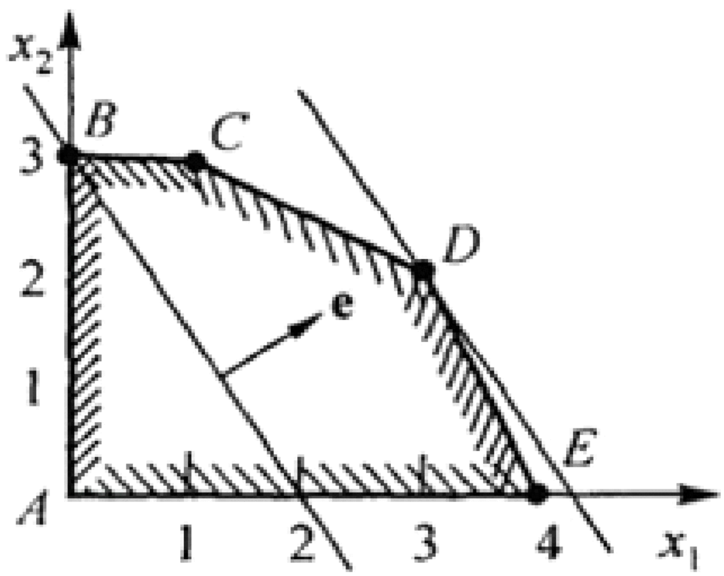
УНИВЕРСИТЕТ ИТМО

В.Н. Кудашов, Е.Г. Селина

ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ



Санкт-Петербург
2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

В.Н. Кудашов, Е.Г. Селина

ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 09.03.04 Программная инженерия
в качестве учебно-методического пособия для реализации основных профессиональных образовательных программ высшего образования бакалавриата

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2020

Кудашов В.Н., Селина Е.Г., Основы линейного программирования– СПб: Университет ИТМО, 2020. – 42 с.

Рецензент(ы):
Малышева Татьяна Алексеевна, кандидат технических наук, доцент, доцент факультета программной инженерии и компьютерной техники, Университета ИТМО.

Учебно-методическое пособие предназначено для студентов, обучающихся по направлению 09.03.04 "Программная инженерия" и включает в себя общие теоретические вопросы, связанные с методами решения задач линейного программирования и практические рекомендации по решению этих задач.

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

Введение

Линейное программирование (ЛП) — математическая дисциплина, посвящённая теории и методам решения задач об экстремумах линейных функций на множествах, задаваемых системами линейных неравенств и равенств. Линейное программирование стало развиваться в первую очередь в связи с решением задач экономики, с поиском способов оптимального распределения и использования ресурсов. Оно послужило основой широкого использования математических методов в этой сфере.

Следует подчеркнуть, что в реальных экономических задачах число независимых переменных обычно бывает очень большим (тысячи, десятки тысяч). Поэтому практическая реализация алгоритмов их решения принципиально невозможна без современной вычислительной техники.

Пособие предназначено для самостоятельного изучения студентами материала разделов 3 и 4 курса «Методы оптимизации» по теоретическим основам линейного программирования и применения этого материала для решения прикладных задач. Ориентировочные затраты времени на изучение теоретического материала по разделам 3 и 4 составляют 16 часов, на подготовку к практическим занятиям, в том числе на оформление отчетов, 20 часов.

1. *Постановка задач линейного программирования*

Общей задачей линейного программирования называется задача, которая состоит в поиске максимума (минимума) линейной функции n переменных

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j \qquad (1)$$

при линейных ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = \overline{1,k}, \qquad (2)$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = \overline{k+1,m}, \qquad (3)$$

$$x_j \ge 0, \quad j = \overline{1,r}, \; r \le n, \qquad (4)$$

где a_{ij}, b_i, c_j — заданные постоянные величины и $k \le m$. Функция (1) называется *целевой функцией*.

Стандартной задачей линейного программирования называется задача, которая состоит в поиске максимума функции (1), при выполнении условий (2) и (4), где $k = m$ и $r = n$.

Введём обозначения:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}.$$

В матричных обозначениях стандартная задача ЛП записывается следующим образом:

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle \to \max \tag{5}$$

при ограничениях

$$A\mathbf{x} \le \mathbf{b}, \tag{6}$$

$$\mathbf{x} \ge 0. \tag{7}$$

Канонической задачей линейного программирования называется задача

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle \to \max \tag{8}$$

при ограничениях

$$A\mathbf{x} = \mathbf{b}, \tag{9}$$

$$\mathbf{x} \ge 0. \tag{10}$$

Любой вектор $\mathbf{x} = (x_1, \dots, x_n)^T$, удовлетворяющий условиям (2) – (4), называется *допустимым решением*. Совокупность всех допустимых решений называется *областью допустимых решений* (ОДР).

Допустимое решение, для которого целевая функция достигает максимума (минимума), называется *оптимальным решением.*

Указанные три формы линейного программирования эквивалентны. Каждая из них с помощью несложных преобразований может быть представлена в любой другой форме.

2. *Примеры задач линейного программирования*

2.1. *Задача планирования производства*

Предприятие выпускает n видов продукции (например, столы, стулья, шкафы и т. д.). Для производства требуется m видов ресурсов (например, станки, вагоны, древесина и т. д.).

Имеется матрица затрат A, в которой a_{ij} – количество i-го ресурса, необходимого для производства единицы j-ой продукции.

Есть вектор $\mathbf{b}=(b_1,\ldots,b_m)^T$, b_i – запас i-го вида ресурса в течении некоторого количества времени. Есть вектор $\mathbf{c}=(c_1,\ldots,c_n)^T$, c_j – прибыль, полученная с единицы j-го продукта. Пусть $\mathbf{x}=(x_1,\ldots,x_n)^T$ – количество выпускаемой продукции каждого вида.

Требуется найти оптимальный план работы предприятия, т. е. определить количество выпускаемой продукции каждого вида так, чтобы прибыль была максимальной.

Математическая постановка задачи:

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

при ограничениях

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

$$x_i \ge 0, \quad i=\overline{1,n}.$$

Таким образом, мы имеем стандартную задачу линейного программирования.

2.2. *Задача о рационе* (*организации питания в большой компании*)

Имеется n продуктов питания, в которых содержится m полезных веществ. Есть матрица A, в которой a_{ij} – количество i-го полезного вещества в единице j-го продукта. Есть вектор $\mathbf{b}=(b_1,\ldots,b_m)^T$, b_i – минимальная потребность i-го полезного вещества, необходимого для поддержания нормального (здорового) состояния организма за определенный промежуток времени. Есть вектор $\mathbf{c}=(c_1,\ldots,c_n)^T$, c_j – стоимость единицы j-го продукта.

Требуется составить оптимальный рацион, который нужной питательности при минимальных затратах на него.

Математическая постановка задачи:

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

при ограничениях

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

$$x_i \ge 0, \quad i=\overline{1,n}.$$

Эта задача легко сводится к стандартной задаче ЛП.

2.3. *Транспортная задача* (*ТЗ*)

Пусть имеется некоторый однородный продукт, который надо доставить от пункта производителя в пункт потребителя. Имеется m пунктов отправления («поставщиков») и n пунктов потребления («потребителей») некоторого одинакового товара.

Есть матрица C, в которой c_{ij} – затраты на перевозку единицы продукции из пункта i в пункт j.

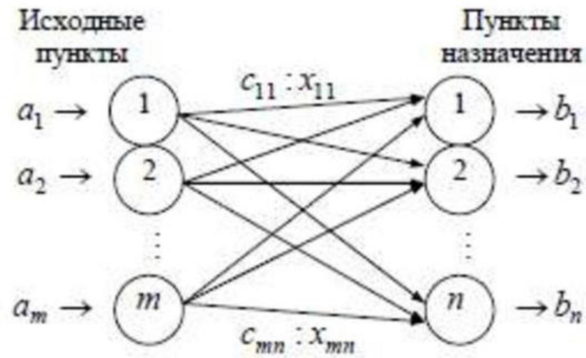


Рис. 1

a_i – количество i-го продукта у j-го производителя.

b_j – количество однородного продукта, который нужно поставить потребителю.

Требуется найти x_{ij} – количество продукта, перевозимого от i-го производителя к j-му потребителю так, чтобы затраты были минимальны.

Математическая постановка задачи:

$$f(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \rightarrow \min.$$

$$x_{ij} \geq 0,$$

$$\sum_{j=1}^{n} x_{ij} = a_i, \;\; i = \overline{1, m},$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1, n}.$$

3. *Графический метод решения задачи линейного программирования*

Если задача линейного программирования содержит только две переменные, и в её условиях нет ограничений – равенств, то такую задачу можно исследовать и решить графически.

Рассматривается задача линейного программирования:

$$f\left(x_1, x_2\right) = c_1x_1 + c_2x_2 \rightarrow \max \qquad (11)$$

при ограничениях

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 \le b_1 \\
a_{21}x_1 + a_{22}x_2 \le b_2 \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 \le b_m
\end{cases}
\qquad (12)
$$

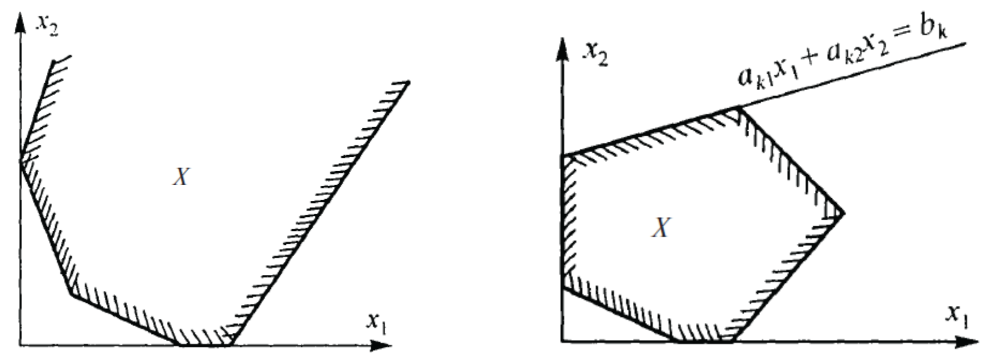
и условиях неотрицательности

$$
x_1, x_2 \ge 0 \ . \qquad (13)
$$

Таким образом, мы имеем стандартную форму задачи линейного программирования двух переменных.

На плоскости (x_1 , x_2) любое из неравенств (12) определяет полуплоскость. Область допустимых решений X задачи ЛП является пересечением первого квадранта (13) и полуплоскостей, соответствующих неравенствам (12).

Область X может быть ограниченной (рис 2,а), неограниченной (рис. 2,б) и даже пустой (тогда задача (11) – (13) не имеет решений из-за несовместимости ограничений).



а) б)

Рис. 2

Таким образом, геометрически задача линейного программирования представляет собой отыскание такой точки многоугольника решений, координаты которой доставляют линейной функции цели максимальное (минимальное) значение, причем допустимыми решениями являются все точки многоугольника решений.

Есть два способа графического решения задачи ЛП на плоскости.

***Способ* 1.** *Перебор вершин*.

Переборный метод решения основан на следующей основной теореме:

Теорема 1. *Если целевая функция имеет максимум (минимум), то он достигается в крайней точке (вершине) области допустимых решений*.

Поэтому для поиска максимума или минимума целевой функции следует:

- перебрать все вершины многоугольника;
- для каждой вершины найти значение целевой функции;

- выбрать вершину, в которой достигается оптимальное значение.

***Способ* 2.** *Градиентный метод*.

В этом случае для нахождения среди допустимых решений оптимального используют *линии уровня* и *опорные прямые*.

Линией уровня целевой функции называется прямая, на которой целевая функция задачи принимает постоянное значение. Уравнение линии уровня в общем случае имеет вид

$$c_1 x_1 + c_2 x_2 = const$$

Все линии уровня параллельны между собой.

Опорной прямой называется линия уровня, имеющая хотя бы одну общую точку с многоугольником решений системы ограничений G и по отношению к которой G находится по одну сторону. Область G имеет не более двух опорных прямых (рис. 3).

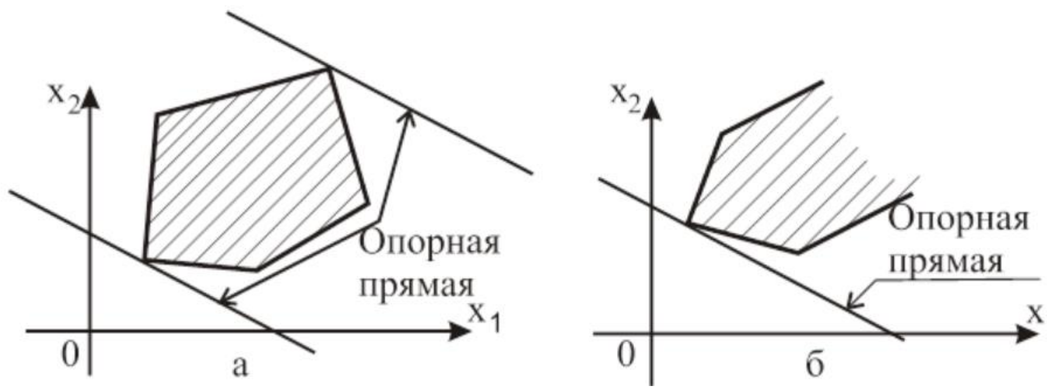


Рис. 3

Находим градиент целевой функции

$$\text{grad}\, f = \frac{\partial f}{\partial x_1}\vec{i} + \frac{\partial f}{\partial x_2}\vec{j} = c_1\vec{i} + c_2\vec{j} = \binom{c_1}{c_2}$$

Известно, что вектор градиента функции показывает направление наибольшего возрастания функции. Таким образом, значение целевой функции в точках линии уровня увеличивается, если линию уровня перемещать параллельно начальному положению в направлении вектора нормали, и убывает при перемещении в противоположном направлении.

Алгоритм метода:

1. На плоскости в системе координат $\{x_1, x_2\}$ строим прямые, уравнения которых получаются в результате замены в ограничениях знаков неравенств на знаки точных равенств.

2. Находим полуплоскости, определяемые каждым из ограничений задачи.

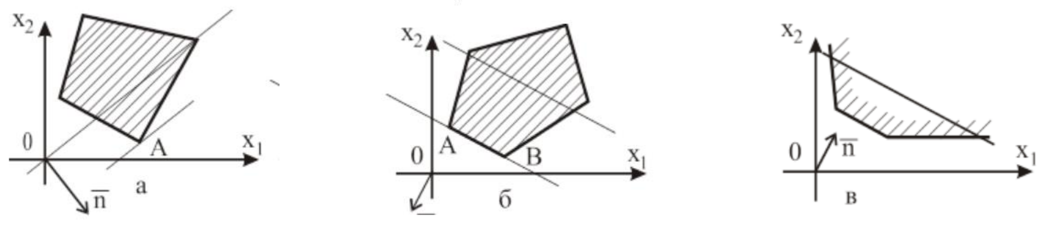
3. Находим область допустимых решений (многоугольник решений).

4. Находим градиент функции.

5. Строим прямую $c_1x_1 + c_2x_2 = const$

6. Перемещаем найденную прямую параллельно самой себе в направлении градиента функции (при поиске максимума) или антиградиента (при поиске минимума) целевой функции. В результате либо отыщется точка или множество точек, в которой целевая функция принимает максимальное (минимальное) значение, либо будет установлено, что задача не имеет решения.

На рис. 4 показаны случаи, когда задача имеет единственное решение (рис. 4, а), бесконечное множество решений (рис. 4, б), не имеет решения (рис. 4, в).



а) б) в)

Рис. 4

Пример 1.

Используя графический метод, найти решение задачи линейного программирования

$$f(x) = -3x_1 - 2x_2 \to min$$

$$\begin{cases} x_1 + 2x_2 \leq 7 \\ 2x_1 + x_2 \leq 8 \\ x_2 \leq 3 \end{cases}$$

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

Решение.

Изобразим на плоскости (x_1, x_2)допустимое множество X данной задачи (многоугольник $ABCDE$) и одну из линий уровня $-3x_1 - 2x_2 = C$ целевой функции.

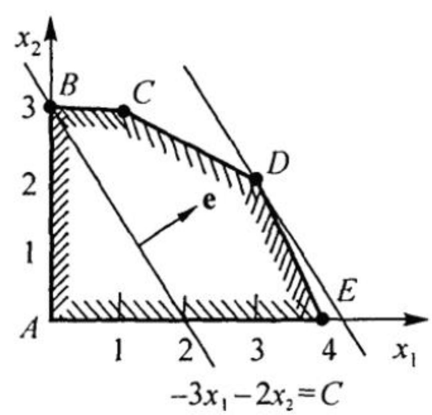


Рис. 5

Антиградиент $-\nabla f(x) = (3,2) = \vec{\mathbf{e}}$ указывает направление убывания функции $f(x)$.

Совершая параллельный перенос линии уровня вдоль напрвления $\vec{e}$, находим её крайнее положение. В этом положении прямая $-3x_1 - 2x_2 = C$ проходит через вершину $D = (3,2)$ многоугольника *ABCDE*. Поэтому целевая функция $f(x)$ принимает единственное значение f^* в точке $x^* = (3,2)$, причём

$$f^* = f(x^*) = f(3,2) = -13.$$

Пример 2.

Используя графический метод, найти решение задачи линейного программирования

$$f(x) = -x_1 - 2x_2 \to min$$
$$\begin{cases} x_1 + 2x_2 \le 7 \\ 2x_1 + x_2 \le 8 \\ \quad x_2 \le 3 \end{cases}$$
$$x_1 \ge 0, x_2 \ge 0$$

Решение.

Изобразим на плоскости (x_1, x_2) допустимое множество X данной задачи (многоугольник *ABCDE*) и одну из линий уровня $-x_1 - 2x_2 = C$ целевой функции.

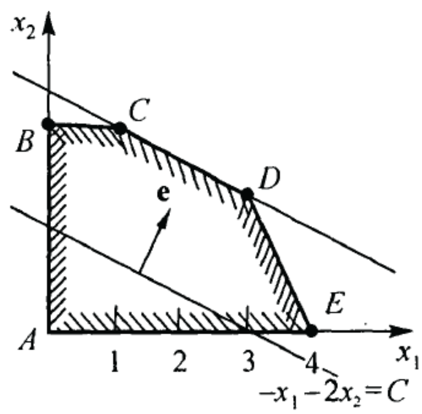


Рис. 6

Антиградиент $-\nabla f(x) = (1,2) = \vec{e}$ указывает направление убывания функции *f(x)*.

Совершая параллельный перенос линии уровня вдоль направления $\vec{e}$, находим её крайнее положение. В этом положении прямая $-x_1 - 2x_2 = C$ содержит сторону *CD* многоугольника *ABCDE*. Таким образом, все точки отрезка

$[C, D]$ являются точками минимума функции *f(x)* на множестве X. Так как концы C и D этого отрезка имеют координаты (1,3) и (3,2) соответственно, то любая точка минимума *f(x)* представима в виде

$x = \lambda(1{,}3) + (1-\lambda)(3{,}2) = (3 - 2\lambda, 2 + \lambda)$, где $\lambda \in [0,1]$.

Целевая функция *f(x)* принимает минимальное значение f^* в точках x^*, причём

$$f^* = f(x^*) = -7.$$

Пример 3.

Решить графическим методом задачу линейного программирования

$$f(x) = -x_1 - 2x_2 \to \min$$

$$\begin{cases} x_1 + x_2 \geq 1 \\ 2x_1 - x_2 \geq -1 \\ x_1 - 2x_2 \leq 0 \end{cases}$$

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

Решение.

Изобразим на плоскости (x_1, x_2) допустимое множество X данной задачи (неограниченное многоугольное множество) и одну из линий уровня $-x_1 - 2x_2 = C$ целевой функции.

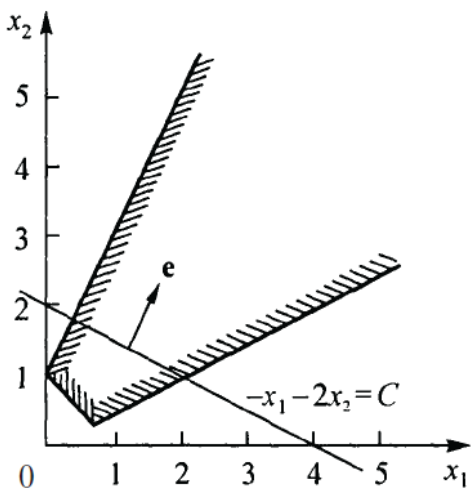


Рис. 7

Антиградиент $-\nabla f(x) = (1{,}2) = \vec{e}$ указывает направление убывания функции *f(x)*.

При параллельном переносе линии уровня $-x_1 - 2x_2 = C$ вдоль направления $\vec{e}$ она всегда пересекает множество X, а целевая функция неограниченно убывает. Поэтому данная задача линейного программирования решений не имеет.

4. *Сведение задачи ЛП к каноническому виду*

Если в исходной задаче требуется найти min, надо изменить знак и искать max этой функции.

$$\min f(\mathbf{x}) = -\max\left(-f(\mathbf{x})\right)$$

Рассмотрим линейное неравенство с n неизвестными:

$$a_{i1}x_1 + \ldots + a_{in}x_n \le b_i. \tag{14}$$

Чтобы преобразовать его в равенство, прибавим к левой части некоторую неотрицательную величину $x_{n+1} \ge 0$ так, чтобы получилось линейное уравнение:

$$a_{i1}x_1 + \ldots + a_{in}x_n + x_{n+1} = b_i. \tag{15}$$

Каждому решению неравенства (14) соответствует единственное решение уравнения (15) и наоборот.

Рассмотрим линейное неравенство с n неизвестными:

$$a_{i1}x_1 + \ldots + a_{in}x_n \ge b_i. \tag{16}$$

Чтобы преобразовать его в равенство, вычтем из левой части некоторую неотрицательную величину $x_{n+2} \ge 0$ так, чтобы получилось линейное уравнение:

$$a_{i1}x_1 + \ldots + a_{in}x_n - x_{n+2} = b_i. \tag{17}$$

Если некоторая переменная x_i не имеет ограничений по знаку, то она заменяется разностью между двумя новыми неотрицательными переменными:

$$\begin{aligned} &x_j = y_j - z_j, \\ &y_j \ge 0, \quad z_j \ge 0. \end{aligned} \tag{18}$$

Пример.

Найти $\min(x_1 - 2x_2)$ при ограничениях:

$$\begin{cases} x_1 + 3x_2 \le 5 \\ x_1 \ge 0 \\ x_2 - \forall \end{cases}$$

Приводим к каноническому виду:

$$\min(x_1 - 2x_2) = -\max(-x_1 + 2x_2)$$

$$\begin{cases} x_1 + 3x_2 + x_3 = 5 \\ x_2 = x_4 - x_5 \\ x_1 \ge 0 \\ x_4 \ge 0 \\ x_5 \ge 0 \end{cases} \qquad \begin{cases} x_1 + 3(x_4 - x_5) + x_3 = 5 \\ x_j \ge 0, \; j = 1,3,4,5 \end{cases}$$

$$\min(x_1 - 2x_2) = -\max\big(-x_1 + 2(x_4 - x_5)\big) = -\max(-x_1 + 2x_4 - 2x_5)$$

5. *Симплекс-метод решения задачи ЛП*

Будем считать, что задана задача линейного программирования в каноническом виде:

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle \to \max \tag{19}$$

при ограничениях

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbf{R}^n, \mathbf{b} \in \mathbf{R}^m, \tag{20}$$

$$\mathbf{x} \geq 0. \tag{21}$$

Идея симплекс-метода состоит в следующем. Примем в качестве начального приближения координаты некоторой вершины многогранника допустимых решений и найдем все рёбра, выходящие из этой вершины. Двигаемся вдоль того ребра, по которому линейная целевая функция возрастает. Приходим в новую вершину, находим все выходящие из нее ребра, двигаемся по одному из них и т.д. В конце концов мы придём в такую вершину, движение из которой вдоль любого ребра приведет к убыванию целевой функции. Следовательно, максимум достигнут, и координаты этой последней вершины принимаются в качестве оптимальных значений рассматриваемых проектных параметров.

Поскольку $f(\mathbf{x})$ – линейная функция, а многогранник выпуклый, данный вычислительный процесс сходится к решению задачи, причем за конечное число шагов k. В данном случае их число порядка n, т.е. значительно меньше числа шагов в методе простого перебора вершин, где k может быть порядка 2^n.

Будем считать, что у матрицы A все строки линейно независимы (если зависимые строки есть, их можно просто отбросить). У нас m – число уравнений, n – число переменных. Тогда m – ранг матрицы и $m < n$, так как при $m = n$ система (20) имеет единственное решение, что исключает оптимизацию (при $m > n$ не выполняются сделанные выше предположения).

Не уменьшая общности, будем полагать, что невырожденная матрица $m \times m$ стоит на первом месте. Если это не так, то путем перемещения переменных получим необходимое. Представим матрицу A и вектор $\mathbf{x}$ в блочном виде:

$$A = (A_{\text{б}} \mid A_{\text{с}}), \quad \mathbf{x}_{\text{б}} = \begin{pmatrix} \mathbf{x}_{\text{б}} \\ \mathbf{x}_{\text{с}} \end{pmatrix},$$

где

$$A_б = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{pmatrix}, \quad A_с = \begin{pmatrix} a_{1m+1} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{mm+1} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{x}_б = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}, \quad \mathbf{x}_с = \begin{pmatrix} x_{m+1} \\ \vdots \\ x_n \end{pmatrix}.$$

Получим

$$A\mathbf{x} = (A_б \mid A_с)\begin{pmatrix} \mathbf{x}_б \\ \mathbf{x}_с \end{pmatrix} = A_б\mathbf{x}_б + A_с\mathbf{x}_с = \mathbf{b}. \qquad (22)$$

Выразим первые m переменных $x_1,\ldots,x_m$ через остальные. Переменные $x_1,\ldots,x_m$ называются *базисными переменными*, а вектор $(x_1,\ldots,x_m)^T$ – *базисом*. Переменные $x_{m+1},\ldots,x_n$ – называются *свободными переменными*.

Свободные переменные принимают любые значения. Придавая им нулевые значения, получим частное решение системы (20):

$$\mathbf{x} = (\beta_1^{(0)},\ldots,\beta_m^{(0)},0,\ldots,0)^T.$$

Допустимое решение, соответствующее нулевым значениям свободных переменных, называется *базисным решением (базисом)*.

Базис называется *невырожденным*, если все базисные переменные больше 0, в противном случае базис называется *вырожденным*.

Суть симплекс-метода заключается в целенаправленном переборе вершин с увеличением функционала до получения оптимального решения.

Симплекс-метод включает три этапа:

1) Алгоритм отыскания начального (опорного) базиса или установления несовместимости ограничений (20) и (21).
2) Проверка текущей вершины на оптимальность полученного базиса.
3) Если критерий оптимальности не выполнен, то переходим от текущего базиса к другому с обязательным увеличением функционала до получения оптимального решения или до установления факта его неограниченности.

Допустим, мы выразили базисные переменные через свободные из (22) и получили:

$$\mathbf{x}_б = A_б^{-1}\left(\mathbf{b} - A_с\mathbf{x}_с\right). \qquad (23)$$

Выразим функцию (19) через свободные переменные:

$$f(\mathbf{x}) = <\mathbf{x},\mathbf{c}> = <\mathbf{x}_б,\mathbf{c}_б> + <\mathbf{x}_с,\mathbf{c}_с> =$$
$$= <A_б^{-1}(\mathbf{b} - A_с\mathbf{x}_c),\mathbf{c}_б> + <\mathbf{x}_c,\mathbf{c}_с> = \mathbf{c}_б^T A_б^{-1}\mathbf{b} + (\mathbf{c}_с^T - \mathbf{c}_б^T A_б^{-1}A_с)\mathbf{x}_c.$$

Обозначим $f_0 = \mathbf{c}_б^T A_б^{-1}\mathbf{b}$ и $\Delta^T = \mathbf{c}_с^T - \mathbf{c}_б^T A_б^{-1} A_с$ – вектор характеристических разностей. Тогда

$$f(\mathbf{x}) = f_0 + \langle \Delta, \mathbf{x}_с \rangle. \qquad (24)$$

Предположим, что существует хотя бы один положительный коэффициент среди Δ_i. Тогда при увеличении переменной x_i функция $f(\mathbf{x})$ увеличивается по сравнению с f_0, поэтому надо искать новое опорное решение. Следовательно, $\Delta \leq 0$ – *критерий оптимальности*. Если все $\Delta_i < 0$, то решение единственно.

Пусть критерий оптимальности не выполнен. Если только одно $\Delta_i > 0$, то увеличиваем соответствующее x_i. Если есть несколько $\Delta_i > 0$, то выбираем наибольшую $\Delta_k = \max_{\Delta_i > 0} \Delta_i$ в надежде, что увеличение соответствующего x_k приведет к увеличению функции (хотя это не обязательно так).

Введем обозначения:

$$\mathbf{x}_б^0 = A_б^{-1}\mathbf{b} = \beta_б, \quad \alpha = -A_б^{-1} A_с.$$

Тогда

$$\mathbf{x}_б = \alpha \mathbf{x}_с + \mathbf{x}_б^0. \qquad (25)$$

Запишем (24) и (25) в развёрнутом виде:

$$x_i = \sum_{j=m+1}^{n} \alpha_{ij} x_j + x_i^0, \; i = \overline{1,m}, \qquad (26)$$

$$f(\mathbf{x}) = f_0 + \sum_{j=m+1}^{n} \Delta_j x_j. \qquad (27)$$

Равенства (26) и (27) удобно оформить в виде симплекс-таблицы.

	x_{m+1}	…	x_k	…	x_n	$\mathbf{x}_б^0 = \beta$
x_1						
⋮						
x_l			α_{lk}			x_l^0
⋮						
x_m						
f	Δ_{m+1}	…	Δ_k	…	Δ_n	$f_0 = \mathbf{c}_б^0 A_б^{-1}\mathbf{b}$

Пусть $\Delta_k = \max\{\Delta_i \mid \Delta_i \geq 0\}$.

1) Если все $\alpha_{ik} \geq 0, i = \overline{1,m}$, тогда увеличение x_k не приведёт к уменьшению базисных значений. В этом случае функция $f(\mathbf{x})$ неограниченно увеличивается.

2) Пусть $\alpha_{ik} < 0$. Тогда увеличение x_k приводит к уменьшению базисных значений. Переменная $x_i = x_i^0 + \alpha_{ik} x_k$ убывает.

Переменную x_k можно увеличивать лишь до тех пор, пока базисные переменные остаются неотрицательными. Это и является условием выбора x_k':

$$x_k' = -\frac{x_i^0}{\alpha_{ik}}$$

Если таких α_{ik} несколько, надо взять наименьшее, так как первой обратится в ноль переменная x_l , для которой отношение $\dfrac{x_i^0}{\alpha_{ik}}$ минимально.

$$x_k' = \min\left(-\frac{x_i^0}{\alpha_{ik}}\right) = -\frac{x_l^0}{\alpha_{lk}} > 0$$

Переменная x_k попадает в базовые переменные, x_l, которая была базовой, стала равной нулю. Действительно,

$$x_l' = x_l^0 + \alpha_{lk}\left(-\frac{x_l^0}{\alpha_{lk}}\right) = 0.$$

Пример. Простейшая задача на симплекс-метод:

$$f(x_1, x_2) = x_1 + x_2 \to \max,$$

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

Приведём к каноническому виду.

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

Следовательно,

$$x_3 = -2x_1 - x_2 + 3,$$
$$x_4 = -x_1 - 2x_2 + 3,$$
$$f = x_1 + x_2.$$

Начальный базис равен $(0\,0\,3\,3)^T$. Составим симплекс-таблицу:

	x_1	x_2	β
x_3	–2	–1	3
x_4	–1	–2	3
f	1	1	0

В силу того, что $\Delta_1 > 0, \Delta_2 > 0,$ критерий оптимальности не выполнен.
Так как min{3/2; 3/1} = 3/2, то x_1 попадает в базовые переменные, x_3 – в свободные.

$$x_1 = -\frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}, \quad x_4 = -\frac{3}{2}x_2 + \frac{1}{2}x_3 + \frac{3}{2},$$
$$f = \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}.$$

	x_3	x_2	β
x_1	–1/2	–1/2	3/2
x_4	1/2	–3/2	3/2
f	–1/2	1/2	3/2

Так как $\Delta_2 > 0$, то критерий оптимальности не выполнен. min{3; 1} = 1, то x_2 попадает в базовые переменные, x_4 – в свободные.

$$x_1 = -\frac{2}{3}x_3 + \frac{1}{3}x_4 + 1, \quad x_2 = \frac{1}{3}x_3 - \frac{2}{3}x_4 + 1,$$
$$f = -\frac{1}{3}x_2 - \frac{1}{3}x_3 + 2.$$

	x_3	x_4	β
x_1	–2/3	1/3	1
x_2	1/3	–2/3	1
f	–1/3	–1/3	2

Критерий оптимальности выполнен. Следовательно, $f_{\max} = 2.$

6. *Получение начального базиса. Метод искусственного базиса*

Начальным шагом симплекс-метода является нахождение начального базиса. В общем случае эта задача нетривиальна.

Наиболее общим способом построения начального допустимого базисного решения задачи ЛП является использование искусственных переменных. Эти переменные в первой итерации играют роль дополнительных остаточных переменных, но на последующих итерациях от них освобождаются.

Пусть у нас есть задача ЛП в каноническом виде (19) – (21) и выбор начального базиса не очевиден.

Будем полагать, что все $b_i \geq 0$. Если есть $b_i < 0$, то домножим соответствующее уравнение на (-1).

Добавим в ограничения искусственные переменные и решим задачу ЛП минимизации суммы искусственных переменных с исходными ограничениями. Имеем

$$w(\mathbf{x}) = -\sum_{i=1}^{m} y_i \to \max \tag{28}$$

при ограничениях

$$A\mathbf{x} + \mathbf{y} = \mathbf{b}, \quad \mathbf{x} \in \mathbf{R}^n, \quad \mathbf{b}, \mathbf{y} \in \mathbf{R}^m, \tag{29}$$

$$\mathbf{x} \geq 0, \mathbf{y} \geq 0. \tag{30}$$

Если минимальное значение этой новой целевой функции больше нуля, значит, исходная задача не имеет допустимого решения, и процесс вычислений заканчивается. (Положительные значения искусственных переменных указывают на то, что исходная система ограничений несовместна.) Если новая целевая функция равна нулю, переходим ко второму этапу.

Для решения вспомогательной задачи можно воспользоваться симплекс-методом. При этом начальный базис уже найден. Действительно, если за базисные переменные принять $y_1, y_2, \dots y_m$, а за свободные $x_1, x_2, \dots, x_n$, то в системе базисные переменные уже выражены через свободные.

$$\mathbf{y} = \mathbf{b} - A\mathbf{x}, \quad \mathbf{y}_б = \mathbf{b}, \quad \mathbf{x}^0 = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}.$$

Находим $w^* = \max w.$ Возможны два варианта:

1) $w^* = 0.$ Очевидно, что это возможно, только в том сучае, когда все вспомогательные переменные равны нулю, а именно, когда

$$y_1^* = y_2^* = \ldots = y_m^* = 0$$

(* – означает оптимальное решение).

Выписав из последней симплекс-таблицы оптимальные значения переменных, получим:

$$x_j = \beta_j - (\alpha_{jm+1} x_{m+1} + \ldots + \alpha_{jn} x_n + \alpha_{j1} y_1 + \ldots + \alpha_{jm} y_m) \tag{31}$$

Причём все $\beta_j \geq 0,\ j = \overline{1, m}$. Это справедливо, так как в столбце свободных членов находятся оптимальные значения базисных переменных, а они всегда неотрицательны.

Полагая теперь в выражении (31)

$$y_1 = y_2 = \ldots = y_m = 0,$$

получаем искомый начальный базис:

$$x_j = \beta_j - (\alpha_{jm+1} x_{m+1} + \ldots + \alpha_{jn} x_n).$$

Далее необходимо выразить целевую функцию f через свободные переменные $x_{m+1}, x_{m+2}, \dots, x_n,$ и решать основную задачу максимизации функции.

2) $w^* < 0.$ Это означает, что система (29) не имеет решений, для которых

$$y_1^* = y_2^* = \ldots = y_m^* = 0.$$

Отсюда следует, что система (20), (21) несовместна.

Пример. Решим следующую задачу, используя метод искусственного базиса:

$$f(x_1, x_2, x_3) = x_1 + 3x_2 - x_3 \to \max$$

при ограничениях

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

В этой задаче выбор начального базиса неочевиден, поэтому применим метод искусственного базиса. Добавим в ограничения искусственные переменные y_1, y_2:

$$\begin{cases} x_1 + 2x_2 - x_3 + y_1 = 5, \\ 2x_1 + x_2 + 2x_3 + y_2 = 4, \\ x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0, \\ y_1 \ge 0,\ y_2 \ge 0. \end{cases}$$

и решим задачу поиска максимума функции $w = -y_1 - y_2$ при этих ограничениях.

$$y_1 = -x_1 - 2x_2 + x_3 + 5,$$
$$y_2 = -2x_1 - x_2 - 2x_3 + 4,$$
$$w = 3x_1 + 3x_2 + x_3 - 9.$$

Составим симплекс-таблицу:

	x_1	x_2	x_3	β
y_1	–1	–2	1	5
y_2	–2	–1	–2	4
w	3	3	1	–9

Так как $\Delta_1 > 0,\ \Delta_2 > 0,\ \Delta_3 > 0,$ то критерий оптимальности не выполнен. $\min\{5/1, 4/2\} = 4/2,$ следовательно, x_1 попадает в базовые, y_2 – в свободные переменные.

$$x_1 = -\frac{1}{2}y_2 - \frac{1}{2}x_2 - x_3 + 2,$$
$$y_1 = \frac{1}{2}y_2 - \frac{3}{2}x_2 + 2x_3 + 3,$$
$$w = -\frac{3}{2}y_2 + \frac{3}{2}x_2 - 2x_3 - 3.$$

	y_2	x_2	x_3	β
y_1	$1/2$	– 3/2	2	3
x_1	–1/2	–1/2	–1	2
W	–3/2	3/2	–2	–3

Критерий оптимальности не выполнен в силу $\Delta_2 < 0$.
$\min\{2, 4\} = 2$, следовательно, x_2 попадает в базовые переменные, y_1 – в свободные.

$$x_2 = \frac{1}{3}y_2 - \frac{2}{3}y_1 + \frac{4}{3}x_3 + 2,$$
$$x_1 = -\frac{2}{3}y_2 + \frac{1}{3}y_1 - \frac{5}{3}x_3 + 1,$$

	y_2	y_1	x_3	β
x_2	1/3	–2/3	4/3	2
x_1	–2/3	1/3	–5/3	1
w	–1	–1	0	0

Решаем основную задачу максимизации функции f. Полагая $y_1 = y_2 = 0$, получим начальный базис:

$$x_1 = -\frac{5}{3}x_3 + 1, \quad x_2 = \frac{4}{3}x_3 + 2,$$

$$f = x_1 + 3x_2 + x_3 = \frac{10}{3}x_3 + 7.$$

	x_3	β
x_1	–5/3	1
x_2	4/3	2
f	10/3	7

Критерий оптимальности не выполнен. Переменная x_3 попадает в базовые, x_1 – в свободные.

$$x_3 = -\frac{3}{5}x_1 + \frac{3}{5}, \quad x_2 = -\frac{4}{5}x_1 + \frac{14}{5}, \quad f = -2x_1 + 9.$$

	x_1	β
x_2	–4/5	14/5
x_3	–3/5	3/5
f	-2	9

В результате

$$\mathbf{x}^* = \left(0, \frac{14}{5}, \frac{3}{5}\right)^T, \quad f_{\max} = 9.$$

7. *Двойственная задача линейного программирования*

Каждой задаче линейного программирования можно поставить в соответствие другую задачу линейного программирования, составленную определенным образом. Связь между первой задачей и второй задачей, которую будем называть двойственной, заключается в том, что из решения одной задачи можно получить решение другой.

Пусть есть общая задача линейного программирования:

$$f(x_1, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j \tag{32}$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = \overline{1,k}, \tag{33}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = \overline{k+1,m}, \tag{34}$$

$$x_j \ge 0, \quad j = \overline{1,r}, \ r \le n. \tag{35}$$

где a_{ij} , b_i , c_j — заданные постоянные величины и $k \le m$. Задачей, двойственной к задаче (32) – (35), называется следующая задача:

$$g\left(y_1,\ldots,y_m\right) = \sum_{i=1}^{m} b_i y_i \to \min \tag{36}$$

при ограничениях

$$\sum_{i=1}^{m} a_{ij} y_i \ge c_j, \quad j = \overline{1,r}, \tag{37}$$

$$\sum_{i=1}^{m} a_{ij} y_i = c_j, \quad j = \overline{r+1,m}, \tag{38}$$

$$y_i \ge 0, \quad i = \overline{1,k}. \tag{39}$$

Переменные **x**, **y** называют двойственными переменными.

При переходе от исходной задачи к соответствующей двойственной задаче производят следующие преобразования:

1) заменяют максимизацию целевой функции минимизацией;

2) матрица *A* системы ограничений исходной задачи транспонируется;

3) число переменных y_i в двойственной задаче (36) – (39) равно числу соотношений в системе (33), (34) исходной задачи (32) – (35), а число ограничений в системе (37), (38) двойственной задачи – числу переменных в исходной задаче;

4) коэффициентами при неизвестных y_i в целевой функции (35) двойственной задачи являются свободные члены в системе (33), (34) исходной задачи, а правыми частями в соотношениях системы (37), (38) двойственной задачи – коэффициенты при неизвестных x_j в целевой функции (32) исходной задачи;

5) если переменная $x_j \geq 0$, то j-условие в системе (37), (38) является неравенством вида « $\geq$ », если же переменная x_j может принимать любые значения, то j-е соотношение представляет собой уравнение;

6) если i-е соотношение в системе (33), (34) исходной задачи является неравенством, то i-я переменная двойственной задачи $y_i \geq 0$, в противном случае y_i может принимать любые значения.

Теорема. *Если* $\mathbf{x}^0$ *– произвольное допустимое решение задачи* (32) - (35), а $\mathbf{y}^0$ *– произвольное допустимое решение задачи* (36) - (39), *тогда*

$$f(\mathbf{x}^0) \leq g(\mathbf{y}^0).$$

Если $f(\mathbf{x}^0) = g(\mathbf{y}^0)$, то $\mathbf{x}^0$ *– оптимальное решение задачи* (32) - (35), а $\mathbf{y}^0$ *– оптимальное решение задачи* (36) - (39).

Теорема двойственности. *Если одна задач* (32) - (35) *или* (36) - (39) *имеет оптимальное решение, то и другая также имеет оптимальное решение, причём*

$$\max f(\mathbf{x}) = \min g(\mathbf{y}).$$

Если целевая функция одной из задач не ограничена, то другая задача не имеет допустимых решений.

Рассмотрим стандартную задачу ЛП:

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle \to \max \tag{40}$$

при ограничениях

$$A\mathbf{x} \leq \mathbf{b}, \tag{41}$$

$$\mathbf{x} \geq 0. \tag{42}$$

В силу вышесказанного двойственной к задаче (40) – (42) является задача:

$$g(\mathbf{y}) = \langle \mathbf{b}, \mathbf{y} \rangle \to \min \tag{43}$$

при ограничениях

$$A^T\mathbf{y} \geq \mathbf{c}, \tag{44}$$

$$\mathbf{y} \geq 0. \tag{45}$$

Для канонической задачи:

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle \to \max \tag{46}$$

при ограничениях

$$A\mathbf{x} = \mathbf{b}, \tag{47}$$
$$\mathbf{x} \geq 0, \tag{48}$$

двойственной задачей является задача:

$$g(\mathbf{y}) = \langle \mathbf{b}, \mathbf{y} \rangle \to \min \tag{49}$$

при ограничениях

$$A^T \mathbf{y} \geq \mathbf{c}. \tag{50}$$

8. Методы решения транспортных задач

8.1 Постановка транспортных задач

Под названием «транспортная задача» объединяется широкий круг задач с единой математической моделью. Данные задачи относятся к задачам линейного программирования и могут быть решены симплексным методом. Однако матрица системы ограничений транспортной задачи настолько своеобразна, что для ее решения разработаны специальные методы. Эти методы, как и симплексный метод, позволяют найти начальное опорное решение, а затем, улучшая его, получить оптимальное решение.

Рассмотрим три основных вида транспортных задач.

I. Общая транспортная задача (ОТЗ).

Имеется n пунктов (n вершин сети), производящих, потребляющих или пропускающих некоторый однородный продукт в количествах d_i единиц. Будем считать, что если $d_i > 0$, то такой пункт - источник и производит d_i единиц продукта, если $d_i < 0$, то такой пункт – потребитель, если $d_i = 0$, то такой пункт – промежуточный перевалочный пункт.

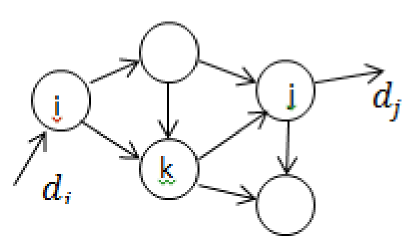


Рис.8

Дана матрица C – матрица промежуточных расходов.

$C = \{c_{ij}\}$, $c_{ij} \geq 0$

c_{ij} – затраты на перевозку однородного продукта из пункта i в пункт j
Требуется найти оптимальный грузопоток $X_{ij} \geq 0$

$$f = \sum_{i,j}^{n} c_{ij} X_{ij} \to \min. \tag{51}$$

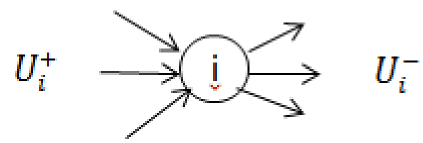


Рис.9

Обозначим U_i^- - множество индексов, соответствующее выходящим дугам, U_i^+ - множество индексов, соответствующее входящим дугам.

Условие баланса:

$$\sum_{k \in U_i^+} X_{ki} + d_i = \sum_{j \in U_i^-} X_{ij}, i = 1, 2, \ldots \tag{52}$$

$\sum_i d_i = 0$ – естественные условия баланса (весь произведенный продукт будет потреблен).

Это задача линейного программирования.

II. Классическая транспортная задача (КТЗ).

Рассмотрим транспортную сеть в более простой структуре.

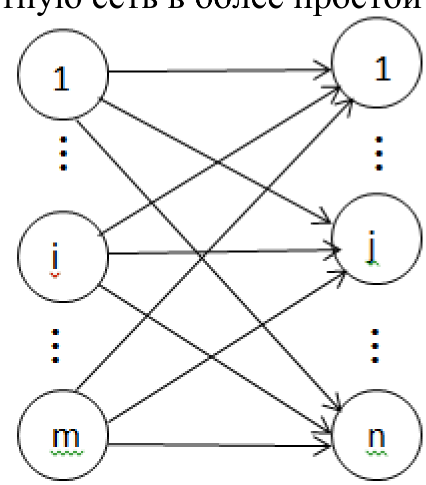


Рис.10

Есть m производителей, n потребителей. Каждый производитель соединен с каждым потребителем дорогой.

Дана матрица расходов: $C = \{C_{ij}\}$.

$C_{ij} \geq 0$ – стоимость провоза по данной дороге единицы продукта.

a_i – мощность производителя , $i = 1, m$

b_j- интенсивность потребителя, $j = 1, n$

Нет дорог внутри группы производителей и внутри группы потребителей.

Найти $X_{ij} \geq 0$ – объёмы поставок от i-го производителя к j-му потребителю так, чтобы суммарная стоимость перевозок товара от производителей к потребителям была минимальной.

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \rightarrow min$$

Рис.11

Условие баланса:

$$\sum_{j=1}^{n} X_{ij} = a_i$$

$$\sum_{i=1}^{m} X_{ij} = b_j$$

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

Это задача линейного программирования.

III. Задача определения кратчайшего пути на сети.

Задача состоит в том, чтобы найти кратчайший путь на графе от какой-то выделенной вершины до любой другой вершины.

Даны длины всех дорог C_{ij} .

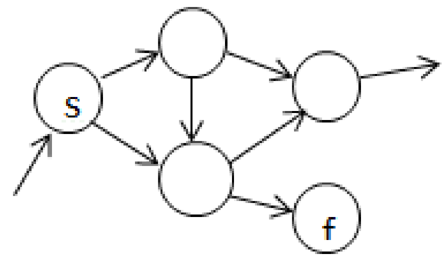


Рис.12

Найти кратчайший путь из s в f.

$$\min \sum_{(i,j)} C_{ij}$$

8.2 Алгоритм поиска кратчайшего пути

Алгоритм заключается в последовательном нахождении кратчайшего расстояния от стартовой вершины до всех вершин.

Требуется найти кратчайший путь из s в f. Даны длины всех дорог C_{ij} .

Обозначим L_i – кратчайшее расстояние до вершины i.

Вершины, до которых мы можем найти кратчайшее расстояние, будем называть помеченными. Алгоритм состоит из прямого прохода и обратного.

Этап I

Прямой проход.

1) Пусть на k-ом шаге есть множество J помеченных вершин.

2) Выделим все дуги, начинающиеся на помеченных вершинах и заканчивающиеся на непомеченных вершинах.

Обозначим множество таких дуг F^k- фронт прокладки.

3)Найдём $\min_{(i,j)\in F^k}(L_i + C_{ij}) = \quad L_{i^*} + C_{i^*j^*}$

Если таких много, то выбираем одну.

4)Помечаем j^*.

$J^{k+1} = J^k \cup j^*$

Продолжаем процесс и таким образом найдём L_f.

Итог I этапа – длина кратчайшего пути.

Этап II

Обратный проход.

Идём от вершины f до вершины s по нашим зарубкам и восстанавливаем наш путь. Если требуется несколько оптимальных маршрутов, то надо помечать не одну дугу, а несколько.

8.3 Классическая транспортная задача

Есть m производителей, n потребителей. Каждый производитель соединен с каждым потребителем дорогой.

Дана матрица расходов $C = \{C_{ij}\}$.

$C_{ij} \geq 0$ – стоимость провоза по данной дороге единицы продукта.

a_i – мощность производителя , $i = 1, m$

b_j- интенсивность потребителя, $j = 1, n$

Нет дорог внутри группы производителей и внутри группы потребителей.

Найти $X_{ij} \geq 0$ – объёмы поставок от i-го производителя к j-му потребителю так, чтобы суммарная стоимость перевозок товара от производителей к потребителям была минимальной.

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \to min$$

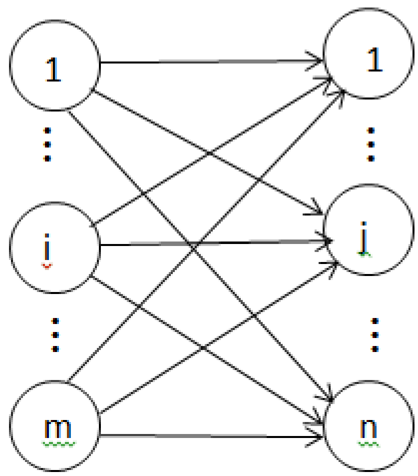


Рис.13

Суммарное количество дорог $m \cdot n$

Обычно классическая транспортная задача имеет дело с так называемой «сбалансированной» моделью, условия баланса имеют вид:

$$\sum_{j=1}^{n} X_{ij} = a_i$$

$$\sum_{i=1}^{m} X_{ij} = b_j$$

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

Первое и второе ограничения означают, что суммарное количество товара, вывозимого из пункта i , точно равно имеющимся там запасам a_i, а ввозимый груз на пункт j точно равен заявкам этого пункта.

Последнее условие называется балансным условием и означает равенство всего количества товара, имеющегося у производителей, и товара, который необходим потребителям. На практике чаще встречаются два варианта:

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

и

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

Пусть $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$. Добавим фиктивный пункт-потребитель с потребностью $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$. Теперь надо добавить дороги к этому пункту от всех производителей с $C_{i,n+1} = 0$. Тогда $X^*_{i,n+1} \geq 0$ – оптимальное количество грузов, оставшееся у i-го производителя.

Пусть $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$. Тогда добавим фиктивного (m+1)-го производителя и дороги от этому пункта ко всем производителям с $C_{m+1,j} = 0$. Тогда $X^*_{m+1,j} \geq 0$ – оптимальная недопоставка j-му потребителю.

Таким образом, любая несбалансированная задача легко приводится к сбалансированной. Поэтому здесь рассмотрим только сбалансированную задачу:

Получим задачу линейного программирования.

$$\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} X_{ij} \to \min \qquad (53)$$

при ограничениях:

$$\begin{cases} \sum_{j=1}^{n} X_{ij} = a_i, \\ \sum_{i=1}^{m} X_{ij} = b_j, \\ X_{ij} \geq 0. \end{cases} \qquad (54)$$

$$\begin{cases} \sum_{j=1}^{n} X_{ij} = a_i \\ \sum_{i=1}^{m} X_{ij} = b_j \\ X_{ij} \geq 0 \end{cases} \qquad (55)$$

Эту задачу можно решить с помощью симплекс-метода, но удобнее её решать другим методом – методом потенциалов.

8.4 Решение классической транспортной задачи методом потенциалов

В представленной задаче m× n неизвестных. Введем ряд понятий.

Матрица X размерности m× n называется планом перевозок, X_{ij} – перевозкой, $C_{ij}X_{ij}$ – матрицей издержек.

План называется допустимым, если он удовлетворяет указанным выше ограничениям.

План называется оптимальным, если он минимизирует функцию $f\,(x)$.

Метод потенциалов состоит из нескольких этапов:

I. Поиск начального базиса.
II. Проверка текущего базиса на оптимальность.
III. Если критерий оптимальности не выполнен, то улучшение базиса.

I. Нахождение первоначального базиса. Метод наименьшего элемента.

В системе ограничений $m{+}n$ уравнений, число линейно-независимых уравнений $m+n-1$ (или, иначе, ранг системы (37) равен $m+n-1$), так как одно уравнение можно исключить, используя уравнение баланса. Значит, у нас $m+n-1$ базисных переменных.

Как и в задаче линейного программирования, в транспортной задаче необходимо сначала найти первый допустимый базис. Рассмотрим нахождение первоначального базиса методом наименьшего элемента.

Сначала определим $C_{i^*j^*} = \min C_{ij}$, то есть найдём самую дешёвую дорогу. Её надо загрузить наибольшим возможным количеством товара.

Присвоим $X^0_{i^*j^*} = \min(a_{i^*}, b_{j^*})$. Пусть для определенности $a_{i^*} < b_{j^*}$, тогда $X^0_{i^*j^*} = a_{i^*}$. Следовательно, i-й производитель полностью использовал свои запасы, и при установлении остальных перевозок его можно не учитывать.

Следовательно, строка, соответствующая a_{i^*}, из таблицы вычеркивается. Теперь потребность j-го потребителя будет составлять $b_{j^*} - a_{i^*}$.

Если наоборот, $a_{i^*} > b_{j^*}$, то соответствующий столбец из таблицы вычеркивается. Далее процесс повторяется. На каждом шаге вычеркиваем строку из матрицы C, если мощность производителя больше мощности потребителя, или столбец, если наоборот. Если же они равны, то это вырожденный случай, можно вычеркнуть или строку, или столбец. На последнем шаге вычеркиваем и строку, и столбец.

В результате получим начальный базис.

$$X = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{mn} \end{pmatrix}$$

II. Проверка текущего базиса на оптимальность.

Запишем ограничения (37) в виде:

$$AX = \bar{B} \qquad \bar{B} = \binom{a}{b}$$

A – матрица, состоящая из нулей и единиц

$$A = \begin{array}{cc} \begin{array}{ccccc} A_{11} & \cdots & A_{ij} & \cdots & A_{mn} \end{array} & \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \vdots & \cdots & 0 \\ \vdots & \cdots & 1 & \cdots & \vdots \\ 0 & \cdots & \vdots & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 1 \\ 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & \vdots \\ \vdots & \cdots & \vdots & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} & \begin{array}{c} 1 \\ \vdots \\ i \\ \vdots \\ m \\ 1 \\ \vdots \\ j \\ \vdots \\ n \end{array} \end{array}$$

Симплекс-метод решает задачу поиска *max f*

$$f \downarrow = f_0 + \sum_{j \in J_c} \Delta_j X_j \uparrow$$

Критерий оптимальности $\Delta_j \leq 0$

$\Delta^T = C_c^T - C_б^T А_б^{-1} A_c$ – вектор характеристических разностей.

При доказательстве теоремы двойственности видели, что $C_б^T А_б^{-1} = \lambda^T$

$$\Delta^T = C_c^T - \lambda^T A_c$$

Применительно к нашей задаче:

$$\Delta_{ij} = C_{ij} - \lambda^T A_{ij}$$

$$f = f_0 + \sum_{(i,j)\in J_c} \Delta_{ij} X_{ij}$$

Пусть

$$\Delta_{ij} = \lambda^T A_{ij} - C_{ij}$$

(поменяем знак). Тогда

$$f \uparrow = f_0 - \sum_{(i,j)\in J_c} \Delta_{ij} X_{ij} \uparrow$$

$$\Delta_{ij} \le 0. \qquad (56)$$

Это критерий оптимальности.

Перейдём к двойственным переменным.

$$\lambda = \begin{pmatrix} u \\ v \end{pmatrix} \begin{matrix} m \\ n \end{matrix}$$

Назовём эти переменные потенциалами. u – потенциал источников, v – потенциал потребителей.

Вспомним, что Δ, соответствующие базисным переменным, равны нулю.

$\Delta_{ij} = 0, (i,j) \in J_б$, то есть для заполненных дорог

$\lambda^T A_{ij} = u_i + v_j$

Значит,

$\Delta_{ij} = \lambda^T A_{ij} - C_{ij} = u_i + v_j - C_{ij}$

Получили формулу:

$$\Delta_{ij} = u_i + v_j - C_{ij}. \qquad (57)$$

Для $(i,j) \in J_б$ $\Delta_{ij} = 0 \Rightarrow u_i + v_j = C_{ij}$

Это система для определения потенциалов, в которой $m+n-1$ уравнений, $m+n$ переменных. Уравнений на одно больше, чем неизвестных, поэтому одному неизвестному, обычно u_1, придают значение 0. После этого определяем остальные потенциалы. После нахождения λ можем найти все Δ_{ij} из (39).

III. Улучшение базиса.

Пусть критерий (38) не выполняется, то есть существует $\Delta_{ij} > 0$. Тогда

$$f \downarrow= f_0 - \sum_{(i,j)} \Delta_{ij} X_{ij} \uparrow$$

$\Delta_{i^* j^*} = \max \Delta_{ij} > 0$

Пунктиром рисуем дорогу от i^* к j^*, по которой не везли груз, а теперь надо везти. Добавим на неё грузопоток и обозначим +Ѳ. Выделим контур, содержащий эту дорогу. Нужно сохранить баланс, поэтому прибавляем и вычитаем Ѳ на контуре.

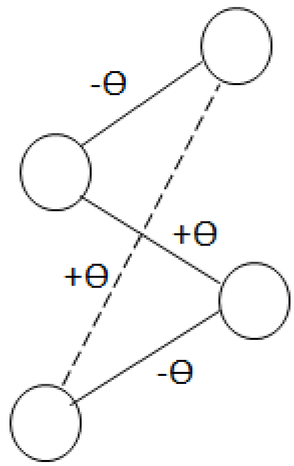


Рис. 14

Найдём $\theta^* = \min_{-\theta} X_{ij} = X_{i^* j^*}$. По дороге $i^* j^*$ пойдёт нулевой грузопоток.

$$f = f_0 - \Delta_{ij} \theta^* < f_0$$

Пример. Дана транспортная сеть, состоящая из семи вершин. Пункты 1,2,3 – производители, пункты 4,5,6,7 – потребители.

Источники: $\begin{cases} d_1 = 11 \\ d_2 = 11 \\ d_3 = 8 \end{cases}$

Получатели: $\begin{cases} d_4 = -5 \\ d_5 = -9 \\ d_6 = -9 \\ d_7 = -7 \end{cases}$

Дана матрица транспортных расходов:

$$C = \begin{pmatrix} 7 & 8 & 5 & 3 \\ 2 & 4 & 5 & 9 \\ 6 & 3 & 1 & 2 \end{pmatrix}$$

Оформим в виде таблицы:

C/X^0

	4	5	6	7	a			
1	7	8 3	5 1	3 7	11	4	3	0
2	2 5	4 6	5 -	9 -	11	6	0	
3	6 -	3 -	1 8	2 -	8	0		
b	5	9	9	7	30			
	0	3	1	0				
		0	0					

Получаем начальный базис:

	3	1	7
5	6		
		8	

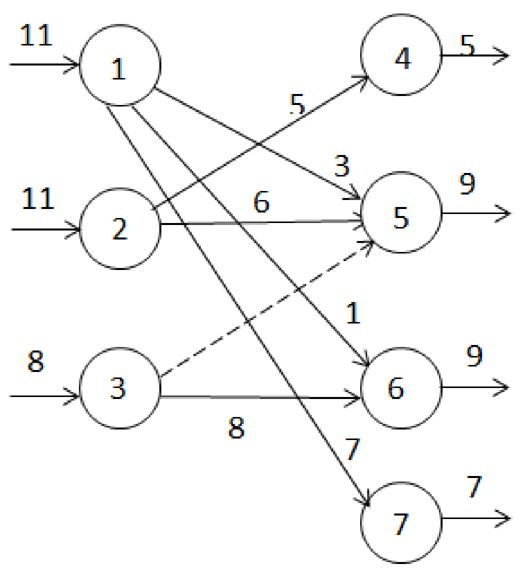


Рис.15

$$f_0 = \sum_{i=1}^{3}\sum_{j=1}^{4} C_{ij}X_{ij} = 24 + 5 + 21 + 10 + 24 + 8 = 92$$

Для $(i,j) \in J_6$ $\Delta_{ij} = 0 \Rightarrow u_i + v_j = C_{ij}$

Это система для определения потенциалов, в которой $m+n-1=6$ уравнений, $m+n=7$ переменных. Уравнений на одно больше, чем неизвестных, поэтому одному неизвестному, u_1, присвоим значение 0. После этого определяем остальные потенциалы.

$$\begin{cases} u_1 + v_2 = 8 \\ u_1 + v_3 = 5 \\ u_1 + v_4 = 3 \\ u_2 + v_1 = 2 \\ u_2 + v_2 = 4 \\ u_3 + v_3 = 1 \end{cases} \qquad \begin{cases} u_1 = 0 \\ u_2 = -4 \\ u_3 = -4 \\ v_1 = 6 \\ v_2 = 8 \\ v_3 = 5 \\ v_4 = 3 \end{cases}$$

Теперь находим все остальные Δ по формуле $\Delta_{ij} = u_i + v_j - C_{ij}$

C/Δ

	4	5	6	7	u
1	7 / -1	8 / -	5 / -	3 / -	0
2	2 / -	4 / -	5 / -4	9 / -10	-4
3	6 / -4	3 / 1	1 / -	2 / -3	-4
v	6	8	5	3	

$\Delta_{32} = 1 > 0$

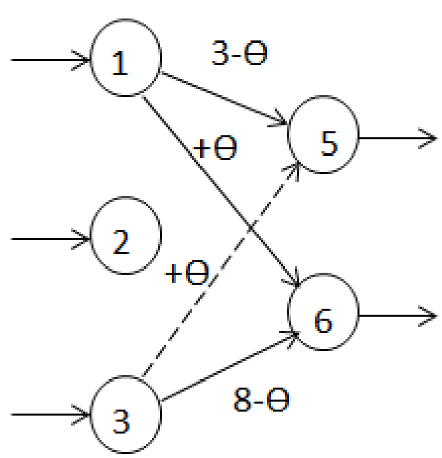


Рис. 16

$\theta^* = \min\limits_{-\theta} X_{ij} = 3$

		4	7
5	6		
	3	5	

$$f_1 = \sum_{i=1}^{3}\sum_{j=1}^{4} C_{ij}X_{ij} = 20 + 21 + 10 + 24 + 9 + 5 = 89 < f_0$$

C/Δ

	4	5	6	7	u
1	7 -2	8 -1	5 -	3 -	0
2	2 -	4 -	5 -3	9 -9	-4
3	6 -5	3 -	1 -	2 -3	-4
v	5	7	5	3	

Получили оптимальный грузопоток:

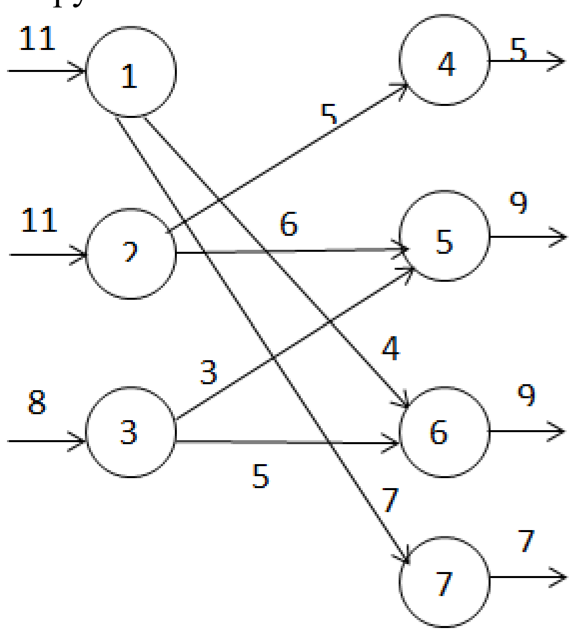


Рис. 17

8.5 Решение общей транспортной задачи путём сведения к задаче поиска кратчайшего пути и классической транспортной задаче

Общая транспортная задача решается через задачу поиска кратчайшего пути и классическую транспортную задачу.

Дан транспортный граф, состоящий из пунктов и дорог, соединяющих эти пункты. Даны интенсивности источников $d_i > 0$ и интенсивность потребителей $d_j < 0$. Могут быть пункты, не потребляющие и не производящие продуктов $d_k = 0$.

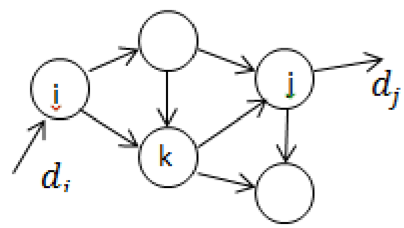


Рис.18

Дана матрица $C = \{c_{ij}\}$ – матрица промежуточных расходов, $C_{ij} \geq 0$ – стоимость провоза продукта из i-го пункта в j-ый (не обязательно полностью заполненная, не обязательно симметричная).

Требуется найти оптимальный грузопоток $X_{ij} \geq 0$

$f = \sum_{(i,j)} c_{ij} X_{ij} \rightarrow min$

Ограничения, учитывающие сбалансированность:

$$\sum_{k \in U_i^+} X_{ki} + d_i = \sum_{j \in U_i^-} X_{ij} \qquad ,i = 1,2, \dots$$

U_i^- - множество индексов, соответствующее выходящим дугам, U_i^+ - множество индексов, соответствующее входящим дугам.

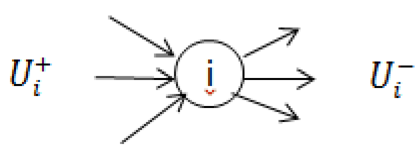


Рис. 19

$\sum_i d_i = 0$ – естественные условия баланса (весь произведенный продукт будет потреблен)

8.6 Алгоритм решения общей транспортной задачи

I. 1) Из множества всех вершин выбираем производителей и нумеруем их I=1,…,M. Выбираем всех потребителей и нумеруем J=1,…,N.

2) Делаем преобразования: $a_I = d_I > 0,\ b_J = -d_J > 0$

3) Решаем задачу поиска кратчайшего пути и находим наиболее дешевые пути от каждого производителя к каждому потребителю. Если между каким-то производителем и каким-то потребителем не найдется дороги, то $C_{I'J'} = \infty$. Найдём Π_{IJ}- самые дешевые пути от каждого производителя к каждому потребителю.

II. Решаем получившуюся классическую транспортную задачу, в которой фигурируют $a_I,\ b_J\ C_{IJ}$.

$\sum_{(i,j)} c_{ij} X_{ij} \to min$. Находим $X_{IJ}^* \geq 0$

III. Находим оптимальный грузопоток исходной общей транспортной задачи.

8.7 Ограничения на пропускную способность

Пусть в нашей транспортной задаче имеются ограничения на пропускную способность на одну или несколько дорог.

$$0 \leq x_{ij} \leq r_{ij}$$

Пусть есть ограничение на дорогу из пункта 1 в пункт 2.

$$0 \leq x_{12} \leq r$$

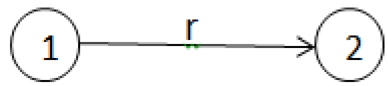


Рис. 20

Введём два промежуточных пункта (пункт 3 и пункт 4) следующим образом:

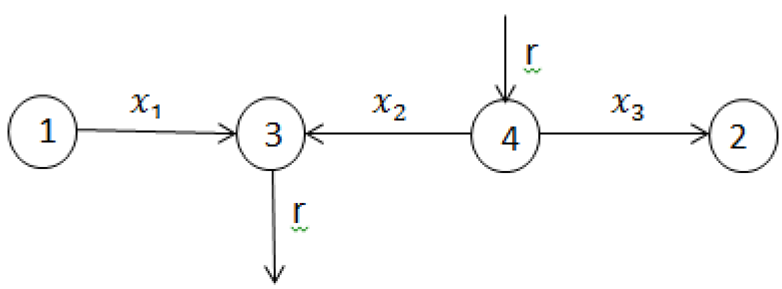


Рис. 21

Соединим пункты 3 и 4 встречным потоком и объявим пункт 4 потребителем с интенсивностью r, а пункт 3 – производителем интенсивности r.

Условие баланса:

$$\begin{cases} x_1 + x_2 = r \\ x_2 + x_3 = r \end{cases}$$

$$x_1 = x_3 = x_{12}$$

$x_1 \le r \Rightarrow x_{12} \le r$

Таким образом, если имеются ограничения на пропускную способность, то изменяем сеть путём увеличения количества пунктов с дорогами без ограничения пропускной способности.

Контрольные вопросы и задания

1. Вопросы по разделу 1:
   1.1. В чём состоит общая задача линейного программирования?
   1.2. В чём состоит стандартная задача линейного программирования?
   1.3. В чём состоит каноническая задача линейного программирования?
   1.4. Дайте определение допустимого решения задачи линейного программирования.
   1.5. Дайте определение области допустимых решений задачи линейного программирования.
   1.6. Дайте определение оптимального решения задачи линейного программирования.
2. Вопросы по разделу 2:
   2.1. Сформулируйте задачу планирования производства.
   2.2. Сформулируйте задачу о рационе.
   2.3. Сформулируйте транспортную задачу.
3. Вопросы по разделу 3:
   3.1. В чём состоит графический метод решения задачи линейного программирования?
   3.2. Перечислите способы графического решения задачи ЛП на плоскости.
   3.3. Опишите переборный метод решения задачи ЛП на плоскости.
   3.4. Опишите градиентный метод решения задачи ЛП на плоскости.
   3.5. Дайте определение линии уровня целевой функции.
   3.6. Дайте определение опорной прямой.
4. Вопросы по разделу 4:
   4.1. Как привести задачу ЛП к каноническому виду?
   4.2. Приведите пример приведения задачи ЛП к каноническому виду.
5. Вопросы по разделу 5:
   5.1. В чём состоит идея симплекс-метода?
   5.2. Дайте определение базисных и свободных переменных.
   5.3. Какие этапы включает симплекс-метод?
6. Вопросы по разделу 6:
   6.1. Опишите способ построения начального допустимого базисного решения задачи ЛП с использованием искусственных переменных.

6.2. Приведите пример построения начального допустимого базисного решения задачи ЛП с использованием искусственных переменных.

7. Вопросы по разделу 7:
   7.1. В чём состоит двойственная задача линейного программирования?
   7.2. Какие преобразования производят при переходе от исходной задачи ЛП к соответствующей двойственной задаче?
   7.3. Сформулируйте теорему двойственности.
8. Вопросы по разделу 8:
   8.1. Перечислите три основных вида транспортных задач.
   8.2. В чём состоит общая транспортная задача?
   8.3. В чём состоит классическая транспортная задача?
   8.4. В чём состоит задача определения кратчайшего пути на сети?
   8.5. Опишите способ решения классической транспортной задачи методом потенциалов.
   8.6. Опишите способ решения общей транспортной задачи путём сведения к задаче поиска кратчайшего пути и классической транспортной задаче.

Список литературы

1. Банди Б. *Методы оптимизации. Вводный курс*. – М.: Радио и связь, 1988. – 128 с.
2. Банди Б. *Основы линейного программирования*. – М.: Радио и связь, 1989. – 176 с.
3. Васильев Ф.П., Иваницкий А.Ю. *Линейное программирование*. – М.: Изд-во «Факториал Пресс», 2008. – 328 с.
4. Лесин В.В., Лисовец Ю.П. *Основы методов оптимизации*.– СПб: Лань, 2016. – 344 с.
5. Сухарев А.Г., Тимохов А.В., Фёдоров В.В. *Курс методов оптимизации* – М: ФИЗМАТЛИТ, 2011.– 384 с.

Содержание

Кудашов Вячеслав Николаевич
Селина Елена Георгиевна

Основы линейного программирования

Учебно-методическое пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверский пр., 49