невырожденного опорного плана и имеет вид:

$$u_i + \upsilon_j = c_{ij},$$

где c_{ij} – стоимость перевозки единицы груза *занятой* (базисной) клетки в i – й строке и j–м столбце.

Вычисление потенциалов удобно проводить по таблице, положив равным нулю один из потенциалов и затем последовательно находя все остальные потенциалы вычитанием из стоимостей базисных клеток уже известных потенциалов. Потенциалы поставщиков u_i записывают справа, а потенциалы потребителей υ_j - внизу транспортной таблицы.

3. *Проверка опорного плана на оптимальность*. Для каждой свободной клетки вычисляют оценки

$$s_{ij} = c_{ij} - u_i - \upsilon_j.$$

Если $s_{ij} \geq 0$, то опорный план оптимален и задача решена. В противном случае план не является оптимальным, следовательно, его нужно улучшить.

4. Построение цикла и нахождение нового опорного плана. Среди отрицательных оценок выбираем наименьшую. Пусть

$$s_{lk} = \min_{(i,j)} s_{ij}.$$

В клетке (l, k) нарушено условие оптимальности. Для улучшения опорного плана необходимо в клетку (l, k) отправить некоторое количество груза, превратив ее тем самым в базисную. Эта операция эквивалентна действию по замене базиса в симплекс-методе.

Клетку (l, k) отмечают знаком «+» и затем строят цикл, расставляя поочередно знаки «-» и «+» в базисных клетках так, чтобы в строках и столбцах стояло по одному знаку «+» иди «-». Цикл строится единственным образом.

Обозначим $\lambda = min\, x_{ij}$, где x_{ij} - перевозки, стоящие в вершинах цикла, отмеченных знаком «-». Величина λ определяет количество груза, которое можно перераспределить по найденному циклу. Значение λ записывают в незанятую клетку (l, k). Двигаясь по циклу, вычитают или прибавляют λ к объемам перевозок, расположенных в клетках, помеченными знаками «-» или «+» соответственно. Перевозки в остальных базисных клетках остаются без изменения. Переходим к построению системы потенциалов.

Замечание. Если условие баланса нарушено, т.е. $\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{m} b_j$, *то вводят фиктивного поставщика* $\left(при \quad \sum_{i=1}^{m} a_i < \sum_{j=1}^{m} b_j \right)$ *или потребителя* $\left(при \quad \sum_{i=1}^{m} a_i > \sum_{j=1}^{m} b_j \right)$ *с потребностью* $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$ *или поставкой* $a_{m+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{m} b_j$ *соответственно. Стоимости* *соответствующих перевозок полагаются равными нулю. При построении начального опорного плана в этом случае фиктивные клетки заполняются в последнюю очередь.*

Пример. Компания контролирует 3 фабрики, производительность которых в неделю (в тыс. изделий) задается вектором $\vec{a}=(30,25,45)$. Компания заключила договоры с четырьмя заказчиками, еженедельные потребности которых (в тыс.изделий) задаются вектором $\vec{b}=(20,15,25,30)$. Стоимость транспортировки 1 тыс. изделий с i-й фабрики j-му заказчику задается матрицей тарифов

$$C=\begin{pmatrix} 3 & 5 & 2 & 8 \\ 6 & 7 & 5 & 3 \\ 8 & 6 & 4 & 9 \end{pmatrix}.$$

Требуется определить оптимальный план перевозок с целью минимизации суммарных затрат на транспортировку.

Так как $\sum\limits_{i=1}^{3} a_i = 100 > \sum\limits_{j=1}^{4} b_j = 90$, то введем в рассмотрение фиктивного 5-го заказчика с потребностью в $b_5 = 100 - 90 = 10$ (тыс. ед.) груза. При этом положим $c_{i5} = 0$, i = 1, 2, 3. Исходные данные запишем в виде таблицы.

Таблица 1

b_j / a_i	*20*	*15*	*25*	*30*	*10*
30	5 + 3	5	25 - 2	8	0
25	6	7	5	25 3	0
45	15 - 8	15 6	+ 4	5 9	10 0

Построим начальный опорный план методом минимального элемента. Первой заполним клетку (1, 3) т. к. тариф этой клетки $c_{13} = 2$ меньше других тарифов (фиктивный столбец заполняется в последнюю очередь). Поставка для клетки (1, 3) будет равна $x_{13} = min(30, 25) = 25$. Записываем это число в верхний левый угол клетки. Это означает, что с первой фабрики третьему заказчику планируется поставить 25 тыс. ед. груза. При этом требования 3-го заказчика будут полностью удовлетворены и мы закрываем 3-й столбец. Затем в оставшейся части таблицы (без 3-го столбца) ищем клетку с минимальным тарифом. Таких клеток две (1, 1) и (2, 4). Заполняем любую из них, например, клетку (1, 1). Остаток продукции 1-й фабрики равен 30 – 25 = 5. Поэтому записать в клетку (1, 1) можно $x_{11} = \min(5, 20) = 5$. Поскольку с первой фабрики вывезен весь груз (30 тыс. ед.), то закрываем первую строку. Далее поступаем аналогично, заполняя свободные клетки в порядке возрастания тарифов, закрывая каждый раз нужные строку или столбец. В результате начальный план имеет вид (см. табл.1):

$$X^0 = \begin{pmatrix} 5 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 \\ 15 & 15 & 0 & 5 & 10 \end{pmatrix}.$$

Проверим этот план на оптимальность. Для этого найдем потенциалы u_i и v_j поставщиков и потребителей. Для этого по *занятым* клеткам составим систему уравнений вида $u_i + v_j = c_{ij}$:

$$\begin{array}{lll} u_1 + v_1 = 3, & u_2 + v_4 = 3, & u_3 + v_1 = 8, \\ u_1 + v_3 = 2, & & u_3 + v_2 = 6, \\ & & u_3 + v_4 = 9, \\ & & u_3 + v_5 = 0. \end{array}$$

Поскольку уравнений в системе столько же, сколько занятых клеток, то есть 7, а неизвестных - 8, то система имеет бесконечное множество решений. Положим, например, $u_3 = 0$. Тогда остальные потенциалы находятся однозначно: $v_5 = 0$; $v_4 = 9$; $v_2 = 6$; $v_1 = 8$; $u_2 = -6$; $u_1 = -5$; $v_3 = 7$.

Теперь вычисляем оценки s_{ij} свободных клеток по формуле $s_{ij} = c_{ij} - (u_i + v_j)$:

$s_{12} = 5 - (6\text{-}5) = 4 > 0$, $s_{21} = 6 - (-6 + 8) = 4 > 0$, $s_{25} = 0 - (6{+}0) = 6 > 0$,
$s_{14} = 6 - (-5{+}9) = 2 > 0$, $s_{22} = 7 - (-6{+}6) = 7 > 0$, $s_{33} = 4 - (0{+}7) = -3 < 0$.
$s_{15} = 0 - (-5{+}0) = 5 > 0$, $s_{23} = 5 - (-6{+}7) = 4 > 0$,

Среди оценок есть отрицательная (s_{33} = -3 < 0), следовательно план не является оптимальным. Необходимо улучшить план, загружая клетку с отрицательной оценкой.

Для этого построим для клетки (3, 3) цикл с вершинами в загруженных клетках (см. табл. 1), расставляя поочередно в вершинах, начиная с клетки (3, 3), знаки «+» и « - ». Из поставок в клетках, помеченных знаком «минус», выбираем наименьшую: $\lambda = min(15, 25) = 15$.

Для получения нового опорного плана изменим поставки в вершинах цикла: к поставкам в клетках, помеченных знаком «+», прибавляем величину λ=15, в клетках, помеченных знаком «-», вычитаем эту величину 15. Новый опорный план поместим в табл. 2.

Таблица 2

a_i \ b_j	*20*	*15*	*25*	*30*	*10*	
30	20 3	 1 5	10 2	 1 8	 2 0	-2
25	 7 6	 7 7	 7 5	25 3	 6 0	-6
45	 3 8	15 6	15 4	5 9	10 0	0
	5	6	4	9	0	

Исследование этого плана на оптимальность аналогично предыдущему. Вычисленные значения потенциалов записаны справа и снизу таблицы, а оценки s_{ij} свободных клеток поместим в левых нижних углах этих клеток. Поскольку среди оценок нет отрицательных, то найденный план является оптимальным.

Выписываем матрицу X^* (без последнего столбца):

$$X^* = \begin{pmatrix} 20 & 0 & 10 & 0 \\ 0 & 0 & 0 & 25 \\ 0 & 15 & 15 & 5 \end{pmatrix}.$$

Минимальные суммарные затраты по оптимальному плану составляют:

$Z_{min} = 20 \cdot 3 + 10 \cdot 2 + 25 \cdot 3 + 15 \cdot 6 + 15 \cdot 4 + 15 \cdot 9 = 430$ *ден. ед*. Из таблицы 2 видно, что избыточная продукция в количестве 10 тыс. изд. остается на третьей фабрике.

Контрольные задания для самостоятельного решения

Задание 5. Определить оптимальный план перевозок с целью минимизации общих затрат на транспортировку с заданными векторами $\bar{a}$ и $\bar{b}$ и матрицей стоимостей C.

Варианты	$\bar{a}$	$\bar{b}$	C
1	*2*	*3*	*4*
1	$(1,24,2,13)$	$(3,10,10,15,4)$	$\begin{pmatrix} 10 & 4 & 3 & 6 & 10 \\ 10 & 2 & 7 & 10 & 3 \\ 7 & 10 & 10 & 2 & 3 \\ 10 & 3 & 2 & 10 & 1 \end{pmatrix}$
2	$(28,9,1,2)$	$(4,8,12,15,3)$	$,\begin{pmatrix} 8 & 10 & 8 & 10 & 1 \\ 3 & 10 & 10 & 6 & 1 \\ 10 & 9 & 10 & 8 & 3 \\ 1 & 2 & 10 & 1 & 10 \end{pmatrix}$

1	*2*	*3*	*4*
3	$(27,13,2,8)$	$(7,12,16,1,4)$	$\begin{pmatrix} 10 & 2 & 3 & 10 & 2 \\ 10 & 8 & 7 & 10 & 1 \\ 10 & 1 & 6 & 10 & 8 \\ 7 & 10 & 10 & 5 & 1 \end{pmatrix}$
4	$(20,21,7,2)$	$(2,15,10,7,6)$	$\begin{pmatrix} 10 & 6 & 10 & 7 & 3 \\ 2 & 10 & 1 & 9 & 10 \\ 1 & 3 & 5 & 10 & 10 \\ 10 & 5 & 10 & 6 & 5 \end{pmatrix}$
5	$(11,7,8,14)$	$(13,5,6,4,8)$	$\begin{pmatrix} 10 & 10 & 5 & 1 & 5 \\ 4 & 10 & 2 & 10 & 2 \\ 10 & 8 & 10 & 5 & 2 \\ 4 & 2 & 10 & 10 & 8 \end{pmatrix}$
6	$(10,11,7,2)$	$(2,15,10,7,6)$	$\begin{pmatrix} 10 & 6 & 10 & 7 & 3 \\ 2 & 10 & 1 & 9 & 10 \\ 1 & 3 & 5 & 10 & 10 \\ 10 & 5 & 10 & 6 & 5 \end{pmatrix}$
7	$(5,15,2,2)$	$(21,2,11,1,5)$	$\begin{pmatrix} 1 & 9 & 10 & 10 & 2 \\ 7 & 10 & 10 & 3 & 3 \\ 9 & 10 & 10 & 1 & 3 \\ 10 & 6 & 10 & 9 & 6 \end{pmatrix}$
8	$(10,1,14,5)$	$(3,8,4,15,10)$	$\begin{pmatrix} 10 & 10 & 1 & 2 & 9 \\ 10 & 6 & 10 & 5 & 3 \\ 10 & 8 & 4 & 5 & 10 \\ 9 & 10 & 2 & 6 & 10 \end{pmatrix}$

1	2	3	4
9	$(6, 25, 2, 1)$	$(3, 18, 3, 11, 6)$	$\begin{pmatrix} 2 & 4 & 10 & 10 & 3 \\ 9 & 10 & 7 & 7 & 10 \\ 9 & 8 & 10 & 8 & 10 \\ 2 & 10 & 2 & 6 & 10 \end{pmatrix}$
10	$(15, 10, 2, 6)$	$(13, 9, 7, 4, 7)$	$\begin{pmatrix} 10 & 6 & 7 & 6 & 10 \\ 6 & 10 & 9 & 10 & 2 \\ 10 & 9 & 10 & 2 & 7 \\ 8 & 10 & 10 & 2 & 4 \end{pmatrix}$

ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ В СЕТИ

Рассмотрим сеть $G(V, U)$, где V множество вершин, а U множество дуг их соединяющих, дуге $(i, j) \in U$ поставлено в соответствие неотрицательное число d_{ij} $(d_{ij} \geq 0)$, называемое пропускной способностью этой дуги (если дуга (i, j) – неориентированная, то полагаем $d_{ij} = d_{ji}$). Выделим в сети две вершины. Одну из них назовем источником и обозначим s, а другую – стоком и обозначим t.

Каждой дуге (i, j) сети поставим в соответствие неотрицательное число x_{ij}, которое назовем потоком на дуге (i, j). Потоком из источника s в сток t в сети $G\ (V, U)$ называется множество неотрицательных чисел $\{x_{ij}\}$, удовлетворяющих ограничениям:

$$\sum_i x_{is} - \sum_j x_{sj} = -v\ , \tag{19}$$

$$\sum_i x_{it} - \sum_j x_{tj} = v\,, \tag{20}$$

$$\sum_i x_{ik} - \sum_j x_{kj} = 0, \quad k \neq s, t \tag{21}$$

$$v \geq 0, \qquad 0 \leq x_{ij} \leq d_{ij}, \quad (i, j) \in U\ . \tag{22}$$

Неотрицательная величина v называется величиной потока в сети. Ограничения (19), (20) означают, что суммарная величина v потока, выходящего из источника s, равна суммарной величине v потока, входящего в сток t. Ограничения (21) выражают тот факт, что в каждую вершину (кроме источника и стока) приходит столько потока, сколько из нее уходит. Если для дуги (i, j) имеем $x_{ij} = d_{ij}$, то дуга (i, j) называется *насыщенной потоком.*

Итак, задача нахождения максимального потока является задачей линейного программирования с целевой функцией

$$v = \sum_k x_{sk} - \sum_i x_{is}$$

и ограничениями (19) – (22). В силу своей специфики для ее решения существует более эффективный алгоритм, чем симплекс-метод.

Разобьем множество вершин v сети $G(V, U)$ на два непересекающихся подмножества R и $\overline{R}$ $(R \cap \overline{R} = \varnothing$ *и* $R \cup \overline{R} = V$. Разрезом $(R, \overline{R})$ сети $G(V,U)$ называется множество всех дуг (i, j), таких, что-либо $i \in R, j \in \overline{R}$, либо $j \in R$, $i \in \overline{R}$. Т.е. разрез - это множество всех дуг, концевые вершины которых принадлежат разным множествам: R и $\overline{R}$. При этом положим $s \in R$, $t \in \overline{R}$. Тогда мы получим разрез, отделяющий источник s от стока t. Если удалить все дуги некоторого разреза, отделяющего источник s от стока t, то на сети не останется пути ведущего из s в t.

Пропускной способностью разреза $(R, \overline{R})$ называется величина

$$C(R,\overline{R}) = \sum_{i \in R, j \in \overline{R}} d_{ij} .$$

Разрез, отделяющий источник от стока и обладающий минимальной пропускной способностью, называется *минимальным разрезом.*

Теорема (*о максимальном потоке и минимальном разрезе*). В любой сети величина максимального потока из источника s в сток t равна пропускной способности минимального разреза.

Пусть на сети имеется некоторый поток, который не является максимальным. Покажем, как найти путь, увеличивающий этот поток.

Алгоритм расстановки пометок нахождения увеличивающего пути

Шаг 1. Источник s получает метку (s^{+}).

Шаг 2. Для всех дуг (s,j) выходящих из вершины s, соответствующие вершины j получают метку s^{+}, если $x_{sj} < d_{sj}$. Для дуг (i,s), входящих в вершину s, соответствующие вершины i получают метку (s^{-}), если $x_{is} > 0$.

Шаг k. Просматриваем все вершины, помеченные на $(k$-1)— м шаге. Для каждой такой вершины k, соответствующие им вершины j, для которых $x_{kj} < d_{kj}$, получают метку (k^{+}), для всех дуг (i,k), входящих в вершину k, соответствующих им вершины i, для которых $x_{ik} > 0$, получают метку (k^{-}).

Алгоритм заканчивает работу одним из двух состояний: а) после некоторого шага мы не можем пометить ни одной вершины, и сток t

остался непомеченным. Это означает, что имеющийся поток является максимальным, а $(R, \overline{R})$, где R –множество помеченных, $\overline{R}$ – множество непомеченных вершин, образует минимальный разрез; б) сток t оказался помеченным. Двигаясь от стока t к вершине, номер которой указан в ее метке и т.д., мы придем к источнику.

Алгоритм Форда – построения максимального потока в сети

Начальный. Выбираем некоторый поток $\{x_{ij}\}$ в сети $G\ (V,U)$ (например $x_{ij}= 0$ для всех дуг $(i,j) \in U$).

Общая итерация 1. Применяем алгоритм нахождения увеличивающего пути из источника s в сток t. Если такого пути нет, то построенный поток является максимальным. Алгоритм заканчивает работу. Если увеличивающий путь найден, то переходим к 2.

2. В найденном увеличивающем пути обозначим через P^+ множество прямых, а P^- - множество обратных дуг пути и вычисляем величину $\varepsilon_1 = \min\limits_{(i,j)\in P^+} (d_{ij} - x_{ij}) > 0$ и $\varepsilon_2 = \min\limits_{(i,j)\in P^-} x_{i,j} > 0$. Полагаем $\varepsilon = \min(\varepsilon_1,\ \varepsilon_2)$.

Увеличиваем поток вдоль пути P на величину ε, полагая

$$
\begin{aligned}
x_{ij}' &= x_{ij} + \varepsilon, \text{ если } (i,j) \in P^+,\\
x_{ij}' &= x_{ij} - \varepsilon, \text{ если } (i,j) \in P^-,\\
x_{ij}' &= x_{ij}. \text{ для остальных дуг пути.}
\end{aligned}
\qquad (23)
$$

Переходим к 1.

Рассмотрим пример. На рис.1. изображена сеть. Первое число в скобках указывает пропускную способность дуги, второе – дуговой поток.

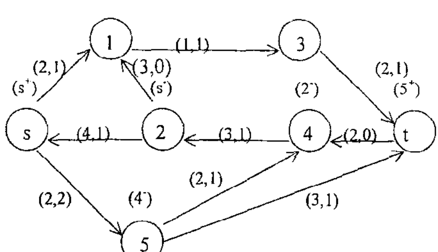


Рис. 1

Найдем увеличивающий путь.

Общая итерация: Шаг 1. Источник s получает пометку (s^+).

Шаг 2. Так как $x_{s1} = 1 < d_{s1} = 2$, то вершина 1 получает метку (s^+). Вершина 2 получает метку (s^-), т.к. $x_{21} = 1 > 0$. Вершина 5 не может быть помечена, т.к. $x_{s5} = d_{s5}$ (дуга $(s, 5)$ – насыщенная).

Шаг 3. Соседними вершинами с вновь помеченными являются вершины 3 и 4. Вершина 3 помечена не может быть, так как $x_{13} = d_{13}$..Вершина 4 получает пометку (2^-), т.к. $x_{42} = 1 > 0$.

Шаг 4. Соседними с помеченной вершиной 4 являются вершины t и 5. Вершина t помечена не может быть, т.к. $x_{t4} = 0$. Вершина 5 получает пометку (4^-), т.к. $x_{54}=1>0$.

Шаг 5. Помечаем вершину t с меткой (5^+), $x_{5t}=1 < d_{5t}=3$.

Увеличивающий путь: s – 2 – 4 – 5 - t , причем на этом пути дуга $(5, t)$ является прямой, а дуги $(2, s)$; $(4, 2)$; $(5, 4)$ обратными.

Увеличим поток вдоль этого пути по формулам (23).

Находим

$$\varepsilon_1 = \min_{(i,j)\in P^+} (d_{5t} - x_{5t}) = 2,$$

$$\varepsilon_2 = \min_{(i,j)\in P^-} (x_{S2}, x_{24}, x_{45}) = \min(1;1;1) = 1,$$

т.е. $\varepsilon = \min(\varepsilon_1, \varepsilon_2) = 1$.

Полагаем

$$x^{+1}{}_{2S} = x_{2S} - 1 = 0,\ x^{+1}{}_{42} = x_{42} - 1 = 0,\ x^{+1}{}_{54} = x_{54} - 1 = 0,\ x^{+1}{}_{5t} = x_{5t} + 1 = 2.$$

Величина суммарного потока в сети равна 3.

Общая итерация 1. Для нового потока ищем увеличивающий путь методом расстановки пометок. Пометить удается только вершины s и 1. Следовательно, увеличивающего пути нет, и построенный поток является максимальным. Минимальный разрез $(R, \overline{R})$, где $R = \{S;\ 1\}$, $\overline{R} = \{2;\ 3;\ 4;\ 5;\ t\}$ состоит из дуг $(R, \overline{R}) = \{(1, 3);\ (s, 5);\ (2,s);\ (2,1)\}$ и обладает пропускной способностью $C(R, \overline{R}) = d_{13} + d_{S5} = 3$.

На рис. 2 минимальный разрез показан пунктирной линией.

На рис. 2 минимальный разрез показан пунктирной линией

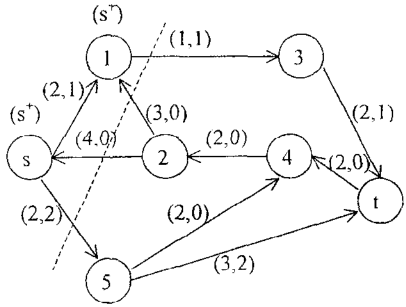


Рис. 2

Контрольные задания для самостоятельного решения

Задание 6. Сеть задана матрицей $D = \|d_{ij}\|$ пропускных способностей дуг ($d_{ij} = 0$ означает, что в сети отсутствует дуга, ведущая из вершины 1 в вершину j). Требуется по матрице D построить сеть и найти в ней максимальный поток из вершины 1 в вершину 10, определив при этом минимальный разрез.

Вариант 1

$$\begin{pmatrix} 0 & 10 & 4 & 13 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 6 & 0 & 15 & 0 & 0 & 0 \\ 4 & 5 & 0 & 7 & 2 & 5 & 0 & 0 & 0 & 0 \\ 13 & 0 & 7 & 0 & 0 & 1 & 0 & 0 & 6 & 0 \\ 0 & 6 & 0 & 0 & 0 & 10 & 3 & 7 & 0 & 0 \\ 0 & 0 & 5 & 1 & 0 & 0 & 0 & 12 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 9 \\ 0 & 0 & 0 & 0 & 7 & 12 & 5 & 0 & 9 & 7 \\ 0 & 0 & 0 & 6 & 0 & 4 & 0 & 9 & 0 & 11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 7 & 11 & 0 \end{pmatrix}.$$

Вариант 2

$$\begin{pmatrix} 0 & 2 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 7 & 0 & 15 & 0 & 5 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 8 & 3 & 0 & 0 & 0 & 0 \\ 6 & 0 & 6 & 0 & 0 & 10 & 0 & 0 & 2 & 0 \\ 0 & 15 & 8 & 0 & 0 & 4 & 1 & 5 & 0 & 0 \\ 0 & 0 & 3 & 10 & 4 & 0 & 0 & 4 & 11 & 0 \\ 0 & 5 & 0 & 0 & 1 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 4 & 9 & 0 & 0 & 6 \\ 0 & 0 & 0 & 2 & 0 & 11 & 0 & 7 & 0 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 6 & 12 & 0 \end{pmatrix}.$$

Вариант 3

$$\begin{pmatrix} 0 & 0 & 12 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 5 & 0 & 0 & 0 & 9 & 0 & 0 & 0 \\ 12 & 5 & 0 & 7 & 3 & 8 & 0 & 0 & 0 & 0 \\ 16 & 0 & 7 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 10 & 3 & 0 & 0 & 13 & 4 & 8 & 0 & 0 \\ 0 & 0 & 8 & 5 & 13 & 0 & 0 & 7 & 2 & 0 \\ 0 & 9 & 0 & 0 & 4 & 0 & 0 & 10 & 0 & 12 \\ 0 & 0 & 0 & 0 & 8 & 7 & 10 & 0 & 9 & 13 \\ 0 & 0 & 0 & 6 & 0 & 2 & 0 & 9 & 0 & 11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 13 & 11 & 0 \end{pmatrix}.$$

Вариант 4

$$\begin{pmatrix} 0 & 20 & 15 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 0 & 0 & 7 & 0 & 9 & 0 & 0 & 0 \\ 15 & 5 & 0 & 11 & 3 & 6 & 0 & 0 & 0 & 0 \\ 7 & 0 & 11 & 0 & 0 & 12 & 0 & 0 & 8 & 0 \\ 0 & 7 & 3 & 0 & 0 & 4 & 9 & 6 & 0 & 0 \\ 0 & 0 & 6 & 12 & 4 & 0 & 0 & 10 & 12 & 0 \\ 0 & 9 & 0 & 0 & 9 & 0 & 0 & 5 & 0 & 10 \\ 0 & 0 & 0 & 0 & 6 & 7 & 5 & 0 & 9 & 0 \\ 0 & 0 & 0 & 8 & 0 & 2 & 0 & 9 & 0 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 15 & 20 & 0 \end{pmatrix}.$$

Вариант 5

$$\begin{pmatrix} 0 & 20 & 25 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 7 & 0 & 15 & 0 & 0 & 0 & 0 & 0 \\ 25 & 7 & 0 & 8 & 0 & 9 & 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 & 6 & 10 & 0 & 0 & 13 & 0 \\ 0 & 15 & 0 & 6 & 0 & 9 & 0 & 7 & 0 & 0 \\ 0 & 0 & 9 & 10 & 9 & 0 & 0 & 12 & 0 & 0 \\ 0 & 5 & 0 & 0 & 8 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 12 & 10 & 0 & 3 & 15 \\ 0 & 0 & 0 & 13 & 0 & 9 & 0 & 3 & 0 & 17 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 15 & 17 & 0 \end{pmatrix}.$$

Вариант 6

$$\begin{pmatrix} 0 & 7 & 16 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 9 & 0 & 4 & 0 & 6 & 0 & 0 & 0 \\ 16 & 9 & 0 & 3 & 11 & 2 & 0 & 0 & 0 & 0 \\ 20 & 0 & 3 & 0 & 8 & 0 & 0 & 0 & 10 & 0 \\ 0 & 4 & 11 & 8 & 0 & 9 & 0 & 12 & 0 & 0 \\ 0 & 0 & 2 & 10 & 9 & 0 & 0 & 0 & 12 & 0 \\ 0 & 6 & 0 & 0 & 10 & 0 & 0 & 15 & 0 & 12 \\ 0 & 0 & 0 & 0 & 12 & 10 & 15 & 0 & 10 & 25 \\ 0 & 0 & 0 & 10 & 0 & 12 & 0 & 10 & 0 & 13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 13 & 0 \end{pmatrix}.$$

Вариант 7

$$\begin{pmatrix} 0 & 20 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 9 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 10 & 9 & 0 & 0 & 7 & 11 & 0 & 0 & 0 & 0 \\ 12 & 0 & 5 & 0 & 0 & 8 & 0 & 0 & 10 & 0 \\ 0 & 6 & 7 & 0 & 0 & 4 & 9 & 3 & 0 & 0 \\ 0 & 0 & 11 & 8 & 4 & 0 & 0 & 12 & 6 & 0 \\ 0 & 3 & 0 & 0 & 9 & 0 & 0 & 15 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 & 12 & 15 & 0 & 0 & 25 \\ 0 & 0 & 0 & 10 & 0 & 6 & 0 & 9 & 0 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 25 & 20 & 0 \end{pmatrix}.$$

Вариант 8

$$\begin{pmatrix} 0 & 0 & 19 & 13 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 10 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 19 & 10 & 0 & 3 & 10 & 0 & 0 & 0 & 0 & 0 \\ 13 & 0 & 3 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 8 & 10 & 0 & 0 & 15 & 0 & 10 & 6 & 0 \\ 0 & 0 & 6 & 9 & 15 & 0 & 0 & 12 & 2 & 0 \\ 0 & 7 & 0 & 0 & 7 & 0 & 0 & 12 & 0 & 9 \\ 0 & 0 & 0 & 0 & 10 & 12 & 12 & 0 & 10 & 20 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 10 & 0 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 20 & 15 & 0 \end{pmatrix}.$$

Вариант 9

$$\begin{pmatrix} 0 & 12 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12 & 0 & 10 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 19 & 10 & 0 & 9 & 0 & 10 & 0 & 0 & 0 & 0 \\ 6 & 0 & 9 & 0 & 0 & 12 & 0 & 0 & 10 & 0 \\ 0 & 4 & 6 & 0 & 0 & 10 & 2 & 9 & 0 & 0 \\ 0 & 0 & 10 & 12 & 10 & 0 & 0 & 12 & 10 & 0 \\ 0 & 9 & 0 & 0 & 9 & 0 & 0 & 5 & 0 & 10 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 10 & 0 & 12 & 0 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 9 & 20 & 0 \end{pmatrix}.$$

Вариант 10

$$\begin{pmatrix} 0 & 7 & 13 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 10 & 0 & 0 & 0 & 12 & 0 & 0 & 0 \\ 13 & 10 & 0 & 12 & 3 & 6 & 0 & 0 & 0 & 0 \\ 15 & 0 & 12 & 0 & 0 & 7 & 0 & 0 & 8 & 0 \\ 0 & 0 & 3 & 0 & 0 & 4 & 8 & 2 & 0 & 0 \\ 0 & 0 & 6 & 7 & 4 & 0 & 0 & 10 & 3 & 0 \\ 0 & 12 & 0 & 0 & 8 & 0 & 0 & 9 & 0 & 2 \\ 0 & 0 & 0 & 0 & 2 & 10 & 9 & 0 & 7 & 12 \\ 0 & 0 & 0 & 8 & 0 & 3 & 0 & 7 & 0 & 25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 12 & 25 & 0 \end{pmatrix}.$$

СЕТЕВОЕ ПЛАНИРОВАНИЕ

Основой методов сетевого планирования является сетевая модель (сетевой график), отражающая логическую взаимосвязь работ, входящий в некоторый комплекс, что позволяет осуществлять управление ходом выполнения этих работ.

Для построения сетевой модели необходимо, прежде всего, разбить весь комплекс на отдельные работы или операции и составить очередность выполнения этих работ. Для этого составляется список работ, которые непосредственно предшествуют каждой работе, а так же планируется время, необходимое для их выполнения.

Полученные данные удобно заносить в таблицу. В табл. 1 приведены данные для проекта, состоящего из девяти работ.

Таблица 1

№ работы	1	2	3	4	5	6	7	8	9
Предшествующие работы	-	-	1	1, 2	1, 2	3, 4	3, 4	6	7, 5
Продолжительность работы	10	15	5	20	15	6	8	10	15

На основании данных, приведенных в таблице, строится график комплекса работ, входящих в проект. Каждая работа изображается в виде ориентированного отрезка (дуги). Связи между работами изображаются пунктирными линиями (дуги-связи). В результате получается сетевой график (начальная вершина дуги – начало, а конечная – завершение соответствующей работы):

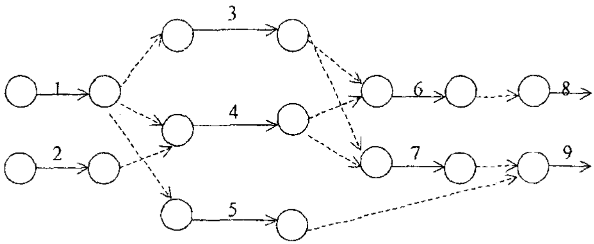


Рис. 1

Предварительно следует упростить полученную сеть. Можно удалить некоторые дуги-связи, а начало и конец удаляемой дуги объединить в одну вершину. На рис. 2 изображена сеть, полученная после упрощения сети, изображенной на рис. 1.

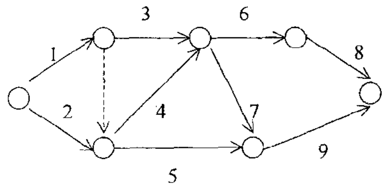


Рис. 2

В сетевом графике каждая вершина является конечной для некоторых дуг(операций), входящих в нее или начальной для дуг (операций) из нее выходящих. Поэтому каждая вершина может трактоваться как событие, означающее завершение всех операций (дуг), для которых она является конечной и возможность начала выполнения всех операций (дуг), для которых она является начальной. Начальной вершине соответствует событие, под которым подразумевается начало осуществления проекта, а конечной вершине соответствует событие – завершения выполнения всего комплекса работ.

После построения сетевого графика все его вершины нумеруются так, что нумерация является правильной.

Алгоритм правильной нумерации

Шаг 1. Нумеруем начальную вершину номером 1. Переходим к шагу 2.

Шаг 2. Удаляем из сети все выходящие из пронумерованных вершин дуги. Нумеруем в произвольном порядке вершины, в которые не входит ни одна дуга, произвольным образом возрастающими по порядку номерами. Шаг 2 проделываем до тех пор, пока не дойдем до конечной вершины, которой присваиваем следующий по порядку номер.

В результате правильной нумерации вершин сетевой график, приведенный на рис. 2 примет вид

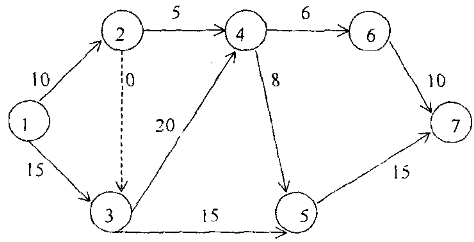


Рис. 3

Номера работ на дугах соответственно заменены продолжительностью их выполнения (продолжительность фиктивной работы соответствующей дуги-связи полагаем равной 0).

Рассмотрим основные временные параметры сетевого графика. Пусть t_{ij} –продолжительность работы, для которой соответствующая дуга *(i, j)* в сетевом графике имеет в качестве начальной – вершину с номером *i* , а в качестве конечной – вершину с номером *j*.

Ранним сроком начала работы *(i, j)* называется наименьшее допустимое время t_{ij}^{PH} , когда может быть начато ее выполнение.

Если работа начата в ранний срок, то время ее окончания t_{ij}^{PO} называется ранним сроком окончания

$$t_{ij}^{PH} = t_{ij}^{PO} - t_{ij} .$$

Ранний срок начала всех работ, для которых вершина *i* – начальная, называется ранним сроком наступления события *i* и обозначается T_i^P.

Ранний срок наступления конечного события называется критическим временем и обозначается $T_{кр}$. Таким образом, критическое время – это минимальный срок, за который может быть выполнен весь комплекс работ.

Каждый путь из начальной вершины в конечную, состоящий из дуг (работ) и дуг-связей продолжительностью $T_{кр}$, называется критическим путем, а работы, составляющие такие пути – критическими работами.

Поздними сроками начала и окончания работы *(i, j)* называется наибольшее допустимое время начала $(t_{ij}^{ПН})$ и окончания $(t_{ij}^{ПО})$ этой работы без нарушения сроков выполнения всего комплекса работ. Очевидно:

$$t_{ij}^{ПН} = t_{ij}^{ПО} - t_{ij}.$$

Наиболее поздний из поздних сроков окончания работ, входящих в вершину j, называется поздним сроком наступления события j и обозначается $T_j^П$.

Рассмотрим работу (i, j). Плановая продолжительность этой работы равна t_{ij}. Максимально допустимое время, на которое можно увеличить продолжительность работы (i, j) или задержать начало ее выполнения, при котором не изменится время выполнения всего проекта, называется полным резервом R_{ij} времени этой работы. Он равен:

$$R_{ij} = T_j^П - T_i^P - t_{ij}.$$

Резерв времени для работы (i, j), использование которого не изменит ранние сроки наступления всех событий (т.е. все работы смогут начать выполняться в минимально возможные сроки), называется свободным и может быть вычислен по формуле

$$r_{ij} = T_j^P - T_i^P - t_{ij}.$$

Очевидно, полный и свободный резерв времени любой работы, лежащей на критическом пути, равен нулю.

Алгоритм нахождения ранних сроков наступления событий

1. Полагаем $T_1^P = 0$.
2. Для $j = 2, 3, \ldots, n$ вычисляем

$$T_j^P = \max_{(k,j)\in I(j)} (T_k^P + t_{kj}).$$

Здесь $I(j)$ – множество всех дуг, входящих в вершину j.
Критическое время $T_{kp} = T_n^P$.

Алгоритм нахождения поздних сроков наступления событий

1. Полагаем $T_n^П = T$ (как правило $T = T_{kp}$.).
2. Для $i = n-1, n-2, \ldots 1$, вычисляем

$$T^{П}_i = \min_{j\in 0(i)} (T_j^{П} - t_{ij}).$$

Здесь $0\ (i)$ – множество вершин, которые являются конечным для дуг, выходящих из вершины i.

Рассмотрим сетевой график, описанный в табл. 1. События (вершины) сетевого графика изображены следующим образом:

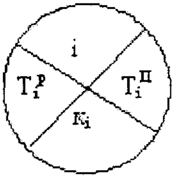


Рис. 4

В верхней четверти записан номер события (вершины) в соответствии с правильной нумерацией. Номер вершины k_i, при движении из которой получено значение T_i^P, заносится в нижнюю четверть. В левой четверти записывается ранний срок наступления события T_i^P, а в правой четверти – его поздний срок наступления $T_i^П$.

Найдем ранние сроки наступления каждого события для сетевого графика, изображенного на рис. 3.

Полагаем $T_1^P = 0,\ k_1 = 0$. Рассматриваем вершины в порядке возрастания их номеров.

$$T_2^P = T_1^P + t_{12} = 0 + 10 = 10,\ k_2 = 1,$$

$$T_3^P = \max (T_1^P + t_{13};\ T_2^P + t_{23}) =$$
$$= \max (0 + 15;\ 10 + 0) = T_1^P + t_{13} = 15, \qquad k_3 = 1,$$

$$T_4^P = \max (T_2^P + t_{24};\ T_3^P + t_{34}) =$$
$$= \max (10 + 5;\ 15 + 20) = T_3^P + t_{34} = 35, \qquad k_4 = 3,$$

$$T_5^P = \max (T_3^P + t_{35},\ T_4^P + t_{45}) =$$
$$= \max (15 + 15;\ 35 + 8) = T_4^P + t_{45} = 43, \qquad k_5 = 4,$$

$$T_6^P = T_4^P + t_{46} = 35 + 6 = 41,\ k_6 = 4,$$

$$T_{kP} = \max (T_5^P + t_{57};\ T_6^P + t_{67}) =$$
$$= \max (43 + 15;\ 41 + 10) = T_5^P + t_{57} = 58. \qquad k_7 = 5.$$

Построим критический путь, начиная с конечной вершины, двигаясь по номерам вершин k_i, стоящих в нижней четверти.

В результате получим 1 – 3 – 4 – 5 – 7. Найдем поздние сроки наступления событий. Полагаем время окончания всего проекта $T = T_7^П = T_{kp.} = 58$. Поставим это значение в правую четверть конечной вершины 7.

$$T_6^П = T_7^П - t_{67} = 58 - 10 = 48,$$

$$T_5^П = T_7^П - t_{57} = 58 - 15 = 43,$$

$$П_4^П = min\ (T_6^П - t_{46};\ T_5^П - t_{45}) = min\ (48 - 6;\ 43 - 8) = 35,$$

$$T_3^П = min\ (T_5^П - t_{35};\ T_4^П - t_{34}) = min\ (43 - 15;\ 35 - 20) = 15,$$

$$T_2^П = min\ (T_4^П - t_{24};\ T_3^П - t_{23}) = min\ (35 - 5;\ 15 - 0) = 15,$$

$$T_1^П = min\ (T_П^3 - t_{13};\ T_2^П - t_{1П}) = (15 - 15;\ 15 - 10) = 0.$$

В результате получаем следующую сетевую модель, содержащую подробную информацию о ранних, поздних сроках наступления событий, критическом времени и критическом пути. Критический путь отмечен двойными линиями.

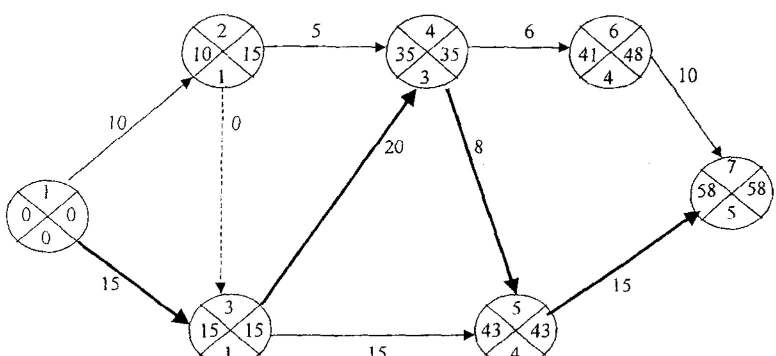


Рис. 5

Контрольные задания для самостоятельного решения

Задание 7. *В приведенных ниже таблицах комплекс работ задан их порядковыми номерами, отношением предшествования. Указаны продолжительности работ. Необходимо составить сетевой график выполнения работ и посчитать все его числовые характеристики.*

№ работ / № варианта		1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11	12
1	Каким работам предшествует	4,10	10,5	5	8	9	8	9	-	-	6
	Продолжительности работ	10	2	6	3	12	8	4	1	15	7
2	*Каким работам предшествует*	4	10,6	5,10	8	10,9	8	9	-	-	4
	Продолжительности работ	12	6	1	12	5	7	9	10	4	2
3	*Каким работам предшествует*	4	10,4,7	5	8	9	8	9	-	-	5
	Продолжительности работ	7	11	12	6	2	10	1	8	10	9
4	*Каким работам предшествует*	4,9,5	9,8	5	8	10	8	10	-	10	-
	Продолжительности работ	7	3	10	12	4	5	9	4	8	11
5	*Каким работам предшествует*	4,9	9	5,9	8	10	8	10	-	6,7	-
	Продолжительности работ	10	13	2	8	15	1	6	2	9	7
6	*Каким работам предшествует*	3,4	5	5	8	9,7	10	6	5	10	-
	Продолжительности работ	10	1	15	6	7	4	12	3	10	2
7	*Каким работам прешествует*	3,4	5	5	8	7,9	10	6	7,9	10	-
	Продолжительности работ	10	1	8	2	6	8	12	3	5	3
8	*Каким работам предшествует*	3,4	5,8	7,9	5,8	6	10	6	7,9	10	-

Окончание таблицы

		1	2	3	4	5	6	7	8	9	10
	Продолжительности работ	9	3	4	12	6	5	7	10	7	4
9	*Каким работам предшествует*	3,4	7	6,8,9	7	7	5	10	10	-	-
	Продолжительности работ	9	5	12	8	7	6	6	4	3	8
10	*Каким работам предшествует*	3	5,7	8,9	10	4,6	8,9	10	10	-	-
	Продолжительности работ	5	10	6	7	9	12	10	8	9	7

ЗАДАЧА О КРАТЧАЙШЕМ ПУТИ

Рассмотрим ориентированный граф $G(V, U)$. Каждой дуге $u_{ij} \in U$ поставлено в соответствие неотрицательное число l_{ij} , которое мы будем называть длиной дуги u_{ij} (если граф содержит неориентированную дугу, мы заменим ее парой противоположно ориентированных дуг, каждой из которых ставим в соответствие одно и то же число $l_{ij} = l_{ji}$). Рассмотрим некоторый ориентированный $(s - t)$ путь, соединяющий вершину s с вершиной t, и обозначим множество дуг входящих в него через $P(s, t)$.

Длиной пути $P(s, t)$ называется сумма $L[\, P(s, t)\,] = \sum\limits_{(i,j)\in P(s,t)} l_{ij}$, длин всех дуг, входящих в путь $P(s, t)$.

Теперь может быть сформулирована следующая задача : для выделенных вершин s и t сети $G(V, U)$, среди всех путей их соединяющих, требуется найти путь $P^*(s, t) = L[P^*(s,t)] = \min\limits_{P(s,t)} L\,[P(s, t)]$, длина которого минимальна Если вершины сети трактовать как, например, города, а пути - как дороги между ними, протяженность которых известна, то задача состоит в нахождении кратчайшего маршрута между выделенными городами.

Прежде чем описать алгоритм нахождения кратчайших путей из выделенной вершины $s \in V$ сети $G(V, U)$ во все остальные ее вершины, введем следующие обозначения. Для каждой вершины j сети $G(V, U)$, $l^*(j)$ будет обозначать длину кратчайшего $(s - j)$ пути, а $l(j)$ - длину некоторого (не обязательно кратчайшего) пути $(s - j)$ пути, а $v^*(j)$ – номер предпоследней вершины кратчайшего пути, а $v(j)$ – номер предпоследней вершины рассматриваемого пути. В процессе работы алгоритма, на каждой его итерации очередной вершине j присваивается постоянная метка вида $(l^*(j), v^*(j))$, где $v^*(j)$ - номер предпоследней вершины в кратчайшем $(s - j)$ пути. Эта вершина присоединяется к множеству вершин R, имеющих постоянную метку.

Обозначим через $arg(\min\limits_{j\in R} l(j))$ значение индекса j , при котором достигается минимальное значение величин $l(j)$, то есть

$$arg(\min_{j\in R} l(j)) = i, \text{ } если \Rightarrow l(i) = \min_{j\in R} l(j).$$

Алгоритм построения кратчайших путей в сети

Начальный шаг. Полагаем $l^*(s) := 0$, $R := \{s\}$, $l(j) = l_{sj}$, если дуга $(s, j) \in U$ и $l(j) = \infty$ в противном случае. Для всех вершин $v(j) = s$.

Общая итерация. Шаг 1. Пусть $\arg(\min\limits_{j \in R} l(j)) = i$ и вершина i имеет метку $(l(i), k)$. Если $l(k) = \infty$ и $R \neq V$ - алгоритм заканчивает работу.

Если $l(k) < \infty$ полагаем $R := R \cup \{i\}$ и $l^*(i) = l(i)$, $v^*(i) = k$.

Если $R = V$ - алгоритм заканчивает работу.

Если $R \neq V$ - переходим к шагу 2.

Шаг 2. Для всех $j \notin R$, таких, что $(i, j) \in U$ полагаем

$$l(j) := \min_{j \in R} [l^*(i) + l_{ij}, l(j)].$$

Если $l^*(i) + l_{ij} < l(j)$, то $v(j) := i$; в противном случае $v(j)$ не меняется. Переходим к шагу 1 следующей итерации.

Рассмотрим итерацию, на которой алгоритм заканчивает работу. Это происходит на шаге 1, когда либо $l(j) = \infty$ для всех $j \notin R$ и $R \neq V$, либо $R = V$.

В первом случае ни одна дуга, начальной вершиной которой являются вершина множества R, не ведет в вершины, не принадлежащие этому множеству, а значит, не существует путей, ведущих из вершины s в вершины, не принадлежащие множеству R.

Во втором случае все вершины получили постоянную метку $(l^*(i), v^*(i))$, т.е. найдены кратчайшие расстояния от вершины S ко всем остальным вершинам сети.

Заметим, что алгоритм явно не указывает кратчайший путь к вершине, а только его длину. Но воспользовавшись второй частью метки: $v^*(i)$ - его легко восстановить. Действительно, $v^*(i)$ - номер предпоследней вершины в кратчайшем пути из S в i; пусть $v^*(i) = i_1$.

Но $v^*(i_1) = i_2$ - номер предпоследней вершины в кратчайшем пути из S в i_1. Продолжая, мы найдем последовательность вершин $S, i_k, i_{k-1}, \ldots, i_2, i_1, k$, через которые проходит кратчайший путь.

Рассмотрим работу алгоритма на следующем примере.

Найти кратчайшие пути из вершины S во все остальные вершины сети, изображенной на рис.1 (числа над дугами равны их длинам).

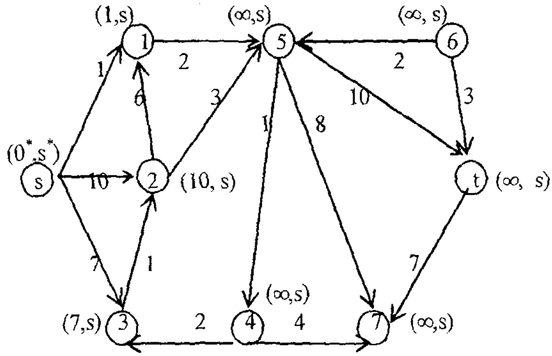


Рис. 1

На начальном плане вершина S получает постоянную метку (0^*, S^*), $R = \{S\}$, соседние с ней вершины 1, 2, 3 получают временные метки (1, S), (10, S) и (7, S) соответственно, а остальные вершины получают временные метки (∞, S) (рис. 1).

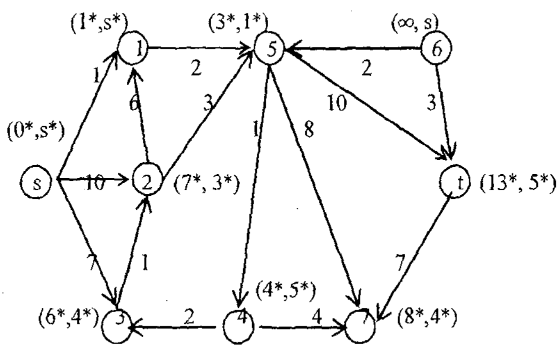


Рис. 2

Итерация 1.

1) Минимальное значение первой части меток всех вершин равно 1 для первой вершины, т.е. $arg(\min_{j \in R} l_j) = 1$. Метка первой вершины становится постоянной. Полагаем $R := R \cup \{1\} = \{S, 1\}$, переходим к шагу 2.

2) Просматриваем все вершины, соседние с вершиной, получившей постоянную метку (вершиной 1).

Для вершины 5 имеем $l^*(1) + l_{15} = 1 + 2 = 3 < l(5) = \infty$, поэтому полагаем $l(5) = 3$, $V(5) = 1$.

Для вершины 2 имеем $min(l^*(1) + l_{12}, l^*(S) + l_{32} = min(1 + \infty, 10) = 10$. Так как $l(2) = 10$, то метка вершины 2 не меняется. Переходим ко второй итерации и т.д.

Заметим, что на каждой итерации алгоритма одна очередная вершина i присоединяется к множеству R и получает постоянную метку $(l^*(i), v^*(i))$, которая в дальнейшем не меняется, а для остальных вершин $j \notin R$ пересматриваются текущие значения величин $l(j)$, некоторые из которых могут меняться и в дальнейшем. Результаты вычислений на начальном шаге (итерация 0) и на всех последующих итерациях удобно заносить в табл. 1. Если пара чисел $(l(i), v(i))$ помечается символом (*), это означает, что вершина i получила постоянную метку $(l^*(i), v^*(i))$, которая в дальнейшем не меняется.

Таблица 1

итерация / *№ вершины*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
s	*0*, s**							
1	*1, s*	*1*, s**						
2	*10, s*	*10, s*	*10, s*	*10, s*	*7, 3*	*7*, 3**		
3	*7, s*	*7, s*	*7, s*	*6, 4*	*6*, 4**			
4	*∞, s*	*∞, s*	*4, 5*	*4*, 5**				
5	*∞, s*	*3, 1*	*3*, 1**					
6	*∞, s*	*∞, s*	*∞, s*	*∞, s*	*∞, s*	*∞, s*	*∞, s*	*∞, s*
7	*∞, s*	*∞, s*	*11, 5*	*8, 4*	*8, 4*	*8, 4*	*8*, 4**	
t	*∞, s*	*∞, s*	*13, 5*	*13, 5*	*13, 5*	*13, 5*	*13, 5*	*13*, 5**

Алгоритм закончил работу на 7-й итерации случаем, когда $R = \{s, 1, 2, 3, 4, 5, 7, t\}$, $\{6\} \notin R$ и $R \neq V$, а $l(6) = \infty$. Это означает, что не существует пути, ведущего из вершины s в вершину 6. Для всех остальных вершин сети длины кратчайших путей найдены, а сами пути могут быть построены, как описано выше. Например, для вершины 2 имеем: *(l*(2), v*(2)) = (7*, 3*);* предыдущая вершина кратчайшего пути - 3. Для вершины *3 (l*(3), v*(3)) = (6*, 4*);* для вершины *4 - ((l*(4), v*(4)) = (4*, 5*);*для вершины *5 - ((l*(5), v*(5)) = (3*, 1*);* а для вершины *1 - ((l*(1), v*(1)) = (1*, s*).* Таким образом, кратчайший (s - 2) путь проходит через вершины *s, 1, 5, 4, 3, 2* и его длина равна 7.

Все дуги сети, входящие в кратчайшие пути, изображены на рис. 3. Пары чисел около вершин (рис. 2, 3) – это найденные в результате работы алгоритма постоянные метки вершин, первая часть которых *l*(i)* - длина кратчайшего *(s - i)* пути *P*(s, i),* а вторая - предпоследняя вершина этого пути (последней является вершина *i).*

Кратчайшие пути образуют дерево, но не остовное, так как вершина 6 не соединена ни с одной другой вершиной.

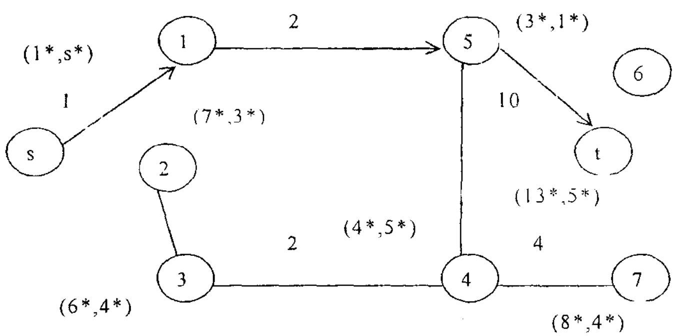


Рис. 3

В заключение отметим, что поскольку на каждой итерации алгоритма только одна *новая вершина* и соответствующая дуга добавляются к множеству дуг и вершин, образующих кратчайшие пути, то отсюда следует, что множество кратчайших путей в любой сети образует дерево (т.е. не содержит цикла).

Контрольные задания для самостоятельного решения

Задание 8. *Дана матрица* $L = \left\| l_{ij} \right\|$ *расстояний между каждой парой вершин сети. Если* $l_{ij} = \infty$, *это означает, что в сети нет дуги, ведущей из вершины i в вершину j. Если* $l_{ij} = l_{ji}$, *то вершины i и j соединены неориентированной дугой длины* l_{ij}. *Требуется по матрице L построить сеть и найти кратчайшие пути из вершины 1 во все остальные вершины сети.*

Вариант 1

$$\begin{pmatrix}
\infty & 2 & 8 & 9 & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & 7 & 6 & 10 & 6 & 5 & \infty & \infty & \infty \\
\infty & 7 & \infty & \infty & 8 & 3 & \infty & \infty & \infty & \infty \\
\infty & \infty & 6 & \infty & \infty & 10 & 9 & \infty & 2 & \infty \\
\infty & 10 & 8 & \infty & \infty & 4 & 1 & 5 & \infty & \infty \\
\infty & \infty & 3 & 10 & 4 & \infty & \infty & 4 & 11 & \infty \\
\infty & 5 & \infty & \infty & 1 & \infty & \infty & 9 & \infty & \infty \\
\infty & \infty & \infty & 8 & 5 & \infty & 9 & \infty & \infty & 6 \\
\infty & \infty & \infty & 2 & \infty & 11 & \infty & 7 & \infty & 12 \\
\infty & \infty & \infty & \infty & \infty & \infty & 4 & 6 & 12 & \infty
\end{pmatrix}.$$

Вариант 2

$$\begin{pmatrix}
\infty & 10 & 9 & 10 & \infty & \infty & \infty & \infty & \infty & \infty \\
10 & \infty & \infty & \infty & \infty & \infty & 15 & \infty & \infty & \infty \\
\infty & 5 & \infty & 8 & 2 & 6 & \infty & \infty & \infty & \infty \\
10 & \infty & \infty & \infty & \infty & 1 & \infty & \infty & 8 & \infty \\
\infty & 6 & \infty & \infty & \infty & 10 & 5 & 7 & \infty & \infty \\
\infty & \infty & 6 & 1 & \infty & \infty & \infty & 12 & 4 & \infty \\
\infty & \infty & 6 & \infty & \infty & \infty & \infty & 6 & \infty & \infty \\
\infty & \infty & \infty & 8 & 5 & \infty & 9 & \infty & \infty & 6 \\
\infty & \infty & \infty & \infty & 17 & 12 & 6 & \infty & 9 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & 8 & 7 & 11 & \infty
\end{pmatrix}.$$

Вариант 3

$$\begin{pmatrix}
\infty & 20 & \infty & 10 & \infty & \infty & \infty & \infty & \infty & \infty \\
20 & 0 & 9 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
10 & 9 & \infty & 6 & 7 & 11 & \infty & \infty & \infty & \infty \\
10 & 9 & \infty & \infty & \infty & 8 & \infty & \infty & 10 & \infty \\
\infty & 6 & 7 & \infty & \infty & 4 & 9 & 3 & \infty & \infty \\
\infty & 8 & 11 & 8 & 4 & \infty & \infty & 12 & 6 & \infty \\
\infty & 3 & \infty & 3 & \infty & \infty & \infty & 15 & \infty & 4 \\
\infty & \infty & \infty & 4 & 3 & \infty & 15 & \infty & \infty & 25 \\
\infty & \infty & \infty & 10 & \infty & \infty & \infty & 9 & \infty & 20 \\
\infty & \infty & \infty & \infty & \infty & \infty & 4 & 25 & 20 & \infty
\end{pmatrix}.$$

Вариант 4

$$\begin{pmatrix}
\infty & 12 & 10 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
12 & \infty & 10 & 6 & 4 & \infty & \infty & \infty & \infty & \infty \\
10 & 10 & \infty & 9 & \infty & \infty & \infty & \infty & \infty & \infty \\
6 & \infty & 9 & \infty & \infty & 12 & \infty & \infty & 10 & \infty \\
\infty & 4 & 6 & \infty & \infty & 10 & 2 & 9 & \infty & \infty \\
\infty & \infty & 10 & 12 & 10 & \infty & \infty & 10 & 10 & \infty \\
\infty & \infty & \infty & \infty & 2 & \infty & \infty & 9 & \infty & \infty \\
\infty & \infty & \infty & 2 & 9 & 10 & 9 & \infty & 12 & \infty \\
\infty & \infty & \infty & 10 & \infty & 10 & \infty & 12 & \infty & 20 \\
\infty & \infty & \infty & \infty & \infty & \infty & 5 & 9 & 20 & \infty
\end{pmatrix}.$$

Вариант 5

$$\begin{pmatrix}
\infty & \infty & 10 & 13 & \infty & \infty & \infty & \infty & \infty & \infty \\
7 & \infty & \infty & \infty & 8 & \infty & \infty & \infty & \infty & \infty \\
10 & 20 & \infty & 3 & 10 & \infty & \infty & \infty & \infty & \infty \\
13 & \infty & 3 & \infty & \infty & 9 & \infty & \infty & \infty & \infty \\
\infty & 8 & 10 & \infty & \infty & 10 & \infty & 10 & 6 & \infty \\
\infty & \infty & 6 & 9 & \infty & \infty & \infty & 12 & 2 & \infty \\
\infty & 7 & \infty & \infty & 7 & \infty & \infty & 12 & \infty & 9 \\
\infty & \infty & \infty & \infty & 10 & 12 & 12 & \infty & 10 & 20 \\
\infty & \infty & \infty & 8 & \infty & \infty & \infty & 10 & \infty & 10 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 20 & 10 & \infty
\end{pmatrix}.$$

Вариант 6

$$\begin{pmatrix}
\infty & 8 & 13 & 5 & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & 10 & \infty & \infty & \infty & 12 & \infty & \infty & \infty \\
13 & 10 & \infty & 10 & \infty & 6 & \infty & \infty & \infty & \infty \\
5 & 2 & \infty & \infty & \infty & 7 & \infty & \infty & 8 & \infty \\
\infty & \infty & 3 & \infty & \infty & 4 & 8 & 2 & \infty & \infty \\
\infty & \infty & 6 & 7 & \infty & \infty & \infty & 10 & 3 & \infty \\
\infty & 12 & \infty & 2 & 8 & \infty & \infty & 9 & \infty & 2 \\
\infty & \infty & \infty & \infty & 2 & \infty & 9 & \infty & 7 & 4 \\
\infty & \infty & \infty & \infty & \infty & 3 & 2 & 7 & \infty & 2 \\
\infty & \infty & \infty & \infty & \infty & \infty & 2 & 4 & 2 & \infty
\end{pmatrix}.$$

Вариант 7

$$\begin{pmatrix}
\infty & 10 & 5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
10 & \infty & 7 & \infty & 10 & \infty & \infty & \infty & \infty & \infty \\
\infty & 7 & \infty & \infty & \infty & 9 & \infty & \infty & \infty & \infty \\
8 & 2 & 8 & \infty & 6 & \infty & \infty & \infty & \infty & \infty \\
\infty & 10 & 2 & 6 & \infty & 9 & \infty & 7 & \infty & \infty \\
\infty & \infty & \infty & 6 & 9 & \infty & \infty & 12 & \infty & \infty \\
\infty & 5 & \infty & \infty & 8 & \infty & \infty & 10 & \infty & \infty \\
\infty & \infty & 2 & \infty & 7 & 12 & 10 & \infty & 3 & 15 \\
\infty & \infty & \infty & 3 & \infty & 8 & \infty & 3 & \infty & 17 \\
\infty & \infty & \infty & \infty & \infty & \infty & 3 & 15 & 17 & \infty
\end{pmatrix}.$$

Вариант 8

$$\begin{pmatrix}
\infty & 7 & \infty & 20 & \infty & \infty & \infty & \infty & \infty & \infty \\
7 & \infty & 9 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & 9 & \infty & 3 & 11 & \infty & \infty & \infty & \infty & \infty \\
20 & \infty & \infty & \infty & 8 & \infty & \infty & \infty & 10 & \infty \\
\infty & 4 & 11 & 8 & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & 2 & 10 & 9 & \infty & \infty & \infty & 12 & \infty \\
\infty & 6 & \infty & \infty & 10 & \infty & \infty & 15 & \infty & 12 \\
\infty & \infty & \infty & \infty & 12 & 10 & 15 & \infty & 10 & 25 \\
\infty & \infty & \infty & \infty & \infty & 12 & 2 & 10 & \infty & 13 \\
\infty & \infty & \infty & \infty & \infty & \infty & 12 & \infty & 13 & \infty
\end{pmatrix}.$$

Вариант 9

$$\begin{pmatrix}
\infty & \infty & 2 & 10 & \infty & \infty & \infty & \infty & \infty & \infty \\
10 & \infty & \infty & \infty & \infty & \infty & 9 & \infty & \infty & \infty \\
\infty & 5 & \infty & 7 & \infty & \infty & \infty & \infty & \infty & \infty \\
10 & 2 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & 10 & 3 & \infty & \infty & 13 & \infty & 8 & \infty & \infty \\
\infty & \infty & 8 & 5 & 13 & \infty & \infty & 7 & 2 & \infty \\
\infty & \infty & \infty & \infty & 4 & \infty & \infty & \infty & \infty & 12 \\
\infty & \infty & \infty & \infty & \infty & 7 & 10 & \infty & \infty & 13 \\
\infty & \infty & \infty & 6 & \infty & \infty & \infty & 9 & \infty & 11 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 13 & 11 & \infty
\end{pmatrix}.$$

Вариант 10

$$\begin{pmatrix}
\infty & 2 & 15 & 7 & \infty & \infty & \infty & \infty & \infty & \infty \\
2 & \infty & \infty & \infty & 7 & \infty & 9 & \infty & \infty & \infty \\
15 & 5 & \infty & 11 & \infty & \infty & \infty & \infty & \infty & \infty \\
7 & 2 & 11 & \infty & \infty & 12 & \infty & \infty & 8 & \infty \\
\infty & 7 & 3 & \infty & \infty & \infty & 9 & \infty & \infty & \infty \\
\infty & \infty & 6 & 12 & 4 & \infty & \infty & \infty & 12 & \infty \\
\infty & \infty & \infty & 2 & 9 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & 6 & 2 & 5 & \infty & 9 & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & 2 & \infty & \infty & 2 \\
\infty & \infty & \infty & \infty & \infty & \infty & 1 & 15 & 1 & \infty
\end{pmatrix}.$$

Содержание

Литература

1. Кузнецов, А.В., Холод, Н.И. Математическое программирование. – Мн: Выш. шк., 1984.

2. Балашевич, В.А. Математические методы в управление производством. – Мн.:Выш. шк., 1976.

3. Банди, Б. Основы линейного программирования. – М.: Радио и связь, 1988.

4. Банди, Б. Методы оптимизации. Вводный курс. – М.: Радио и связь, 1989.

5. Калихман, И.Л. Сборник задач по математическому программированию. – М.:Высш. шк., 1975.

6. Сборник задач и методические указания к решению задач по математическому программированию /Е.В. Емеличева [и др.]. – Мн.: БГПА, 1996.

7. Математические методы в технико-экономических задачах / Н.Е. Гайков [и др.]. – Мн.: БПИ, 1991.

Учебное издание

ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Методические указания и контрольные задания
для студентов экономических специальностей БНТУ

Составители:
КОРЗНИКОВ Александр Дмитриевич
МАТВЕЕВА Людмила Дмитриевна
СМИРНОВ Михаил Борисович

Технический редактор М.И. Гриневич
Компьютерная верстка О.В. Дубовик

Подписано в печать 29.05.2006.
Формат 60×84 1/16. Бумага офсетная.
Отпечатано на ризографе. Гарнитура Таймс.
Усл. печ. л. 3,95. Уч.-изд. л. 3,09. Тираж 300. Заказ 346.

Издатель и полиграфическое исполнение:
Белорусский национальный технический университет.
ЛИ № 02330/0131627 от 01.04.2004.
220013, Минск, проспект Независимости, 65.