

МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ
РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

Кафедра управления автомобильными перевозками
и дорожным движением

И. Н. КРАВЧЕНЯ, С. А. АЗЕМША

МАТЕМАТИЧЕСКИЕ МОДЕЛИ В ТРАНСПОРТНЫХ СИСТЕМАХ.

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

*Рекомендовано учебно-методическим объединением по образованию
в области транспорта и транспортной деятельности
для обучающихся по специальностям*

*1-44 01 01 «Организация перевозок и управление на автомобильном
и городском транспорте»,*

1-44 01 02 «Организация дорожного движения»,

*1-44 01 03 «Организация перевозок и управление
на железнодорожном транспорте»*

в качестве учебно-методического пособия

Гомель 2023

УДК 519.85(075.8)
ББК 22.18
К78

Рецензенты: кафедра информационно-вычислительных систем УО «БГЭУПК» (зав. кафедрой – канд. физ.-мат. наук, доцент *Л. П. Авдашкова*; доцент кафедры – канд. физ.-мат. наук, доцент *М. А. Грибовская*); доцент кафедры математических проблем управления и информатики, канд. техн. наук, доцент *С. И. Жогаль* (УО «ГГУ им. Ф. Скорины»)

Кравченя, И. Н.

К78 Математические модели в транспортных системах. Математическое программирование : учеб.-метод. пособие / И. Н. Кравченя, А. С. Аземша ; М-во трансп. и коммуникаций Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2023. – 142 с.
ISBN 978-985-891-098-3

Содержатся теоретические сведения по основным разделам курса «Математические модели в транспортных системах», примеры решения задач. Включает в себя разделы математического программирования: линейного, целочисленного и нелинейного программирования, динамического программирования.

Предназначено для студентов специальностей 1-44 01 01 «Организация перевозок и управление на автомобильном и городском транспорте», 1-44 01 02 «Организация дорожного движения», 1-44 01 03 «Организация перевозок и управление на железнодорожном транспорте». Может быть использовано студентами специальности 1-27 02 01 «Транспортная логистика (по направлениям)», а также при курсовом и дипломном проектировании студентами технических специальностей.

УДК 519.85(075.8)
ББК 22.18

ISBN 978-985-891-098-3

© Кравченя И. Н., Аземша С. А., 2023
© Оформление. БелГУТ, 2023

ОГЛАВЛЕНИЕ

Введение.....	5
1 Задачи линейного программирования	8
1.1 Постановка задачи линейного программирования. Формы записи задач линейного программирования	8
1.2 Графический метод решения задач линейного программирования	10
1.3 Симплексный метод решения задач линейного программирования.....	15
2 Транспортная задача линейного программирования.....	24
2.1 Отыскание кратчайших расстояний между пунктами транспортной сети.....	25
2.2 Транспортная задача линейного программирования в матричной форме.....	28
2.2.1 Постановка и математическая модель транспортной задачи	28
2.2.2 Построение начального базисного плана перевозок	31
2.2.3 Построение оптимального плана методом потенциалов	33
2.2.4 Пример решения транспортной задачи в матричной форме	36
2.2.5 Решение усложненных транспортных задач.....	40
2.3 Транспортная задача линейного программирования в сетевой форме	50
2.3.1 Постановка транспортной задачи в сетевой форме.....	50
2.3.2 Решение транспортной задачи в сетевой форме методом потенциалов..	51
2.3.3 Пример решения транспортной задачи в сетевой форме.....	54
3 Задачи целочисленного программирования	58
3.1 Основные понятия	58
3.2 Математическая модель задачи целочисленного линейного программирования	59
3.3 Методы решения задач целочисленного линейного программирования	60
3.3.1 Суть методов решения задач целочисленного линейного программирования.....	60
3.3.2 Метод отсечений (отсекающих плоскостей) Гомори.....	61
3.3.3 Метод ветвей и границ решения задач целочисленного линейного программирования.....	64
3.4 Задача о коммивояжере.....	69
3.4.1 Математическая модель задачи о коммивояжере и методы ее решения .	69
3.4.2 Решение задачи о коммивояжере методом ветвей и границ (алгоритм Литтла).....	71
3.5 Задача о назначениях (задача выбора).....	81
3.5.1 Решение задачи о назначениях венгерским методом.....	81
3.5.2 Применение задачи о назначениях для оптимизации расписания общественного транспорта на дублирующих участках.....	89
4 Задачи нелинейного программирования	96
4.1 Постановка задачи нелинейного программирования. Виды экстремумов	96
4.2 Одномерная оптимизация.....	100

4.3 Многомерная оптимизация.....	103
4.3.1 Методы оптимизации функций нескольких переменных.....	103
4.3.2 Особенности задач нелинейного программирования и графический метод решения.....	106
4.3.3 Метод неопределенных множителей Лагранжа.....	109
4.3.4 Решение задачи о производстве продукции графическим методом и методом неопределенных множителей Лагранжа.....	113
4.4 Градиентные методы.....	116
5 Динамическое программирование.....	120
5.1 Основные понятия динамического программирования.....	120
5.2 Принцип оптимальности Беллмана.....	122
5.3 Решение задачи о выборе маршрута методом динамического программирования.....	123
5.4 Решение задачи о распределении средств на расширение производства методом динамического программирования.....	128
5.5 Решение задачи о ремонте и замене оборудования методом динамического программирования.....	133
Список литературы.....	141

ВВЕДЕНИЕ

Моделирование представляет собой процесс замещения изучаемого объекта или процесса некоторой его моделью и проведение исследований на модели с целью получения необходимой информации об объекте. В общем случае под **моделью** понимается физический или абстрактный образец моделируемого объекта (или процесса), удобный для проведения исследований и позволяющий адекватно отображать интересующие исследователя свойства и характеристики объекта (процесса).

Модели позволяют:

- исследовать свойства реальных объектов и процессов;
- прогнозировать поведение объекта или процесса при различных условиях;
- определять оптимальные способы управления объектами или процессами.

В современных условиях наибольшее распространение получило **математическое моделирование**, основанное на использовании математических моделей изучаемых объектов и процессов. Масштабное и успешное применение математического моделирования вызвано, в частности, широкими возможностями для его реализации вычислительной техники.

Математическая модель представляет собой приближенное представление реальных объектов, процессов или систем, выраженное в математических терминах и отражающее существенные черты оригинала.

Необходимо отметить тот факт, что в силу высокой сложности изучаемых процессов учет всех условий реальных задач для построения их математических моделей в большинстве случаев становится невозможным. Поэтому важно понимать, что любая модель представляет оригинал только в некотором определенном смысле. Следовательно, для одного и того же объекта могут быть построены несколько различных математических моделей, отражающих его разные свойства или описывающих этот объект с разной степенью детализации.

Основная часть данного пособия будет посвящена рассмотрению математических моделей, относящихся к разделу математического программирования.

Математическое программирование – это математическая дисциплина, в которой разрабатываются методы отыскания экстремума (т. е. максимума или минимума) целевой функции среди множества ее возможных значений, определяемых ограничениями. Методы математического программирования успешно используются для решения задач планирования выпуска продукции, ценообразования, распределения ресурсов, задач транспортного типа и т. п.

Для построения математической модели необходимо иметь четкое представление, во-первых, о *цели* функционирования данной системы (которая описывается целевой функцией); во-вторых, о наборе *переменных*, которыми можно варьировать; и в третьих, об *ограничениях*, накладываемых на переменные, которые описывают реальные условия функционирования моделируемой системы.

В общем виде задача математического программирования может быть записана следующим образом:

$$z = f(X) \rightarrow \max(\min), \quad (1)$$

$$X \in \Omega. \quad (2)$$

Здесь $z = f(X)$ – целевая функция, экстремум которой требуется определить. Целевая функция отражает критерий поиска решения оптимизационной задачи. При построении математических моделей экономических процессов, целевая функция чаще всего описывает прибыль предприятия (которую требуется максимизировать) либо суммарные издержки (которые подлежат минимизации).

$X = (x_1, x_2, \dots, x_n)$ – вектор переменных задачи, т. е. набор неизвестных величин, значения которых необходимо определить в процессе решения задачи. В качестве переменных, например, могут выступать объемы производства продукции разных видов, объемы перевозок и др.

Ω – область допустимых решений задачи, определяемая набором ограничений на значения переменных. Эти ограничения отражают реальные условия изучаемого процесса и могут иметь экономический, ресурсный или временной характер. Например, ограничения могут быть вызваны заданиями на объем производимой продукции, лимитом используемых ресурсов, балансовыми соотношениями между переменными, сроками выполнения работ и др. В практических задачах *допустимые решения* часто называют *допустимыми планами*, поскольку они представляют собой возможные варианты планов производства, перевозок, управления и т. д.

Область допустимых решений может быть задана, например, с помощью системы ограничений:

$$\begin{aligned} g_i(X) &= b_i, & i = \overline{1, k}, \\ g_i(X) &\leq b_i, & i = \overline{k+1, m}. \end{aligned} \quad (3)$$

В зависимости от свойств целевой функции и ограничений среди задач математического программирования можно выделить два важнейших класса.

1 Если целевая функция $f(X)$ и все функции ограничений $g_i(X)$, $i = \overline{1, m}$ – линейные функции, то задача (1)–(3) называется **задачей линейного программирования**.

Линейное программирование является наиболее изученным разделом математического программирования. Для решения задач линейного программирования разработан целый ряд эффективных методов, алгоритмов и программ, которые рассматриваются в разделах 1–3.

2 Если хотя бы одна из функций $f(X)$ или $g_i(X)$ – нелинейны, то задача (1)–(3) называется **задачей нелинейного программирования**. Задачи этого типа рассматриваются в пятом разделе.

1 ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1 Постановка задачи линейного программирования. Формы записи задач линейного программирования

Линейное программирование (ЛП) – раздел математического программирования, содержащий методы отыскания экстремума (максимума или минимума) линейных функций нескольких переменных при линейных ограничениях, наложенных на переменные.

В общем виде задача ЛП состоит в нахождении некоторой совокупности значений переменных $X = (x_1, x_2, \dots, x_n)$, доставляющих линейной функции цели экстремальное значение и удовлетворяющих системе ограничений в виде линейных равенств или неравенств.

Различают несколько форм записи задач линейного программирования.

Задачей линейного программирования, заданной в произвольной форме записи, называют задачу, которая формулируется следующим образом.

Найти значения переменных $X = (x_1, x_2, \dots, x_n)$, который доставляет экстремум функции

$$z(X) = \sum_{j=1}^n c_j x_j \rightarrow \max(\min) \quad (1.1)$$

при системе ограничений:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m_1}, \quad (1.2)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{m_1 + 1, m_2}, \quad (1.3)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{m_2 + 1, m}, \quad (1.4)$$

$$x_j \geq 0, \quad j = \overline{1, r}, \quad r \leq n. \quad (1.5)$$

Линейная функция (1.1) называется *целевой функцией*; множество решений или планов X , удовлетворяющих системе ограничений (1.2)–(1.5), называется *множеством (областью) допустимых решений* Ω , $X \in \Omega$.

Допустимый план $X \in \Omega$, доставляющий экстремальное значение целевой функции (1.1), называется *оптимальным*.

Выделяют также *симметричную форму записи* задач ЛП, которая характеризуется тем, что если целевая функция подлежит максимизации, то все ограничения имеют вид неравенств типа « \leq ». Если же целевая функция подлежит минимизации, то все ограничения имеют вид неравенств типа « \geq ». Таким образом, симметричная форма записи может иметь вид

$$\begin{aligned} z(X) &= \sum_{j=1}^n c_j x_j \rightarrow \max, \\ \sum_{j=1}^n a_{ij} x_j &\leq b_i, i = \overline{1, m}, \\ x_j &\geq 0, j = \overline{1, n} \end{aligned} \quad (1.6)$$

или

$$\begin{aligned} z(X) &= \sum_{j=1}^n c_j x_j \rightarrow \min, \\ \sum_{j=1}^n a_{ij} x_j &\geq b_i, i = \overline{1, m}, \\ x_j &\geq 0, j = \overline{1, n}. \end{aligned} \quad (1.7)$$

Если в задаче ЛП все ограничения задачи имеют вид равенств и на все переменные величины наложено условие неотрицательности $x_j \geq 0, j = \overline{1, n}$, то задача ЛП представлена в *канонической форме записи*:

$$\begin{aligned} z(X) &= \sum_{j=1}^n c_j x_j \rightarrow \max, \\ \sum_{j=1}^n a_{ij} x_j &= b_i, i = \overline{1, m}, \\ x_j &\geq 0, j = \overline{1, n}. \end{aligned} \quad (1.8)$$

Перечисленные выше формы записи задач ЛП эквивалентны в том смысле, что от задачи ЛП, записанной в одной из форм, можно перейти к задаче ЛП в другой форме. Поясним, как это можно сделать.

От задачи минимизации целевой функции можно перейти к задаче максимизации, и наоборот. Задача минимизации целевой функции $z(X)$ эквивалентна задаче максимизации функции $-z(X)$:

$$\min z(X) = -\max(-z(X)). \quad (1.9)$$

Чаще всего ограничения в задаче ЛП имеют вид неравенств. Однако для применения универсальных методов решения задач ЛП необходимо использовать запись задачи ЛП в канонической форме. Любую задачу линейного программирования можно свести к задаче линейного программирования в канонической форме записи.

Правило приведения задачи линейного программирования (1.1)–(1.5) к канонической форме записи состоит в следующем:

1) если в исходной задаче требуется минимизировать целевую функцию $z(X)$, то необходимо перейти от задачи на минимум к задаче на максимум на основании соотношения (1.9);

2) если среди ограничений есть неравенства, то необходимо перейти от ограничений в виде неравенств к ограничениям в виде равенств. Для этого вводят дополнительные неотрицательные переменные $x_{n+i} \geq 0$, $i = m_1 + 1, m$, которые прибавляются к левым частям ограничений (1.3) и вычитаются из левых частей ограничений (1.4). В целевую функцию (1.1) дополнительные переменные вводятся с коэффициентами, равными нулю.

1.2 Графический метод решения задач линейного программирования

Графический метод решения задач линейного программирования применим для решения задач с двумя переменными, когда ограничения выражены неравенствами. Он также может быть использован для решения задач ЛП с числом переменных $n > 2$, если в ее канонической записи число неизвестных n и число линейно независимых уравнений m связано соотношением: $n - m \leq 2$. В этом случае от задачи в канонической форме записи можно перейти к задаче в симметричной форме, которая будет содержать не более двух переменных. Решая эту задачу графическим методом, находят переменные оптимального плана. Подставляя их в ограничения исходной задачи, определяют и остальные неизвестные оптимального решения.

Пусть задача линейного программирования имеет две переменные:

$$z(X) = c_1 x_1 + c_2 x_2 \rightarrow \max (\min), \quad (1.10)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 \leq b_2, \\ \dots \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 \leq b_m, \\ x_1 \geq 0, \quad x_2 \geq 0. \end{cases} \quad (1.11)$$

Дадим геометрическую интерпретацию составляющих этой задачи.

Каждое из неравенств (1.11) системы ограничений определяет на плоскости $x_1 O x_2$ полуплоскость, ограниченную прямой $a_{i1}x_1 + a_{i2}x_2 = b_i$ ($i = \overline{1, m}$). А пересечение полученных полуплоскостей представляет собой область допустимых (ОДР) решений Ω . Если же это множество ограничено, то оно называется *многоугольником решений*.

Особенностью задач линейного программирования является то, что область допустимых решений Ω является выпуклым множеством, и целевая функция достигает экстремума, если он существует, на границе области допустимых решений или в одной из угловых (крайних) точек ОДР.

С геометрической точки зрения множество точек X таких, что $z(X) = \text{const}$, определяет на плоскости семейство параллельных прямых, называемых *линиями уровня целевой функции*, каждой из которых соответствует определенное значение целевой функции $z(X)$.

Вектор $\vec{C} = (\partial z / \partial x_1; \partial z / \partial x_2) = (c_1; c_2)$ называется *градиентом*, или *вектором градиентного направления*, и показывает направление наискорейшего возрастания целевой функции $z(X)$. Вектор, противоположный вектору \vec{C} : $-\vec{C} = (-c_1; -c_2)$, называется *антиградиентом* и показывает направление наискорейшего убывания целевой функции.

Из геометрической интерпретации следует **графический метод решения задачи ЛП** в случае двух переменных.

1 С учетом системы ограничений (1.11) строим область допустимых решений Ω , которая может представлять собой либо выпуклый многоугольник (рисунок 1.1, а), либо неограниченную область (рисунок 1.1, б), либо пустое множество (рисунок 1.1, в).

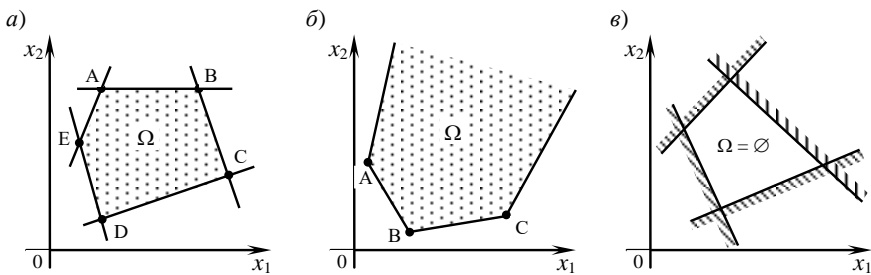


Рисунок 1.1 – Варианты различных областей допустимых решений

2 Строим вектор $\vec{C} = (c_1; c_2)$ градиентного направления целевой функции.

3 Перпендикулярно вектору \vec{C} проводим произвольную линию уровня целевой функции $z(X) = \text{const}$ (например $z(X) = 0$).

4 При решении задачи на максимум перемещаем линию уровня $z(X) = \text{const}$ в направлении, определяемом вектором \vec{c} , до ее последней точки пересечения с областью допустимых решений Ω . В случае решения задачи на минимум линию уровня целевой функции $z(X) = \text{const}$ перемещаем в антиградиентном направлении до ее последней общей точки с ОДР.

5 Определяем оптимальное решение $X^* = (x_1^*; x_2^*)$ и экстремальное значение целевой функции $z^* = z(x_1^*; x_2^*)$.

На рисунке 1.2 приведены примеры возможных вариантов решения задачи линейного программирования.

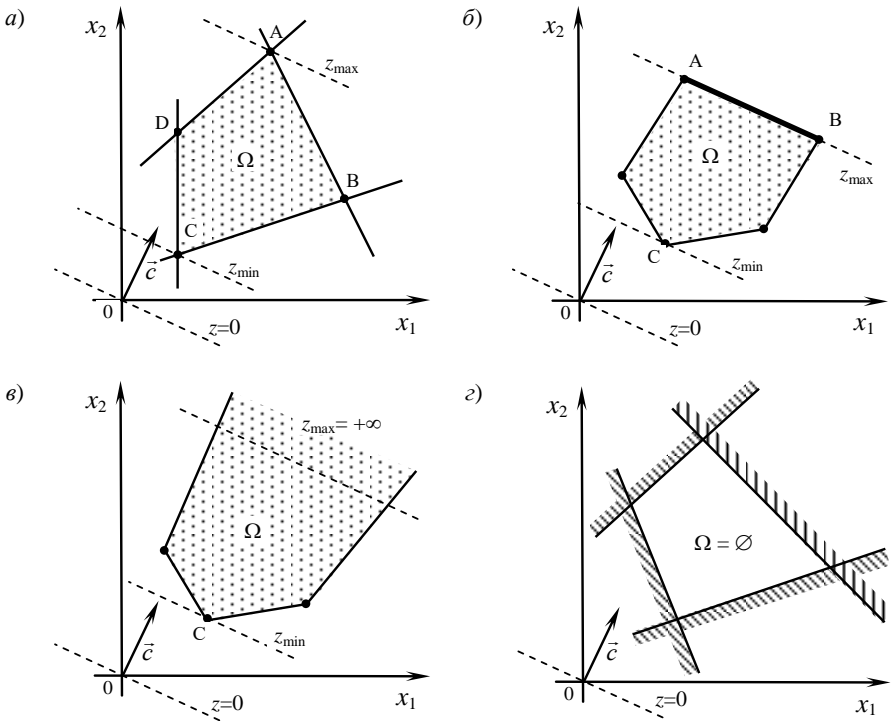


Рисунок 1.2 – Варианты графического решения задач ЛП

На рисунке 1.2, а изображен случай, когда линия уровня целевой функции и область допустимых решений Ω в предельных положениях имеют одну общую точку. В точке A целевая функция достигает максимального значения, а в точке C – минимального.

На рисунке 1.2, б показан случай, когда линия уровня целевой функции параллельна отрезку AB , принадлежащему области допустимых решений Ω .

В этом случае максимум функции z достигается в точке A и в точке B , а следовательно, и в любой точке отрезка AB . Такие решения называются *альтернативными оптимальными решениями*, а множество оптимальных решений представляет собой линейную комбинацию координат угловых точек $A = X_1^*$ и $B = X_2^*$:

$$X^* = \lambda_1 X_1^* + \lambda_2 X_2^*, \quad \lambda_1 + \lambda_2 = 1, \quad \lambda_1 \geq 0, \quad \lambda_2 \geq 0.$$

Минимум функции z на рисунке 1.2, *б* достигается в точке C .

На рисунке 1.2, *в* показан случай, когда область допустимых решений Ω представляет собой неограниченное сверху множество. При решении задачи на максимум целевая функция z не ограничена, так как линия уровня, сколько бы ее не перемещали в направлении градиента, будет иметь общие точки с областью допустимых решений Ω . Оптимального решения задачи на максимум в данном случае не существует. Минимального значения функция z достигает в точке C .

На рисунке 1.2, *г* представлен случай, когда задача ЛП не имеет решений. Область допустимых решений Ω в этом случае – пустое множество, т. е. система ограничений (1.11) несовместна.

Рассмотрим пример решения задачи ЛП графическим методом.

Пример 1.1.

$$\begin{aligned} z(X) &= 3x_1 + 4x_2 \rightarrow \max (\min), \\ x_1 + x_2 &\geq 2, \\ 5x_1 + 3x_2 &\leq 30, \\ x_2 &\leq 5, \\ 5x_1 - 3x_2 &\leq 15, \\ x_1 \geq 0; \quad x_2 &\geq 0. \end{aligned}$$

Для построения области допустимых решений Ω строим в системе координат x_1Ox_2 прямые, соответствующие ограничениям (рисунок 1.3) :

$$\begin{aligned} l_1 : x_1 + x_2 &= 2; \\ l_2 : 5x_1 + 3x_2 &= 30; \\ l_3 : x_2 &= 5; \\ l_4 : 5x_1 - 3x_2 &= 15. \end{aligned}$$

Изобразим полуплоскости, определяемые системой ограничений. Найдим множество допустимых решений Ω как пересечение (общую часть) полученных полуплоскостей – многоугольник $ABCDEF$.

Строим вектор градиентного направления $\vec{C} = (\partial z / \partial x_1; \partial z / \partial x_2) = (3; 4)$.

Строим произвольную линию уровня целевой функции, например $z(X) = 0$, перпендикулярную к вектору \vec{C} . Перемещаем прямую $z(X) = 0$ в направлении, указываемом вектором \vec{C} до последней точки ее пересечения с областью допустимых решений Ω .

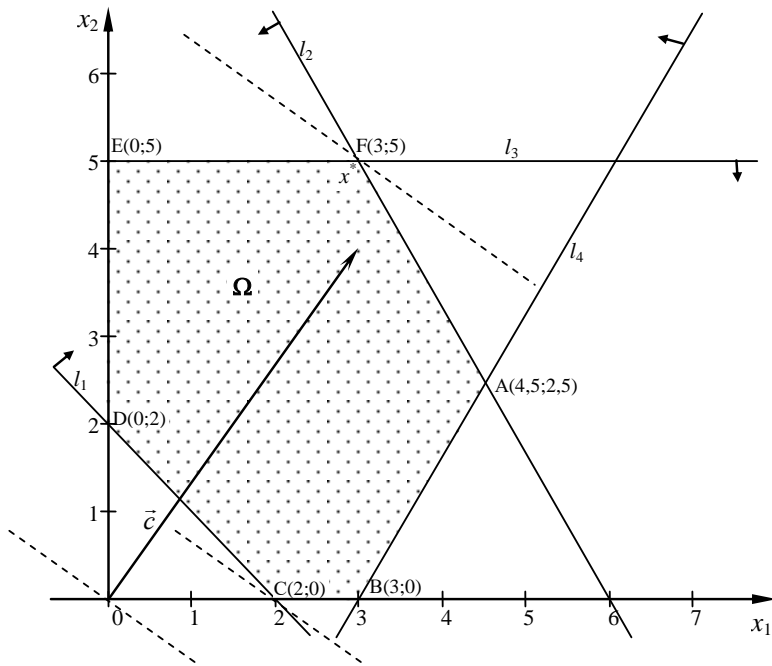


Рисунок 1.3 – Графическое решение задачи

В точке F целевая функция достигает максимума. Находим координаты точки F – точки пересечения прямых l_2 и l_3 путем решения системы уравнений:

$$\begin{cases} 5x_1 + 3x_2 = 30 \\ x_2 = 5 \end{cases} \Rightarrow x_1 = 3, x_2 = 5.$$

Получено оптимальное решение $X_{\max}^* = (3; 5)$.

Вычислим оптимальное значение целевой функции

$$z_{\max} = z(X_{\max}^*) = 3 \cdot 3 + 4 \cdot 5 = 29.$$

В точке C с координатами $(2; 0)$ целевая функция достигает минимума.

Получено оптимальное решение $X_{\min}^* = (2; 0)$.

Вычислим оптимальное значение целевой функции

$$z_{\min} = z(X_{\min}^*) = 3 \cdot 2 + 4 \cdot 0 = 6.$$

1.3 Симплексный метод решения задач линейного программирования

Рассмотренный выше графический метод решения задач линейного программирования применим к весьма узкому классу задач ЛП: эффективно им можно решать задачи, содержащие не более двух переменных.

Одним из универсальных методов решения задач ЛП, применимым для решения задач с любым числом переменных, является *симплексный метод* (*симплекс-метод*), называемый также методом последовательного улучшения плана.

Пусть задача линейного программирования представлена в симметричной форме записи:

$$z(X) = \sum_{j=1}^n c_j x_j \rightarrow \max; \quad (1.12)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad (1.13)$$
$$x_j \geq 0, \quad j = \overline{1, n}.$$

Дадим *геометрическую интерпретацию решения задачи ЛП симплексным методом*.

Допустимый план задачи ЛП называется *базисным*, если в области допустимых решений Ω (многограннике решений) ему соответствует угловая (крайняя) точка.

Если задача линейного программирования (1.12)–(1.13) имеет решение, то ее оптимальный план совпадает по крайней мере с одним из базисных решений (планов) системы ограничений (1.13). Поэтому для решения задачи ЛП симплексным методом необходимо перебрать конечное число базисных планов и выбрать среди них тот, на котором целевая функция принимает экстремальное значение.

С геометрической точки зрения перебор базисных планов можно представить как переход по ребрам от одной вершины многогранника допустимых решений к другой, в которой целевая функция принимает значение, не худшее, чем в предыдущей. В результате такого перебора будет получено оптимальное решение.

На рисунке 1.4 приведена геометрическая интерпретация идеи симплексного метода в случае двух переменных.

Решение задачи (1.12), (1.13) складывается из двух этапов: на первом находят какой-либо начальный базисный план, на втором – по специальным правилам переходят от начального плана к другому, более близкому к оптимальному плану, затем к следующему, и так до тех пор, пока не будет получено оптимальное решение.

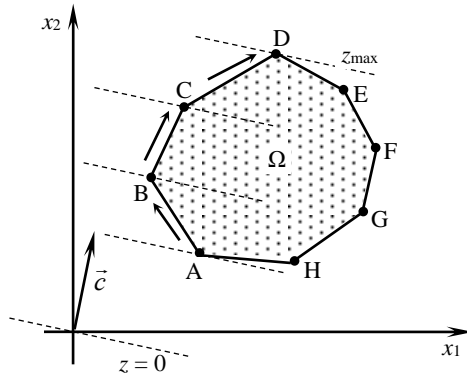


Рисунок 1.4 – Геометрическая интерпретация симплексного метода

Нахождение начального базисного плана

Пусть задача ЛП представлена в канонической форме записи, т. е. все ограничения имеют вид равенств. Переменные, которые входят в левую часть одного из ограничений с коэффициентом, равным единице, а во все остальные ограничения с коэффициентами, равными нулю (при неотрицательности правых частей), называются *базисными*. Переменные, не являющиеся базисными, называются *свободными (небазисными)* переменными. В текущем решении небазисные переменные всегда принимают нулевые значения.

Таким образом, начальный базисный план можно получить, приравняв к нулю все небазисные переменные. Тогда базисные переменные примут значения, равные правым частям ограничений.

Если задача ЛП имеет вид (1.12), (1.13), то приведем ее к канонической форме записи, добавив к левым частям системы ограничений (1.13) дополнительные переменные $x_{n+i} \geq 0$, $i = \overline{1, m}$. В целевую функцию (1.12) дополнительные переменные вводятся с коэффициентами, равными нулю: $c_{n+i} = 0$, $i = \overline{1, m}$. Тем самым задача (1.12)–(1.13) будет приведена к канонической форме записи:

$$z(X) = \sum_{j=1}^n c_j x_j \rightarrow \max,$$

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i, \quad i = \overline{1, m}, \quad (1.14)$$

$$x_j \geq 0, \quad j = \overline{1, n+m}.$$

Дополнительные переменные $x_{n+i} \geq 0$ ($i = \overline{1, m}$) будут являться базисными переменными. Следовательно, *начальный базисный план* задачи будет иметь следующий вид:

$$X^0 = \underbrace{(0, 0, \dots, 0)}_n, \underbrace{(b_1, b_2, \dots, b_m)}_m. \quad (1.15)$$

Построение оптимального плана

Введем обозначения:

$$\Delta_0 = c_1 b_1 + c_2 b_2 + \dots + c_m b_m = C_{\bar{A}} A_0, \quad (1.16)$$

где $C_{\bar{B}} = (c_1, c_2, \dots, c_m)$ – вектор коэффициентов целевой функции при базисных переменных; $A_0 = (b_1, b_2, \dots, b_m)^T$ – вектор-столбец свободных членов у системы ограничений, представленной в канонической форме записи.

Значения Δ_j называются *оценками свободных переменных*:

$$\Delta_j = c_1 a_{1j} + c_2 a_{2j} + \dots + c_m a_{mj} = C_{\bar{B}} A_j - c_j, \quad j = \overline{1, n}, \quad (1.17)$$

где $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$ – вектор-столбец коэффициентов при переменных x_j ; c_j – коэффициент целевой функции при базисной переменной x_j .

Базисный план X^0 является *оптимальным* (доставляет целевой функции максимальное значение $z = z(X^0) = \Delta_0$), если все оценки свободных переменных неотрицательны.

Занесем условие задачи, представленной в канонической форме записи (1.14), в таблицу, которая называется *симплексной* (таблица 1.1).

Таблица 1.1 – Симплексная таблица для начального базисного решения X^0

БП	$C_{\bar{B}}$	x_1	x_2	...	x_n	x_{n+1}	x_{n+2}	...	x_{n+m}	Решение
x_{n+1}	c_{n+1}	a_{11}	a_{12}	...	a_{1n}	1	0	...	0	b_1
x_{n+2}	c_{n+2}	a_{21}	a_{22}	...	a_{2n}	0	1	...	0	b_2
.
.
.
x_{n+m}	c_{n+m}	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m
z -строка		Δ_1	Δ_2	...	Δ_n	0	0	...	0	Δ_0

В первом столбце симплексной таблицы приводится список базисных переменных. В столбце $C_{\bar{B}}$ содержатся коэффициенты целевой функции, стоящие при базисных переменных. Далее в таблице следуют столбцы x_1, x_2, \dots, x_{n+m} , в которых указаны коэффициенты a_{ij} системы ограничений.

Столбец «решение» – столбец свободных членов $A_0 = (b_1, b_2, \dots, b_m)^T$ системы ограничений, в котором находятся значения базисных переменных (решение задачи). Последняя строка таблицы называется z -строкой, в ней расположены коэффициенты Δ_0 (значение целевой функции) и Δ_j , которые рассчитываются по формулам (1.16), (1.17).

Для проверки условия оптимальности текущего решения просматриваем знаки коэффициентов z -строки. Если все оценки $\Delta_j \geq 0$, то текущее базисное решение является оптимальным и $z^* = z(X^*)$.

Если не все $\Delta_j \geq 0$, то необходимо перейти к новому базисному плану. Среди значений $\Delta_j < 0$ находим наибольшее по абсолютной величине и соответствующий ему *столбец* j_0 выбираем в качестве *ведущего столбца*. Соответствующую переменную x_{j_0} необходимо ввести в базис. При этом одну из переменных текущего базисного решения необходимо вывести из базиса.

Возможны два случая.

Первый случай (признак неограниченного возрастания целевой функции). Если в ведущем столбце нет ни одного положительного элемента, т. е. все элементы $a_{ij_0} \leq 0$ для всех $i = \overline{1, m}$, то целевая функция на множестве допустимых решений не ограничена сверху, т. е. задача ЛП не имеет оптимального решения (геометрическая интерпретация данной ситуации для случая двух переменных приведена на рисунке 1.1, в).

Второй случай. Если среди элементов ведущего столбца имеются положительные, то для каждого элемента $a_{ij_0} > 0$ ведущего столбца вычисляем отношения $\theta = b_i / a_{ij_0}$, среди которых выбираем наименьшее:

$$\min \left\{ \frac{b_i}{a_{ij_0}} \right\} = \frac{b_{i_0}}{a_{i_0 j_0}}. \quad (1.18)$$

В качестве *ведущей строки* выбираем строку i_0 .

Элемент $a_{i_0 j_0}$, расположенный на пересечении ведущего столбца и ведущей строки, будем называть *ведущим элементом*.

Выполним переход к «новой» симплексной таблице, соответствующей новому базисному решению, по следующим формальным правилам, которые называются *симплексными преобразованиями*:

1 Элементы ведущей строки новой таблицы $a'_{i_0 j}$ и b'_{i_0} равны соответствующим элементам старой таблицы $a_{i_0 j}$ и b_{i_0} , разделенным на ведущий элемент $a_{i_0 j_0}$:

$$a'_{i_0 j} = \frac{a_{i_0 j}}{a_{i_0 j_0}}, \quad j = \overline{1, n}; \quad b'_{i_0} = \frac{b_{i_0}}{a_{i_0 j_0}}. \quad (1.19)$$

2 Элементы ведущего столбца новой таблицы $a'_{i j_0}$ равны нулю, за исключением ведущего элемента, равного единице:

$$a'_{i j_0} = 0 \quad (i \neq i_0) \quad a'_{i_0 j_0} = 1. \quad (1.20)$$

3 Чтобы вычислить значение всех остальных элементов новой симплексной таблицы $a'_{i j}$ и b'_i , воспользуемся *правилом прямоугольника*. Элементы новой симплексной таблицы находим по формулам:

$$a'_{i j} = a_{i j} - \frac{a_{i j_0} a_{i_0 j}}{a_{i_0 j_0}}; \quad b'_i = b_i - \frac{a_{i j_0} b_{i_0}}{a_{i_0 j_0}}; \quad (1.21)$$

$$j = \overline{1, n}; \quad i = \overline{1, m}; \quad i \neq i_0,$$

где $a_{i j}$, b_i – элементы старой таблицы; $a_{i j_0}$ – элементы ведущего столбца; $a_{i_0 j}$, b_{i_0} – элементы ведущей строки, $a_{i_0 j_0}$ – ведущий элемент.

Правило прямоугольника можно пояснить следующим образом, используя фрагмент симплексной таблицы 1.1 (рисунок 1.5), содержащей элементы, входящие в формулу (1.21).

Для вычисления нового значения некоторого элемента a_{ij} симплексной таблицы, мысленно построим прямоугольник, вершинами которого будут соответствующий элемент ведущей строки $a_{i_0 j}$, соответствующий элемент ведущего столбца $a_{i j_0}$ и ведущий элемент $a_{i_0 j_0}$.

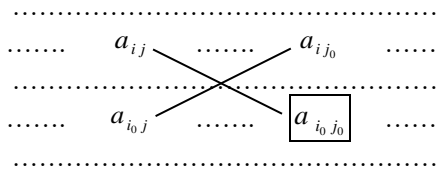


Рисунок 1.5 – Правило прямоугольника

Диагональ прямоугольника, содержащую ведущий и искомый элементы новой таблицы, назовем *главной*, а другую – *побочной*. Тогда, в соответствии с правилом прямоугольника, для получения нового значения элемента $a'_{i j}$ из значения a_{ij} вычитаем произведение элементов с побочной диагонали, деленных на ведущий элемент с главной диагонали.

4 По этому же правилу могут быть вычислены все элементы z -строки:

$$\Delta'_j = \Delta_j - \frac{\Delta_{j_0} a_{i_0 j}}{a_{i_0 j_0}}; \quad \Delta'_0 = \Delta_0 - \frac{\Delta_{j_0} b_{i_0}}{a_{i_0 j_0}}. \quad (1.22)$$

Последовательность операций на шагах 1–4, позволяющая перейти от одного базисного плана к другому, не худшему, называется *итерацией симплексного метода*.

Получив новую симплексную таблицу, снова производим проверку условия оптимальности полученного решения.

Итак, **нахождение оптимального плана симплексным методом** включает следующие шаги.

1 Записываем математическую модель задачи в канонической форме записи.

2 Находим начальный базисный план.

3 Заносим условие задачи, представленной в канонической форме записи в симплексную таблицу 1.1.

4 Проверяем выполнение условия оптимальности. Если условие оптимальности выполняется, то задача решена, $z^* = z(X^*)$. В противном случае, переходим к следующему шагу.

5 В симплексной таблице находим ведущий столбец. Проверяем выполнение условия неограниченности целевой функции. Если оно выполняется, то целевая функция неограниченна ($\max z = \infty$), т. е. оптимального решения не существует. В противном случае определяем ведущую строку. Элемент, расположенный на пересечении ведущего столбца и ведущей строки выбираем в качестве ведущего элемента.

6 По правилам 1–4 (формулы (1.19)–(1.22)) переходим к «новой» симплексной таблице и далее – к шагу 4 проверки условия оптимальности полученного решения.

Альтернативные решения. Если среди оценок Δ_j свободных переменных в последней симплексной таблице имеются нулевые, то полученное оптимальное решение задачи не является единственным, т. е. существуют *альтернативные оптимальные решения*. Нулевое значение оценки свободной переменной свидетельствует о том, что при введении ее в базис, значение целевой функции не изменится, но при этом будет получено новое оптимальное решение. Таким образом, целевая функция будет достигать экстремального значения в любой точке, являющейся линейной комбинацией крайних точек $X_1^*, X_2^*, \dots, X_K^*$, в которых целевая функция достигает экстремального значения. В этом случае общее решение задачи примет вид:

$$X^* = \sum_{k=1}^K \lambda_k X_k^*, \quad \sum_{k=1}^K \lambda_k = 1, \quad \lambda_k \geq 0, \quad k = \overline{1, K}. \quad (1.23)$$

Геометрическая интерпретация данной ситуации для случая двух переменных приведена на рисунке 1.2, б.

Рассмотрим пример решения задачи ЛП симплексным методом.

Пример 1.2.

$$\begin{aligned} z(X) &= 12 x_1 + 20 x_2 \rightarrow \max \\ 50 x_1 + 40 x_2 &\leq 6000, \\ 6 x_1 + 2 x_2 &\leq 480, \\ x_2 &\leq 100, \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Найдем решение задачи табличным симплексным методом.

Запишем математическую модель задачи в канонической форме записи, добавив к левым частям системы ограничений дополнительные переменные x_3, x_4, x_5 , которые показывают величину неиспользованных ресурсов. В целевую функцию дополнительные переменные вводятся с коэффициентами, равными нулю.

$$\begin{aligned} z(X) &= 12 x_1 + 20 x_2 + 0 x_3 + 0 x_4 + 0 x_5 \rightarrow \max, \\ 50 x_1 + 40 x_2 + x_3 &= 6000, \\ 6 x_1 + 2 x_2 + x_4 &= 480, \\ x_2 + x_5 &= 100, \\ x_j \geq 0, j &= \overline{1, 5}. \end{aligned}$$

где x_3, x_4, x_5 – базисные переменные, x_1, x_2 – свободные переменные.

Начальный допустимый базисный план получим, приравняв к нулю все небазисные переменные $x_1 = 0, x_2 = 0$. Тогда базисные переменные примут значения, равные правым частям ограничений: $x_3 = 6000, x_4 = 480, x_5 = 100$.

Начальный допустимый базисный план: $X^0 = (0; 0; 6000; 480; 100)$.

Значение целевой функции при начальном плане

$$z(X^0) = 12 \cdot 0 + 20 \cdot 0 + 0 \cdot 6000 + 0 \cdot 480 + 0 \cdot 100 = 0.$$

Заполним первую симплексную таблицу (таблица 1.2).

Таблица 1.2 – Симплексная таблица начального допустимого решения X^0

БП	СБ	x_1	x_2	x_3	x_4	x_5	Решение	θ
x_3	0	50	40	1	0	0	6000	150
x_4	0	6	2	0	1	0	480	240
x_5	0	0	1	0	0	1	100	100
z-строка		-12	-20	0	0	0	0	

В столбце БП записываем базисные переменные. Столбец C_B содержит коэффициенты целевой функции, стоящие при базисных переменных $c_3 = 0$, $c_4 = 0$, $c_5 = 0$. Основное поле таблицы занимают коэффициенты a_{ij} системы ограничений, представленной в канонической форме записи. Столбец «решение» – столбец свободных членов системы ограничений.

Более подробно остановимся на заполнении z -строки. Здесь расположено значение целевой функции $\Delta_0 = z(X^0) = 0$ и оценки Δ_j , которые рассчитываются по формуле (1.22):

$$\Delta_1 = 0 \cdot 50 + 0 \cdot 6 + 0 \cdot 0 - 12 = -12,$$

$$\Delta_2 = 0 \cdot 40 + 0 \cdot 2 + 0 \cdot 1 - 20 = -20 \text{ и т. д.}$$

В z -строке среди оценок Δ_j есть отрицательные, следовательно, план X^0 не является оптимальным. Среди значений $\Delta_j < 0$ находим наибольшее по абсолютной величине (-20), столбец x_2 выбираем в качестве ведущего. Для положительных элементов ведущего столбца вычисляем симплексное отношение θ (оно приведено в последнем столбце таблицы) и из полученных значений выбираем наименьшее $\min \theta = \min \{150; 240; 100\} = 100$. Соответствующую строку x_5 выбираем в качестве ведущей. Элемент 1, расположенный на пересечении ведущего столбца и ведущей строки, является ведущим элементом. Таким образом, для улучшения текущего базисного решения необходимо ввести в базис переменную x_2 и вывести из него переменную x_5 .

В соответствии с формулами (1.19)–(1.21) переходим к следующей симплексной таблице (таблица 1.3).

Таблица 1.3 – Симплексная таблица улучшенного решения X^1

БП	C_B	x_1	x_2	x_3	x_4	x_5	Решение	θ
x_3	0	50	0	1	0	-40	2000	40
x_4	0	6	0	0	1	-2	280	46,67
x_2	20	0	1	0	0	1	100	–
z -строка		-12	0	0	0	20	2000	

Элементы ведущей строки делим на ведущий элемент $a_{32} = 1$. Элементы ведущего столбца новой таблицы заполняем нулями, кроме ведущего элемента $a'_{32} = 1$. Все остальные элементы таблицы пересчитываем по правилу прямоугольника. Например,

$$a'_{11} = 50 - \frac{40 \cdot 0}{1} = 50; \quad a'_{15} = 0 - \frac{40 \cdot 1}{1} = -40; \quad a'_{21} = 6 - \frac{2 \cdot 0}{1} = 6;$$

$$b'_1 = 6000 - \frac{40 \cdot 100}{1} = 2000; \quad b'_2 = 480 - \frac{2 \cdot 100}{1} = 280;$$

$$\Delta'_0 = 0 - \frac{-20 \cdot 100}{1} = 2000; \quad \Delta'_1 = -12 - \frac{-20 \cdot 0}{1} = -12;$$

$$\Delta'_5 = 0 - \frac{-20 \cdot 1}{1} = 20.$$

$$X^1 = (0; 100; 2000; 280; 0), \quad z(X^1) = 2000.$$

План X^1 не является оптимальным. Переходим к следующей симплексной таблице (таблица 1.4).

Таблица 1.4 – Симплексная таблица оптимального решения X^*

БП	C_b	x_1	x_2	x_3	x_4	x_5	Решение
x_1	12	1	0	0,02	0	-0,8	40
x_4	0	0	0	-0,12	1	2,8	40
x_2	20	0	1	0	0	1	100
z -строка		0	0	0,24	0	10,4	2480

$$X^2 = (40; 100; 0; 40; 0), \quad z(X^2) = 2480.$$

В z -строке среди оценок Δ_j нет отрицательных, следовательно, план X^2 является оптимальным. А так как оценки свободных переменных x_3 и x_5 не равны нулю ($\Delta_3 = 0,24$; $\Delta_5 = 10,4$), то полученное оптимальное решение задачи является единственным.

$$X^* = (40; 100; 0; 40; 0), \quad z^* = z(X^*) = 2480.$$

2 ТРАНСПОРТНАЯ ЗАДАЧА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Под термином «*транспортные задачи*» (ТЗ) понимается широкий круг задач не только транспортного характера. Общим для них является, как правило, распределение ресурсов, находящихся у m производителей (поставщиков), по n потребителям этих ресурсов.

Наиболее часто встречаются следующие задачи, относящиеся к транспортным: прикрепление потребителей ресурса к производителям; привязка пунктов отправления к пунктам назначения; взаимная привязка грузопотоков прямого и обратного направлений; задачи оптимальной загрузки промышленного оборудования; оптимальное распределение объемов выпуска промышленной продукции между заводами-изготовителями и др.

Транспортную задачу можно применять при рассмотрении ряда практических ситуаций, связанных с управлением запасами, составлением сменных графиков, назначением служащих на рабочие места, оборотом наличного капитала и многими другими.

Кроме того, модель транспортной задачи можно видоизменить, с тем чтобы она учитывала перевозку нескольких видов продукции (многопродуктовая ТЗ). Если перевозки связаны с поставкой груза через промежуточные пункты, то задача будет многоэтапной. При этом допускается перевозка груза (частично или полностью) через другие исходные пункты или пункты назначения транзитом, прежде чем он достигнет установленного пункта назначения.

Транспортная сеть представляет систему путей сообщения, которые предназначены для передвижения транспортных средств. Модель транспортной сети – это описание местонахождения ее вершин (пунктов), а также длин и других параметров (характеристик), соединяющих вершины ребрами. Транспортная сеть может быть задана в графическом или табличном виде. Ребром транспортной сети является ее часть, соединяющая две смежные вершины. Длина ребер может быть выражена в единицах длины, приведенных единицах длины, единицах времени и стоимостном выражении. В качестве других параметров ребер может быть задана техническая категория пути сообщения и др. Вершины транспортной сети соответствуют местонахождению ресурсообразующих и ресурсопоглощающих точек, терминалов, пересечений путей.

2.1 Отыскание кратчайших расстояний между пунктами транспортной сети

Постановка задачи определения кратчайших расстояний на сети. Пусть задана схема транспортной сети, отражающая возможные варианты передвижения между пунктами этой сети и длины всех ее ребер (т. е. расстояния между пунктами). Требуется определить кратчайшие расстояния между всеми пунктами этой сети.

Решение этой задачи может быть найдено, например, методом потенциалов. При использовании метода потенциалов, один из пунктов выбирается в качестве начального и определяются минимальные расстояния от начального до каждого из оставшихся пунктов сети. Затем поочередно в качестве начального выбираются все остальные пункты сети и в результате подобных расчетов будет получена матрица кратчайших расстояний между любыми двумя пунктами этой транспортной сети.

Алгоритм метода потенциалов для определения кратчайших расстояний между пунктами транспортной сети.

Шаг 1. Выбирается начальный пункт (обозначим его номером 1), от которого требуется определить расстояния до каждого из остальных пунктов. Начальному пункту присваивается значение потенциала, равное нулю: $v_1 = 0$. На схеме значения потенциалов записываются рядом с соответствующей вершиной, для наглядности его можно заключить в рамку.

На данном этапе множество вершин с известным потенциалом содержит только одну вершину (вершину номер 1), а множество вершин с неизвестным потенциалом – все остальные вершины сети.

Шаг 2. Рассматриваются все ребра (i, j) транспортной сети, для которых потенциал начальной вершины уже определен, а потенциал конечной вершины – еще не определен.

Для каждого из таких ребер (i, j) вычисляется значение суммы потенциала вершины i и длины ребра (i, j) :

$$U(i, j) = v_i + l_{ij},$$

где v_i – потенциал вершины i ; l_{ij} – длина ребра (i, j) .

Шаг 3. Среди вычисленных значений $U(i, j)$ выбирается минимальное (обозначим это значение $U(k, m)$). Ребро (k, m) помечается стрелкой, направленной от вершины k к вершине m . Вершине m присваивается значение потенциала, равное $U(k, m)$: $v_m = U(k, m)$. После этого вершина m исключается из множества вершин сети с неизвестным потенциалом и добавляется во множество вершин с известным потенциалом.

Возможна ситуация, когда минимальное значение $U(i, j)$ достигается на нескольких ребрах.

При этом различают два случая: ребра, которым соответствует минимум $U(i, j)$ «подходят» к одной и той же вершине или к разным вершинам.

Если эти ребра подходят к одной и той же вершине m , то потенциал вершины m принимается равным минимальному значению $U(i, j)$, и все ребра, на которых достигается минимальное значение $U(i, j)$, выделяются стрелками. Эта ситуация указывает на то, что существует несколько кратчайших путей от начальной вершины до вершины m .

В случае, если ребра подходят к разным вершинам, каждой из вершин, к которой подходит ребро с минимальным значением $U(i, j)$ присваивается значение потенциала, равное этому минимальному значению. Соответствующие ребра, подходящие к этим вершинам помечаются стрелками.

Вершины, для которых на этой итерации были вычислены значения потенциалов, перемещаются из множества вершин с неизвестным потенциалом во множество вершин с известным потенциалом.

Шаг 4. Если множество вершин сети с неизвестными потенциалами еще не пусто, то снова переходим к шагу 2 алгоритма. В противном случае решение задачи закончено.

Вычисленные значения потенциалов каждой из вершин представляют собой минимальные расстояния от начального пункта до соответствующей вершины. Маршрут кратчайшего пути от начальной вершины до остальных выделен следующими друг за другом стрелками.

Пример 2.1.

Постановка задачи. На рисунке 2.1 изображена схема дорог, связывающая пункты 1–7. Расстояния между пунктами сети известны и приведены на рисунке. Требуется, используя метод потенциалов, определить кратчайшие расстояния от пункта 1 до каждого из остальных пунктов этой сети, построить кратчайшую связывающую сеть.

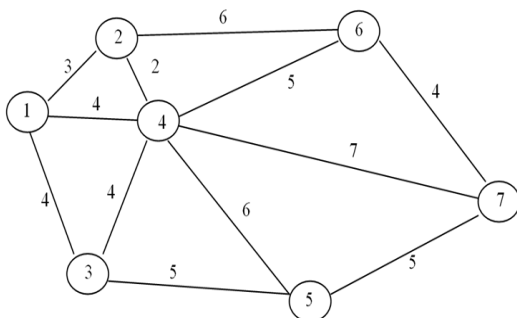


Рисунок 2.1 – Схема транспортной сети

Определим кратчайшие расстояния от пункта 1 до всех остальных пунктов сети.

Первая итерация.

Шаг 1. Вершине 1 присваиваем значение потенциала, равное 0. Укажем это значение возле вершины, заключив его в рамку (рисунок 2.2).

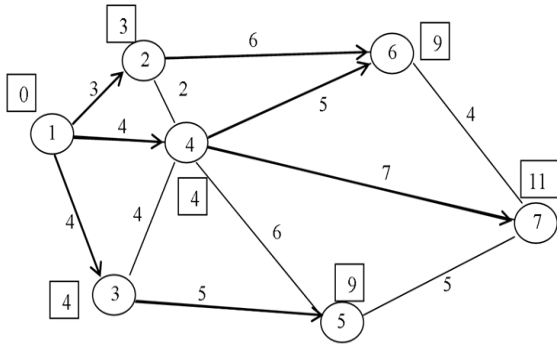


Рисунок 2.2 – Результаты вычисления кратчайших расстояний

Шаг 2. Рассматриваем все ребра сети, удовлетворяющие условию: потенциал одной из вершин этого ребра известен, потенциал второй – неизвестен. На этом этапе это ребра (1, 2), (1, 4), (1, 3). Вычисляем расстояния от первой вершины до вершин 2, 4, 3 и выбираем минимальное значение.

Шаг 3. Минимальное значение суммы (равное трем) соответствует ребру (1, 2). Поэтому потенциал вершины 2 присваивается равным трем, и ребро (1, 2) помечается стрелкой. Результаты выполненных расчетов приведены в таблице 2.1.

Вторая итерация.

Теперь известны потенциалы вершин 1 и 2. Рассмотрим все ребра сети, для которых потенциал одной из вершин – известен, а другой – неизвестен. На этом этапе это ребра (1, 4), (1, 3), (2, 6), (2, 4). Вычисляем суммарные расстояния (потенциал вершины плюс расстояние между пунктами) до пунктов 3, 4, 6 и выбираем минимальное значение. На этой итерации минимальное значение, равное четырем, достигается для двух разных пунктов: третьего и четвертого. Этим пунктам присваиваем значения потенциалов, равные четырем и ребра (1, 4) и (1, 3) помечаем стрелками.

Подобным образом выполняем расчеты на всех последующих шагах. Результаты выполнения расчетов приведены в таблице 2.1.

Используя метод потенциалов, определены кратчайшие расстояния от первого пункта до каждого из пунктов транспортной сети. Эти значения приведены на рисунке 2.2. Последовательность ребер, помеченных следующими друг за другом стрелками, указывает кратчайший путь от первого пункта до всех остальных пунктов этой сети.

Таблица 2.1 – Выполнение расчетов по определению кратчайших расстояний

Номер итерации	Вершины с известным потенциалом	Рассматриваемые ребра	Суммарные расстояния $U(i, j)$	Минимум $U(i, j)$	Назначаемый потенциал
I	1	(1, 2) (1, 4) (1, 3)	0+3=3 0+4=4 0+4=4	3	$v_2 = 3$
II	1 2	(1, 4) (1, 3) (2, 6) (2, 4)	0+4=4 0+4=4 3+6=9 3+2=5	4 4	$v_4 = 4$ $v_3 = 4$
III	2 4 3	(2, 6) (4, 6) (4, 7) (4, 5) (3, 5)	3+6=9 4+5=9 4+7=11 4+6=10 4+5=9	9 9 9	$v_6 = 9$ $v_5 = 9$
IV	6 4 5	(6, 7) (4, 7) (5, 7)	9+4=13 4+7=11 9+5=14	11	$v_7 = 11$

2.2 Транспортная задача линейного программирования в матричной форме

2.2.1 Постановка и математическая модель транспортной задачи

Транспортная задача ЛП формулируется следующим образом.

Пусть имеется m поставщиков A_1, A_2, \dots, A_m , располагающих некоторым однородным грузом в объемах a_1, a_2, \dots, a_m единиц, и n потребителей B_1, B_2, \dots, B_n с объемами потребления b_1, b_2, \dots, b_n единиц. Известна стоимость c_{ij} перевозки единицы груза от i -го ($i = \overline{1, m}$) поставщика j -му ($j = \overline{1, n}$) потребителю. Требуется составить план перевозок, который полностью удовлетворяет спрос потребителей в грузе, и при этом суммарные транспортные издержки будут минимизированы.

Для построения математической модели ТЗ рассмотрим матрицу

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix},$$

где x_{ij} – количество единиц груза, поставляемое от i -го поставщика j -му потребителю.

Матрица $X = \|x_{ij}\|$ называется *планом (матрицей) перевозок*.

Предполагается, что $x_{ij} \geq 0$.

Матрица $C = \|c_{ij}\|$ называется *матрицей тарифов* или *матрицей стоимости*.

Для наглядности модель транспортной задачи записывают в виде таблицы 2.2 – *матричной модели ТЗ*.

Таблица 2.2 – Матричная модель ТЗ

Поставщики	Потребители					Запасы груза a_i
	B_1	...	B_j	...	B_n	
A_1	c_{11} x_{11}	...	c_{1j} x_{1j}	...	c_{1n} x_{1n}	a_1
...
A_i	c_{i1} x_{i1}	...	c_{ij} x_{ij}	...	c_{in} x_{in}	a_i
...
A_m	c_{m1} x_{m1}	...	c_{mj} x_{mj}	...	c_{mn} x_{mn}	a_m
Потребность в грузе b_j	b_1	...	b_j	...	b_n	

Основное положение, используемое при построении математической модели транспортной задачи, состоит в том, что величина транспортных расходов на каждом маршруте прямо пропорциональна объему перевозимого груза.

Транспортная задача называется задачей *закрытого типа*, если суммарный запас грузов у поставщиков равен суммарному спросу потребителей, т. е. выполняется условие общего баланса:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (2.1)$$

Если условие (2.1) не выполняется, то задача называется задачей *открытого типа*.

Любую задачу открытого типа можно свести к задаче закрытого типа.

В случае, если суммарные запасы поставщиков больше потребностей получателей (т. е. $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$), вводят $(n + 1)$ -го фиктивного потребителя,

потребности которого равны излишку запаса, т. е.

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j. \quad (2.2)$$

Стоимости перевозки единицы груза от каждого поставщика к $(n+1)$ -му потребителю c_{in+1} принимаются равными нулю (поставки x_{in+1} в оптимальном плане будут представлять собой остатки продукции на складах поставщиков).

Если суммарные потребности в грузе превышают суммарные запасы (т. е. $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$), то вводят $(m+1)$ -го фиктивного поставщика, мощность которого считается равной недостающему грузу, т. е.

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i. \quad (2.3)$$

Стоимости перевозки единицы груза от $(m+1)$ -го поставщика j -му потребителю c_{m+1j} принимаются равными некоторому большому положительному числу M (поставки x_{m+1j} в оптимальном плане полученной задачи покажут объемы недопоставки груза).

Математическая модель закрытой транспортной задачи формулируется следующим образом.

Требуется найти план перевозок $X = \|x_{ij}\|$ ($i = \overline{1, m}; j = \overline{1, n}$), при котором суммарные затраты на перевозку груза будут минимальными:

$$z(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (2.4)$$

при ограничениях:

– на запасы груза у поставщиков, которые в силу (2.1) должны быть вывезены,

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m}; \quad (2.5)$$

– спрос потребителей, который должен быть полностью удовлетворен,

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n}, \quad (2.6)$$

условия неотрицательности, исключающие обратные перевозки,

$$x_{ij} \geq 0, \quad i = \overline{1, m}; \quad j = \overline{1, n}. \quad (2.7)$$

План перевозок X будем называть **допустимым планом**, если он удовлетворяет ограничениям (2.5)–(2.7). Допустимый план перевозок называется **оптимальным планом**, если он доставляет минимум целевой функции (2.4).

Таким образом, транспортная задача представляет собой задачу линейного программирования. Ее оптимальное решение всегда можно найти симплексным методом. Однако матрица системы ограничений (2.5)–(2.6) специфична. Специфика состоит в следующем:

- 1) коэффициенты при переменных x_{ij} во всех ограничениях равны единице;
- 2) каждая переменная x_{ij} встречается один раз в ограничениях (2.5) и один раз в ограничениях (2.6);
- 3) система ограничений (2.5)–(2.6) симметрична относительно переменных x_{ij} .

Благодаря этой специфике общую процедуру симплексного метода можно значительно упростить. Для решения транспортной задачи используются специальные алгоритмы, которые будут описаны ниже.

Решение транспортной задачи состоит из двух этапов:

- 1) построение начального базисного плана одним из известных методов (например, методом северо-западного угла, методом минимальной стоимости, методом двойного предпочтения и др.);
- 2) переход от начального базисного плана к оптимальному плану перевозок, например, с помощью метода потенциалов.

2.2.2 Построение начального базисного плана перевозок

Построение начального базисного плана, а также его преобразование будем производить непосредственно в таблице 2.2. Если в плане перевозок переменная x_{ik} равна некоторому числу $a \neq 0$, то это число записываем в левый нижний угол клетки $(i; k)$ и считаем ее *загруженной* (или *базисной*), если же $x_{ik} = 0$, то клетку $(i; k)$ оставляем *свободной* (*незагруженной*).

Рассмотрим методы построения начального плана перевозок: метод северо-западного угла, метод минимальной стоимости, метод двойного предпочтения, метод Фогеля построения начального базисного плана.

Метод северо-западного угла.

При использовании этого метода план перевозок начинаем составлять с верхней левой клетки (северо-западного угла), двигаясь затем от нее по строке вправо или по столбцу вниз. В клетку $(1; 1)$ заносим наименьшее из чисел a_1, b_1 , т. е. $x_{11} = \min(a_1, b_1)$. Если $a_1 > b_1$, то $x_{11} = b_1$, вычеркиваем из рассмотрения столбец $j=1$, так как спрос первого потребителя полностью удовлетворен. Если $a_1 < b_1$, то $x_{11} = a_1$, вычеркиваем строку $i=1$, так как в этом случае полностью исчерпаны запасы первого поставщика. Если $a_1 = b_1$,

то вычеркиваем и строку $i = 1$, и столбец $j = 1$ одновременно. Назначив перевозку $x_{11} = \min(a_1, b_1)$, уменьшаем запасы первого поставщика и потребности первого потребителя на величину x_{11} . С оставшейся матрицей поступаем аналогично предыдущему. Продолжая действовать по этой схеме, исчерпаем запасы поставщиков и удовлетворим запросы потребителей. На последнем шаге процесса получится матрица 1×1 , в которой одновременно вычеркиваем и строку, и столбец.

Метод северо-западного угла не учитывает матрицу тарифов, поэтому начальный план может оказаться далеким от оптимального. Более близким к оптимальному обычно является план, построенный методом минимальной стоимости или методом двойного предпочтения.

Метод минимальной стоимости.

Для построения начального плана этим методом в первую очередь назначаем перевозку в клетку с наименьшей стоимостью. Например, если минимальная стоимость перевозки c_{ij} , то в клетку (i, j) заносим наименьшее из чисел a_i, b_j , т. е. $x_{ij} = \min(a_i, b_j)$. Если запасы поставщика меньше спроса потребителя ($a_i < b_j$), то из рассмотрения исключаем строку, соответствующую поставщику, запасы которого полностью израсходованы. Если же запасы поставщика больше спроса потребителя ($a_i > b_j$), то из рассмотрения исключаем столбец, соответствующий потребителю, спрос которого полностью удовлетворен. Если запасы поставщика равны спросу потребителя ($a_i = b_j$), то вычеркиваем и строку, и столбец одновременно. Корректируем запасы поставщика и спрос потребителя на величину поставки. После этого из оставшихся клеток таблицы снова выбираем клетку с наименьшим тарифом и назначаем перевозку минимально возможного объема. Процесс распределения заканчивается, когда все запасы поставщиков исчерпаны, а спрос потребителей полностью удовлетворен.

Метод Фогеля.

Метод Фогеля дает начальный план, близкий к оптимальному. Его используют как приближенный метод решения транспортной задачи. Процедура построения начального опорного плана начинается с определения наибольшей разности между двумя наименьшими тарифами перевозки каждой строки и каждого столбца. Находим в ряду, соответствующем наибольшей разности всех строк и столбцов, клетку с минимальной стоимостью перевозки. Пусть это будет клетка (i_0, j_0) . Аналогично предыдущему методу записываем поставку $x_{i_0 j_0} = \min\{a_{i_0}, b_{j_0}\}$. Вычеркиваем ряд, соответствующий $\min\{a_{i_0}, b_{j_0}\}$. Если же $x_{i_0 j_0} = a_{i_0} = b_{j_0}$, вычеркиваем и строку $i = i_0$, и столбец $j = j_0$ одновременно. Корректируются запасы поставщика либо потребности получателя. С оставшейся матрицей поступаем аналогично предыдущему. На последнем шаге в матрице 1×1 одновременно убираем и строку, и столбец.

Метод двойного предпочтения.

Идея метода двойного предпочтения, используемого для построения начального плана перевозок состоит в следующем. Сначала просматриваются все строки матрицы тарифов и в каждой из них, например символом *, отмечается минимальный элемент. Затем просматривают все столбцы матрицы тарифов и так же символом * помечается минимальный элемент в каждом столбце. Назначение корреспонденций начинают с клеток таблицы, помеченных символами **, следуя правилу минимального элемента. Затем производят назначение перевозок в клетки таблицы, помеченные символом *, а затем – в клетки без пометок, на каждом этапе придерживаясь правила минимального элемента.

Построенный начальный план перевозок должен быть *базисным*, т. е. содержать $m + n - 1$ загруженных клеток. Если число назначенных перевозок меньше $m + n - 1$, то начальный план перевозок – *вырожденный*. Эта ситуация возникает, когда в процессе заполнения таблицы на одном шаге могут быть одновременно исключены и строка, и столбец, т. е. полностью исчерпаны запасы поставщика и полностью удовлетворен спрос потребителя.

Для преодоления вырожденности плана перевозок в свободную клетку необходимо ввести нулевую перевозку так, чтобы не образовывался цикл с ранее загруженными клетками. Введение этой перевозки не влияет на экономическую оценку решения, но оно необходимо для получения начального базисного плана и для последующего поиска оптимального решения.

2.2.3 Построение оптимального плана методом потенциалов

Полученный начальный базисный план перевозок путем постепенного его улучшения доводится до оптимального.

Условие оптимальности плана перевозок можно сформулировать следующим образом.

Если для некоторого плана $X^* = \|x_{ij}^*\| (i = \overline{1, m}; j = \overline{1, n})$ транспортной задачи существуют такие числа $u_i (i = \overline{1, m})$ и $v_j (j = \overline{1, n})$, что

$$v_j - u_i = c_{ij}, \quad \text{если } x_{ij} \geq 0, \quad (2.8)$$

$$v_j - u_i \leq c_{ij}, \quad \text{если } x_{ij} = 0, \quad (2.9)$$

то план $X^* = \|x_{ij}^*\|$ является *оптимальным*.

Числа u_i и v_j называются *потенциалами* соответственно поставщиков и потребителей.

Каждой загруженной клетке будет соответствовать уравнение $v_j - u_i = c_{ij}$. Так как число загруженных клеток равно $m + n - 1$, т. е. на единицу меньше числа потенциалов, то для определения чисел u_i и v_j необходимо решить систему из $n + m - 1$ уравнений $v_j - u_i = c_{ij}$ с $n + m$ неизвестными.

Поэтому одно из неизвестных можно положить равным произвольному значению (например, нулю).

Потенциалы остальных строк и столбцов вычисляются по стоимости перевозок в *загруженных* (базисных) клетках по следующим формулам:

$$\begin{cases} v_j = u_i + c_{ij}, & x_{ij} \geq 0, \\ u_i = v_j - c_{ij}, & x_{ij} \geq 0. \end{cases} \quad (2.10)$$

После вычисления потенциалов u_i и v_j для всех *незагруженных* (свободных) клеток вычисляется величина превышения стоимости по формуле

$$\Delta_{ij} = c_{ij} + u_i - v_j. \quad (2.11)$$

Если для всех незагруженных клеток $\Delta_{ij} \geq 0$, то построенный план перевозок является *оптимальным*, и, следовательно, процесс решения задачи завершен; в противном случае – переходим к построению улучшенного плана перевозок.

Клетки, в которых условие оптимальности не выполняется, называются *потенциальными*.

Для улучшения плана перевозок необходимо загрузить клетку, для которой значение $\Delta_{ij} < 0$ является наибольшим по абсолютной величине. В результате этого переменная x_{ij} будет введена в базис.

Пусть это клетка (i, j) . Для изменения плана перевозок, начиная с клетки (i, j) , строится *замкнутый контур* (*цикл*), перемещаясь в горизонтальном или вертикальном направлении с поворотами в загруженных клетках с таким расчетом, чтобы вернуться в исходную клетку. Если замкнутый контур образует самопересекающаяся ломаная линия, то точки ее самопересечения вершин не образуют.

На рисунке 2.3 приведены примеры построения замкнутых контуров. Загруженные клетки отмечены серым цветом; клетка, которую требуется загрузить – звездочкой, а замкнутый контур – пунктирной линией.

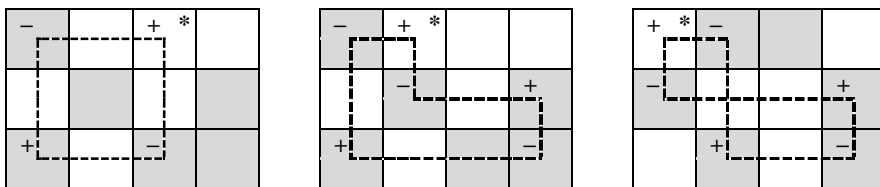


Рисунок 2.3 – Примеры форм замкнутых контуров

При правильном построении плана перевозок, начиная с любой незагруженной клетки можно построить лишь один замкнутый контур.

Вершины полученного контура поочередно помечаются знаками плюс «+» и минус «-», начиная с вводимой в базис клетки, которая всегда помечается знаком «+» (смотри рисунок 2.3).

Среди поставок клеток контура, помеченных знаком «-», выбирается наименьшее значение объема перевозок λ , и на величину λ изменяются объемы перевозок в клетках цикла: увеличиваются в вершинах контура, помеченных знаком «+», и уменьшаются в вершинах, помеченных знаком «-», в результате чего баланс цикла не нарушается. Клетка, по которой была выбрана величина λ , выводится из множества базисных, т. е. становится свободной клеткой.

Затем вновь проверяется выполнение условия оптимальности полученного плана перевозок. При необходимости повторяется процедура улучшения плана.

Сформулируем *алгоритм решения ТЗ* методом потенциалов.

1 Устанавливается тип модели транспортной задачи. Если задача является задачей закрытого типа, то перейти к шагу 2 алгоритма, в противном случае в соответствии формулами (2.2), (2.3) привести ее к задаче закрытого типа.

2 Строится начальный базисный план одним из известных методов (методом минимальной стоимости, методом северо-западного угла, методом двойного предпочтения и метод Фогеля).

3 Вычисляются потенциалы строк u_i и столбцов v_j по стоимости перевозок в загруженных клетках по формуле (2.10).

4 Для незагруженных клеток вычисляются величины превышения стоимости по формуле (2.11). Если все $\Delta_{ij} \geq 0$, то план перевозок является оптимальным, задача решена, $z^* = z(X^*)$. В противном случае, переходим к следующему шагу.

5 Начиная с клетки, для которой значение $\Delta_{ij} < 0$ является наибольшим по абсолютной величине, строится замкнутый контур. Перераспределяем груз по контуру и получаем новый улучшенный план перевозок. Переходим к шагу 3 алгоритма.

Альтернативные решения. Если для незагруженных клеток величина превышения стоимости $\Delta_{ij} = 0$, то полученный оптимальный план перевозок не является единственным, т. е. существуют *альтернативные оптимальные решения*.

Для получения нового оптимального решения транспортной задачи необходимо загрузить клетку с нулевой величиной превышения стоимости (т. е. сделать базисной), построив от нее замкнутый контур и перераспределив груз по контуру. При этом значение целевой функции не изменится, но будут получены новые оптимальные решения.

2.2.4 Пример решения транспортной задачи в матричной форме

Пример 2.2. Постановка задачи. Филиалы транспортно-производственной компании A_1, A_2 и A_3 производят однородную (или взаимозаменяемую) продукцию и доставляют ее потребителям B_1, B_2, B_3, B_4 . В трех пунктах производства сосредоточен груз в количествах соответственно 80, 100, 150 единиц. Имеющийся груз необходимо доставить потребителям, спрос которых выражается величинами 140, 40, 100, 50 единиц. Известна стоимость c_{ij} производства и перевозки единицы продукции из i -го ($i = \overline{1,3}$) пункта отправления в j -й пункт назначения ($j = \overline{1,4}$). Требуется составить план перевозок, который полностью удовлетворяет спрос потребителей в грузе, и при этом суммарные транспортные издержки минимизируются.

Таблица 2.3 – Исходные данные задачи

Поставщики	Потребители				Запасы поставщиков
	B_1	B_2	B_3	B_4	
A_1	12	9	8	6	$a_1 = 80$
A_2	15	11	9	10	$a_2 = 100$
A_3	10	13	7	14	$a_3 = 150$
Спрос потребителей	$b_1 = 140$	$b_2 = 40$	$b_3 = 100$	$b_4 = 50$	330
					330

Суммарные запасы поставщиков равны суммарному спросу потребителей, что составляет 330 ед. (т. е. выполняется условие общего баланса $\sum_{i=1}^3 a_i = \sum_{j=1}^4 b_j$). Следовательно, данная задача является задачей закрытого типа.

Математическую модель транспортной задачи сформулируем следующим образом.

Требуется найти план перевозок

$$X = \| \| x_{ij} \| \| = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{pmatrix},$$

при котором суммарные затраты на перевозку груза от поставщиков к потребителям будут минимальными:

$$z(X) = 12x_{11} + 9x_{12} + 8x_{13} + 6x_{14} + 15x_{21} + 11x_{22} + 9x_{23} + 10x_{24} + 10x_{31} + 13x_{32} + 7x_{33} + 14x_{34} \rightarrow \min$$

при ограничениях:

– на запасы поставщиков:

$$\begin{aligned}x_{11} + x_{12} + x_{13} + x_{14} &= 80; \\x_{21} + x_{22} + x_{23} + x_{24} &= 100; \\x_{31} + x_{32} + x_{33} + x_{34} &= 150;\end{aligned}$$

– спрос потребителей, который должен быть полностью удовлетворен:

$$\begin{aligned}x_{11} + x_{21} + x_{31} &= 140; & x_{12} + x_{22} + x_{32} &= 40; \\x_{13} + x_{23} + x_{33} &= 100; & x_{14} + x_{24} + x_{34} &= 50;\end{aligned}$$

условия неотрицательности, исключающие обратные перевозки,

$$x_{ij} \geq 0, \quad i = \overline{1,3}; \quad j = \overline{1,4}.$$

Построим начальные базисные планы методом северо-западного угла и методом минимальной стоимости и сравним значения целевой функции для полученных планов.

Для построения начального плана *методом северо-западного угла* (таблица 2.4) назначение перевозок начинаем с верхней левой клетки (северо-западного угла).

Таблица 2.4 – План перевозок, построенный методом северо-западного угла

Поставщики	Потребители				Запасы поставщиков
	B_1	B_2	B_3	B_4	
A_1	12 80	9	8	6 0	$a_1 = 80$
A_2	15 60	11 40	9	10	$a_2 = 100, 40$
A_3	10	13	7 100	14 50	$a_3 = 150, 50$
Спрос потребителей	$b_1 = 140,$ 60	$b_2 = 40$	$b_3 = 100$	$b_4 = 50$	

В клетку (1; 1) записываем наименьшее из значений a_1 и b_1 $x_{11} = \min(80, 140) = 80$. Так как запасы первого поставщика полностью исчерпаны, из дальнейшего рассмотрения исключаем первую строку. Вычеркнув строку, корректируем потребности первого потребителя на величину $x_{11} = 80$, $b_1 = 140 - 80 = 60$. Следующая поставка назначается от второго поставщика первому потребителю. В клетку (2; 1) назначаем перевозку $x_{21} = \min(100; 60) = 60$ и исключаем из дальнейшего рассмотрения первого потребителя. Вычеркнув первый столбец, корректируем запасы второго поставщика $a_2 = 100 - 60 = 40$.

Следующая поставка осуществляется от второго поставщика второму потребителю. В клетку (2; 2) назначаем перевозку $x_{22} = \min(40; 40) = 40$. Исключаем из дальнейшего рассмотрения второго поставщика и второго потребителя. С оставшейся матрицей поступаем аналогично предыдущему:

$$x_{33} = \min(150; 100) = 100; \quad x_{34} = \min(50; 50) = 50.$$

Построенный начальный план перевозок является вырожденным, так как число ненулевых перевозок x_{ij} не равно $m + n - 1 = 6$. Для того чтобы полученный план перевозок сделать базисным, необходимо в него добавить перевозку нулевого объема. Например, при назначении перевозки $x_{14} = 0$ в свободную клетку (1; 4) не образуется замкнутый контур из назначенных перевозок.

Вычислим значение целевой функции при начальном плане перевозок:

$$z(X^0) = 12 \cdot 80 + 6 \cdot 0 + 15 \cdot 60 + 11 \cdot 40 + 7 \cdot 100 + 14 \cdot 50 = 3700 \text{ ден. ед.}$$

Построим начальный план перевозок *методом минимальной стоимости* (таблица 2.5). Назначение перевозок начинаем с клетки (1; 4), имеющей минимальную стоимость перевозки. В клетку (1; 4) записываем наименьшее из значений a_1 и b_4 $x_{14} = \min(80, 50) = 50$ и исключаем из дальнейшего рассмотрения четвертый столбец. Вычеркнув четвертый столбец, корректируем запасы первого поставщика на величину $x_{14} = 50$, $a_1 = 80 - 50 = 30$.

В оставшейся таблице снова выбираем клетку с минимальной стоимостью перевозки. Это клетка (3; 3). От третьего поставщика к третьему потребителю назначаем перевозку $x_{33} = \min(150; 100) = 100$. Из дальнейшего рассмотрения исключаем третьего потребителя. Корректируем запасы третьего поставщика $a_3 = 150 - 100 = 50$. С оставшейся матрицей поступаем аналогично предыдущему:

$$\begin{aligned} x_{12} &= \min(30; 40) = 30; & x_{31} &= \min(50; 140) = 50; \\ x_{22} &= \min(100; 10) = 10; & x_{21} &= \min(90; 90) = 90. \end{aligned}$$

Таблица 2.5 – План перевозок, построенный методом минимальной стоимости

Поставщики	Потребители				Запасы поставщиков
	B_1	B_2	B_3	B_4	
A_1	12	9	8	6	$a_1 = 80, 30$
		30		50	
A_2	15	11	9	10	$a_2 = 100, 90$
	90	10			
A_3	10	13	7	14	$a_3 = 150, 50$
	50		100		
Спрос потребителей	$b_1 = 140, 90$	$b_2 = 40, 10$	$b_3 = 100$	$b_4 = 50$	

Построенный начальный план перевозок является базисным, так как число назначенных перевозок x_{ij} равно $m + n - 1 = 6$.

Вычислим значение целевой функции при начальном плане перевозок:

$$z(X^0) = 9 \cdot 30 + 6 \cdot 50 + 15 \cdot 90 + 11 \cdot 10 + 10 \cdot 50 + 7 \cdot 100 = 3230 \text{ ден. ед.}$$

Сравнивая значения целевых функций для планов, полученных методами северо-западного угла и минимальной стоимости, заметим, что транспортные расходы по плану, полученному методом минимальной стоимости, меньше на 470 ден. ед. Следовательно, план, построенный методом минимальной стоимости, ближе к оптимальному. Доведем этот план перевозок до оптимального, используя метод потенциалов.

Построение оптимального плана *методом потенциалов*.

Вычислим потенциалы строк u_i и столбцов v_j по стоимости перевозок в загруженных клетках. Если известен u_i , то $v_j = c_{ij} + u_i$, если известен v_j , то $u_i = v_j - c_{ij}$. Положим, например, $u_1 = 0$. Тогда будут вычислены и остальные потенциалы строк и столбцов (таблица 2.6).

Таблица 2.6 – Начальный план перевозок X^0

$u_i \backslash v_j$	$v_1 = 13$	$v_2 = 9$	$v_3 = 10$	$v_4 = 6$
$u_1 = 0$	12 30	9	8	6 50
$u_2 = -2$	90 -	15 10	11 +	9 10
$u_3 = 3$	50 +	10 -	13 -	7 14
			100	

Для незагруженных клеток вычислим величины превышения стоимости $\Delta_{ij} = c_{ij} + u_i - v_j$:

$$\Delta_{11} = 12 + 0 - 13 = -1;$$

$$\Delta_{24} = 10 - 2 - 6 = 2;$$

$$\Delta_{13} = 8 + 0 - 10 = -2;$$

$$\Delta_{32} = 13 + 3 - 9 = 7;$$

$$\Delta_{23} = 9 - 2 - 10 = -3;$$

$$\Delta_{34} = 14 + 3 - 6 = 11.$$

Поскольку среди оценок Δ_{ij} имеются отрицательные, то полученный план неоптимален. Выбираем клетку с наибольшим по абсолютной величине значением Δ_{ij} . Это клетка (2; 3). Размечаем вершины контура попеременно знаками плюс «+», минус «-», обходя замкнутый контур в любом направлении.

Среди клеток, помеченных знаком «-», выбираем наименьшее значение объема перевозки $\lambda = \min(90, 100) = 90$. Сформируем новый улучшенный план перевозок: на 90 единиц увеличим перевозки в клетках контура, помеченных знаком «+», и уменьшим в клетках, помеченных знаком «-».

Новый улучшенный план перевозок представлен в таблице 2.7.

Вычислим значение целевой функции при улучшенном плане перевозок:

$$z(X^1) = 9 \cdot 30 + 6 \cdot 50 + 11 \cdot 10 + 9 \cdot 90 + 10 \cdot 140 + 7 \cdot 10 = 2960 \text{ ден. ед.}$$

Таблица 2.7 – Улучшенный методом потенциалов план перевозок X^1

$u_i \backslash v_j$	$v_1=10$	$v_2=9$	$v_3=7$	$v_4=6$
$u_1=0$	12	9	8	6
	30		50	
$u_2=-2$	15	11	9	10
	10	90		
$u_3=0$	10	13	7	14
	140		10	

Вычислим потенциалы строк u_i и столбцов v_j по стоимости перевозок в загруженных клетках (таблице 2.7).

Для незагруженных клеток вычислим величины превышения стоимости:

$$\Delta_{11} = 12 + 0 - 10 = 2; \quad \Delta_{24} = 10 - 2 - 6 = 2;$$

$$\Delta_{13} = 8 + 0 - 7 = 1; \quad \Delta_{32} = 13 + 0 - 9 = 4;$$

$$\Delta_{21} = 15 - 2 - 10 = 3; \quad \Delta_{34} = 14 + 0 - 6 = 8.$$

Среди оценок свободных клеток нет отрицательных, следовательно, построенный план перевозок является оптимальным. А так как оценки свободных клеток не равны нулю, то полученный оптимальный план перевозок является единственным.

$$X^* = \begin{pmatrix} 0 & 30 & 0 & 50 \\ 0 & 10 & 90 & 0 \\ 140 & 0 & 10 & 0 \end{pmatrix}.$$

Значение целевой функции при оптимальном плане $z(X^*) = 2960$ ден. ед.

Для того чтобы общие затраты на перевозки были минимальными, рекомендуется придерживаться полученного оптимального плана X^* . В этом случае суммарные затраты будут минимальны и составят 2960 ден. ед.

2.2.5 Решение усложненных транспортных задач

На практике рассмотренная классическая транспортная задача встречается редко. Обычно при составлении математической модели задачи транспортного типа приходится вводить целый ряд дополнительных ограничений, вследствие чего поиск оптимального решения усложняется.

Рассмотрим наиболее часто встречающиеся случаи.

1 Запрет перевозок от i -го поставщика к j -му потребителю.

Часто необходимо вводить ограничения, согласно которым отдельные поставки от определенного поставщика A_i определенному потребителю B_j должны быть исключены (из-за возникновения технологических препят-

ствий, отсутствия достаточного количества транспорта или необходимых условий хранения груза, чрезмерной перегрузки коммуникаций и т. п.).

Чтобы определить оптимальный план в подобных задачах, предполагают, что стоимость перевозки единицы груза c_{ij} от i -го поставщика к j -му потребителю является достаточно большой величиной M , и при этом условии известными методами находят решение новой транспортной задачи.

С таким предположением исключается возможность перевозить груз от i -го поставщика к j -му потребителю при оптимальном плане транспортной задачи. Такой подход к нахождению решения транспортной задачи называют **запрещением перевозок** или **блокированием** соответствующей клетки таблицы.

Если же при решении транспортной задачи все-таки окажется, что запрещенная перевозка отлична от нуля при достаточно высокой стоимости перевозки груза в данной клетке ($x_{ij} \neq 0$), то это значит, что спрос j -го потребителя невозможно удовлетворить без i -го поставщика.

2 Фиксированная поставка.

В отдельных транспортных задачах дополнительным условием является обеспечение перевозки по соответствующим маршрутам определенного количества груза.

Пусть объем поставки груза от i -го поставщика к j -му потребителю должен быть строго определенным ($x_{ij} = d_{ij}$). Эту поставку следует включить в оптимальный план даже в том случае, если она не выгодна. Поэтому уменьшаются мощности i -го поставщика (a_i) и спрос j -го потребителя (b_j) на величину d_{ij} , а также вводится запрет на поставку груза по маршруту (i, j) и далее решается полученная задача.

В полученное оптимальное решение со значением функции z_{\min} подставляется фиксированная поставка $x_{ij} = d_{ij}$ и с учетом ее находится значение функции

$$z^* = z_{\min} + c_{ij} d_{ij}. \quad (2.12)$$

Пример 2.3

У трех поставщиков имеется в наличии $a_1 = 7$, $a_2 = 15$, $a_3 = 13$ единиц однородного груза. Данный груз должен быть доставлен четырем потребителям в количестве $b_1 = 5$, $b_2 = 10$, $b_3 = 6$, $b_4 = 14$ единиц. От третьего поставщика ко второму потребителю необходимо поставить $d_{32} = 5$ единиц груза. Известны также тарифы c_{ij} перевозок одной единицы груза от каждого поставщика каждому потребителю. Составить такой план перевозок груза, при котором общая стоимость перевозок будет минимальной.

Исходные данные приведены в таблице 2.8.

Данная задача является задачей закрытого типа. Суммарная мощность поставщиков равна суммарному спросу потребителей, что составляет 35 ед.

Таблица 2.8 – Исходные данные задачи

Поставщики	Потребители				Мощность поставщиков
	B_1	B_2	B_3	B_4	
A_1	1	9	8	8	$a_1 = 7$
A_2	6	5	2	10	$a_2 = 15$
A_3	7	8	4	3	$a_3 = 13$
Спрос потребителей	$b_1 = 5$	$b_2 = 10$	$b_3 = 6$	$b_4 = 14$	35 / 35

Поскольку объем поставки груза от третьего поставщика ко второму потребителю должен быть строго определенным $x_{32} = d_{32} = 5$, уменьшим мощность третьего поставщика и спрос второго потребителя на величину $d_{32} = 5$: $a_3 = 13 - 5 = 8$, $b_2 = 10 - 5 = 5$. Введем запрет на поставку груза по маршруту (3, 2). Стоимость от третьего поставщика ко второму потребителю примем достаточно большой величиной M (таблица 2.9).

Таблица 2.9 – Данные задачи

Поставщики	Потребители				Мощность поставщиков
	B_1	B_2	B_3	B_4	
A_1	1	9	8	8	$a_1 = 7$
A_2	6	5	2	10	$a_2 = 15$
A_3	7	M	4	3	$a_3 = 8$
Спрос потребителей	$b_1 = 5$	$b_2 = 5$	$b_3 = 6$	$b_4 = 14$	

Построение начального базисного плана методом северо-западного угла (таблица 2.10).

Таблица 2.10 – План перевозок, построенный методом северо-западного угла

Поставщики	Потребители				Мощность поставщиков
	B_1	B_2	B_3	B_4	
A_1	1 5	9 2	8	8	$a_1 = 7, 2$
A_2	6	5 3	2 6	10 6	$a_2 = 15, 12, 6$
A_3	7	M	4	3 8	$a_3 = 8$
Спрос потребителей	$b_1 = 5$	$b_2 = 5, 3$	$b_3 = 6$	$b_4 = 14, 8$	

Построенный начальный план перевозок является базисным, так как число назначенных перевозок x_{ij} равно $m+n-1=6$.

Значение целевой функции при начальном плане перевозок

$$z(X^0) = 1 \cdot 5 + 9 \cdot 2 + 5 \cdot 3 + 2 \cdot 6 + 10 \cdot 6 + 3 \cdot 8 = 134 \text{ ден. ед.}$$

Построение оптимального плана методом потенциалов.

Вычислим потенциалы строк и столбцов по стоимости перевозок в загруженных клетках (таблица 2.11).

Таблица 2.11 – Начальный план перевозок X^0

v_j	$v_1 = -3$	$v_2 = 5$	$v_3 = 2$	$v_4 = 10$
u_i				
$u_1 = -4$	5	1	2	8
$u_2 = 0$		6	3	6
$u_3 = 7$		7	M	8

Для незагруженных клеток вычислим величины превышения стоимости $\Delta_{ij} = v_j - u_i - c_{ij}$:

$$\Delta_{13} = 2 - (-4) - 8 = -2;$$

$$\Delta_{31} = -3 - 7 - 7 = -17;$$

$$\Delta_{14} = 10 - (-4) - 8 = 6;$$

$$\Delta_{32} = 5 - 7 - M = -2 - M;$$

$$\Delta_{21} = -3 - 0 - 6 = -9;$$

$$\Delta_{33} = 2 - 7 - 4 = -9.$$

Полученный план неоптимален. Среди оценок Δ_{ij} имеется положительная. Потенциальной является клетка (1; 4). От клетки (1; 4) строим замкнутый контур: (1; 4), (1; 2), (2; 2), (2; 4). Начиная с клетки (1; 4), разметим вершины контура попеременно знаками плюс «+», минус «-», обходя замкнутый контур в любом направлении. Из клеток, помеченных знаком «-», выбираем наименьшее значение объема перевозки $\theta = \min(2, 6) = 2$. Сформируем новый улучшенный план перевозок: на два увеличим перевозки в клетках, помеченных знаком «+», и уменьшим в клетках, помеченных знаком «-».

Новый улучшенный план перевозок представлен в таблице 2.12.

Таблица 2.12 – Улучшенный методом потенциалов план перевозок X^1

v_j	$v_1 = 3$	$v_2 = 5$	$v_3 = 2$	$v_4 = 10$
u_i				
$u_1 = 2$	5	1	2	8
$u_2 = 0$		6	5	4
$u_3 = 7$		7	M	8

Значение целевой функции при улучшенном плане перевозок:

$$z(X^1) = 1 \cdot 5 + 8 \cdot 2 + 5 \cdot 5 + 2 \cdot 6 + 10 \cdot 4 + 3 \cdot 8 = 122 \text{ ден. ед.}$$

Вычислим потенциалы строк и столбцов по стоимости перевозок в загруженных клетках.

Для незагруженных клеток вычислим величины превышения стоимости:

$$\begin{aligned} \Delta_{12} &= 5 - 9 - 2 = -6; & \Delta_{31} &= 3 - 7 - 7 = -14; \\ \Delta_{13} &= 2 - 8 - 2 = -8; & \Delta_{32} &= 5 - 7 - M = -2 - M; \\ \Delta_{21} &= 3 - 0 - 6 = -3; & \Delta_{33} &= 2 - 7 - 4 = -9. \end{aligned}$$

Среди оценок свободных клеток нет положительных, следовательно, построенный план перевозок является оптимальным.

В полученный оптимальный план подставим фиксированную поставку $x_{32} = d_{32} = 5$ (таблица 2.13).

Таблица 2.13 – Оптимальный план перевозок X^*

Поставщики	Потребители				Мощность поставщиков
	B_1	B_2	B_3	B_4	
A_1	1 5	9	8	8 2	$a_1 = 7$
A_2	6	5 5	2 6	10 4	$a_2 = 15$
A_3	7	8 5	4	3 8	$a_3 = 13$
Спрос потребителей	$b_1 = 5$	$b_2 = 10$	$b_3 = 6$	$b_4 = 14$	35 35

Оптимальный план перевозок с учетом того, что от третьего поставщика ко второму потребителю необходимо поставить пять единиц груза:

$$X^* = \begin{pmatrix} 5 & 0 & 0 & 2 \\ 0 & 5 & 6 & 4 \\ 0 & 5 & 0 & 8 \end{pmatrix}.$$

С учетом фиксированной поставки находим значение целевой функции оптимального плана при улучшенном плане перевозок:

$$z(X^*) = z(X^1) + c_{32} d_{32} = 122 + 8 \cdot 5 = 162 \text{ ден. ед.}$$

3 Нижние границы поставки. Если от i -го поставщика к j -му потребителю необходимо перевезти не менее d_{ij} единиц груза ($x_{ij} \geq d_{ij}$), то как и при фиксированной поставке, мощность соответствующего поставщика и спрос потребителя уменьшаются на величину d_{ij} . Однако в этом случае запрет на поставку груза от i -го поставщика к j -му потребителю не вводится.

4 Верхние границы поставки. Если в транспортной задаче объем поставки груза от i -го поставщика к j -му потребителю не должен превышать d_{ij} единиц ($x_{ij} \leq d_{ij}$), то при решении j -й столбец матрицы перевозок разбивается на два: j' и j'' .

Аналогично спрос j -го потребителя разбивается на две части: $b_{j'} = d$ и $b_{j''} = b_j - d$. Тарифы в обоих столбцах j' и j'' одинаковые, за исключением $c_{ij''} = M$, где M является сколь угодно большой величиной. Большая стоимость перевозки $c_{ij''} = M$ вводится для запрета перевозки от i -го поставщика к j'' -му потребителю.

Измененная задача позволит получить решение, удовлетворяющее поставленному условию.

Пример 2.4

У трех поставщиков имеется в наличии $a_1 = 30$, $a_2 = 26$, $a_3 = 34$ единиц однородного груза. Данный груз должен быть доставлен трем потребителям в количестве $b_1 = 40$, $b_2 = 16$, $b_3 = 6$, $b_3 = 34$ единиц. Объем поставки груза от второго поставщика ко второму потребителю не должен превышать 10 ед. груза ($d_{22} \leq 10$). Известны также тарифы c_{ij} перевозок одной единицы груза от каждого поставщика каждому потребителю. Составить такой план перевозок груза, при котором общая стоимость перевозок является минимальной.

Исходные данные приведены в таблице 2.14.

Таблица 2.14 – Исходные данные задачи

Поставщики	Потребители			Мощность поставщиков
	1	2	3	
1	5	8	14	$a_1 = 30$
2	11	9	7	$a_2 = 26$
3	7	8	6	$a_3 = 34$
Спрос потребителей	$b_1 = 40$	$b_2 = 16$	$b_3 = 34$	90 / 90

Данная задача является задачей закрытого типа. Суммарная мощность поставщиков равна суммарному спросу потребителей, что составляет 90 ед.

Поскольку объем поставки груза от второго поставщика ко второму потребителю не должен превышать 10 ед. ($d_{22} \leq 10$), то при решении второй столбец матрицы перевозок разбивается на два: $2'$ и $2''$.

Аналогично спрос второго потребителя разбивается на две части:

$$b_{2'} = d_{22} = 10; \quad b_{2''} = b_2 - d_{22} = 16 - 10 = 6.$$

Тарифы в обоих столбцах 2' и 2'' одинаковые, за исключением $c_{22''} = M$, где M является сколь угодно большой величиной. Большая стоимость перевозки $c_{22''} = M$ вводится для запрета перевозки от 2-го поставщика ко 2''-му потребителю.

Измененная задача, представленная в таблице 2.15, позволит получить решение, удовлетворяющее поставленному условию.

Таблица 2.15 – Данные задачи

Поставщики	Потребители				Мощность поставщиков
	1	2'	2''	3	
1	5	8	8	14	$a_1 = 30$
2	11	9	M	7	$a_2 = 26$
3	7	8	8	6	$a_3 = 34$
Спрос потребителей	$b_1 = 40$	$b_{2'} = 10$	$b_{2''} = 6$	$b_3 = 34$	

Построение начального базисного плана методом минимальной стоимости (таблица 2.16).

Таблица 2.16 – План перевозок, построенный методом минимальной стоимости

Поставщики	Потребители				Мощность поставщиков
	1	2'	2''	3	
1	5 30	8	8	14	$a_1 = 30$
2	11 10	9 10	M 6	7 0	$a_2 = 26, 16, 6$
3	7	8	8	6 34	$a_3 = 34$
Спрос потребителей	$b_1 = 40,$ 10	$b_{2'} = 10$	$b_{2''} = 6$	$b_3 = 34$	

Построенный начальный план перевозок является вырожденным, так как число ненулевых перевозок x_{ij} не равно $m + n - 1 = 6$. Для построения базисного плана в клетку (3; 2) введем нулевую перевозку $x_{32} = 0$ так, чтобы не образовывался замкнутый контур из назначенных перевозок.

Значение целевой функции при начальном плане перевозок

$$z(X^0) = 5 \cdot 30 + 11 \cdot 10 + 9 \cdot 10 + M \cdot 6 + 7 \cdot 0 + 6 \cdot 34 = 554 + 6M \text{ ден. ед.}$$

Построение оптимального плана методом потенциалов.

Вычислим потенциалы строк и столбцов по стоимости перевозок в загруженных клетках (таблица 2.17).

Таблица 2.17 – Начальный план перевозок X^0

$u_i \backslash v_j$	$v_1 = 11$	$v_2' = 9$	$v_{22}'' = M$	$v_3 = 7$
$u_1 = 6$	30	5	8	14
$u_2 = 0$	10	11	9	7
$u_3 = 1$	7	8	8	6
			6	0
			+	-
			8	34

Для незагруженных клеток вычислим величины превышения стоимости $\Delta_{ij} = v_j - u_i - c_{ij}$:

$$\begin{aligned} \Delta_{12}' &= 9 - 8 - 6 = -5; & \Delta_{31} &= 11 - 7 - 1 = 3; \\ \Delta_{12}'' &= M - 8 - 6 = M - 14; & \Delta_{32}' &= 9 - 8 - 1 = 0; \\ \Delta_{13} &= 7 - 14 - 6 = -13; & \Delta_{32}'' &= M - 8 - 1 = M - 9. \end{aligned}$$

Полученный план неоптимален. Среди оценок Δ_{ij} имеются положительные. Наиболее потенциальной является клетка (3; 2''). От клетки (3; 2'') строим замкнутый контур: (3; 2''), (3; 3), (2; 3), (2; 2''). Начиная с клетки (3; 2''), разметим вершины контура попеременно знаками плюс «+», минус «-», обходя замкнутый контур в любом направлении.

Из клеток, помеченных знаком «-», выбираем наименьшее значение объема перевозки $\theta = \min(6, 34) = 6$. Сформируем новый улучшенный план перевозок: на 6 увеличим перевозки в клетках, помеченных знаком «+», и уменьшим в клетках, помеченных знаком «-».

Новый улучшенный план перевозок представлен в таблице 2.18.

Таблица 2.18 – Улучшенный методом потенциалов план перевозок X^1

$u_i \backslash v_j$	$v_1 = 11$	$v_2' = 9$	$v_{22}'' = 9$	$v_3 = 7$
$u_1 = 6$	30	5	8	14
$u_2 = 0$	10	11	9	7
$u_3 = 1$	7	8	8	6
			6	0
			+	-
			7	28

Значение целевой функции при улучшенном плане перевозок

$$z(X^1) = 5 \cdot 30 + 11 \cdot 10 + 9 \cdot 10 + 7 \cdot 6 + 8 \cdot 6 + 6 \cdot 28 = 608 \text{ ден. ед.}$$

Вычислим потенциалы строк и столбцов по стоимости перевозок в загруженных клетках.

Для незагруженных клеток вычислим величины превышения стоимости:

$$\begin{aligned} \Delta_{12}' &= 9 - 8 - 6 = -5; & \Delta_{22}'' &= 9 - 0 - M = 9 - M; \\ \Delta_{12}'' &= 9 - 8 - 6 = -5; & \Delta_{31} &= 11 - 7 - 1 = 3; \\ \Delta_{13} &= 7 - 14 - 6 = -13; & \Delta_{32}' &= 9 - 8 - 1 = 0. \end{aligned}$$

Полученный план неоптимален. Среди оценок Δ_{ij} имеется положительная. Потенциальной является клетка (3; 1). От клетки (3; 1) строим замкнутый контур: (3; 1), (3; 3), (2; 3), (2; 1). Начиная с клетки (3; 3), разметим вершины контура попеременно знаками плюс «+», минус «-», обходя замкнутый контур в любом направлении.

Из клеток, помеченных знаком «-», выбираем наименьшее значение объема перевозки $\theta = \min(10, 28) = 10$. Сформируем новый улучшенный план перевозок: на 6 увеличим перевозки в клетках, помеченных знаком «+», и уменьшим в клетках, помеченных знаком «-».

Новый улучшенный план перевозок представлен в таблице 2.19.

Таблица 2.19 – Улучшенный методом потенциалов план перевозок X^2

$u_i \backslash v_j$	$v_1 = 8$	$v_2' = 9$	$v_{22}'' = 9$	$v_3 = 7$
$u_1 = 3$	30	5	8	8
$u_2 = 0$		11	9	M
$u_3 = 1$	10	7	8	8
		10	6	16
				18
				6

Значение целевой функции при улучшенном плане перевозок

$$z(X^2) = 5 \cdot 30 + 9 \cdot 10 + 7 \cdot 16 + 7 \cdot 10 + 8 \cdot 6 + 6 \cdot 18 = 578 \text{ ден. ед.}$$

Вычислим потенциалы строк и столбцов по стоимости перевозок в загруженных клетках.

Для незагруженных клеток вычислим величины превышения стоимости:

$$\begin{aligned} \Delta_{12}' &= 9 - 8 - 3 = -2; & \Delta_{21} &= 8 - 0 - 11 = -3; \\ \Delta_{12}'' &= 9 - 8 - 3 = -2; & \Delta_{22}'' &= 9 - 0 - M = 9 - M; \\ \Delta_{13} &= 7 - 14 - 3 = -10; & \Delta_{32}' &= 9 - 8 - 1 = 0. \end{aligned}$$

Среди оценок свободных клеток нет положительных, следовательно, построенный план перевозок является оптимальным.

Объединим столбцы 2' и 2''. Получим оптимальный план исходной задачи (таблица 2.20), в котором объем поставки груза от второго поставщика ко второму потребителю не превышает 10 ед. груза.

Таблица 2.20 – Оптимальный план перевозок X^*

Поставщики	Потребители			Мощность поставщиков		
	1	2	3			
1	30	5	8	14	$a_1 = 30$	
2		11	10	9	7	$a_2 = 26$
3	10	7	6	8	6	$a_3 = 34$
Спрос потребителей	$b_1 = 40$	$b_2 = 16$	$b_3 = 34$	90		

Оптимальный план перевозок с учетом того, что от второго поставщика ко второму потребителю объем поставки не превышает 10 единиц груза:

$$X^* = \begin{pmatrix} 30 & 0 & 0 \\ 0 & 10 & 16 \\ 10 & 6 & 18 \end{pmatrix}.$$

Значение целевой функции при оптимальном плане $z(X^*) = 578$ ден. ед.

5 Максимизация функции в моделях транспортного типа. Многие задачи (например, наиболее эффективная организация работ по сооружению земляного полотна железной дороги, оптимальное закрепление за станками операций по обработке деталей) сводятся к решению транспортной задачи. Как правило, в этих случаях необходимо находить максимум целевой функции.

Для решения задачи транспортного типа на максимум возможны два способа нахождения оптимального плана:

1 Необходимо коэффициенты c_{ij} при неизвестных в целевой функции взять со знаком минус, т. е. необходимо перейти к нахождению минимума измененной функции.

2 При составлении начального плана в первую очередь стараются заполнить клетки с наиболее высокими значениями показателей c_{ij} . Выбор клетки, подлежащей заполнению при переходе от одного допустимого плана к другому, должен производиться не по наибольшей величине превышения стоимости $\Delta_{ij} > 0$, а по минимальной отрицательной величине $\Delta_{ij} = v_j - u_i - c_{ij}$. Оптимальным будет план, которому в последней таблице сопутствуют свободные клетки с неотрицательными оценками (т. е. все $\Delta_{ij} \geq 0$).

2.3 Транспортная задача линейного программирования в сетевой форме

2.3.1 Постановка транспортной задачи в сетевой форме

Помимо матричной формы транспортная задача может быть представлена в сетевой форме. Для этого должны быть заданы транспортная сеть (транспортные пункты и участки дорог, их соединяющие), количественное распределение отправления и прибытия однородного груза по транспортным пунктам; значения показателей оптимальности (тарифы, расстояния и т. п.) для каждого участка сети. Транспортным пунктам соответствуют вершины сети, а участкам, которые их соединяют, ребра.

В общем виде *транспортная задача в сетевой форме* формулируется следующим образом.

Пусть задана транспортная сеть с n вершинами, которые моделируют поставщиков и потребителей, и m ребрами, которые моделируют связывающие их дороги. Для каждого ребра $d = (i, j)$ задана стоимость $c_d = c_{ij}$ перевозки единицы груза. Известны мощности поставщиков $a_i > 0$ и спрос потребителей $a_i < 0$. Требуется составить такой план перевозок $X = \|x_d\|$, который полностью удовлетворит спрос потребителей в грузе, и при этом суммарные транспортные расходы будут минимальными.

Запасы груза у поставщиков будем записывать положительными, а потребности – отрицательными числами. На сети могут быть изображены вершины, в которых нет ни поставщиков, ни потребителей. Наличие таких вершин не влияет на способ решения, если считать, что запасы (потребности) груза в них равны нулю ($a_i = 0$). Такие вершины называют нейтральными или промежуточными вершинами.

Транспортная задача в сетевой форме называется задачей *закрытого типа*, если выполняется условие

$$\sum_{i \in I} a_i = 0, \quad (2.13)$$

где I – множество вершин сети.

Если условие (2.13) не выполняется (т. е. имеет место задача *открытого типа*), то выполняются действия, аналогичные описанным в пункте 2.2.1, – добавляется фиктивная вершина i^* с мощностью

$$a_{i^*} = -\sum_{i \in I} a_i. \quad (2.14)$$

Стоимость перевозок единицы груза между фиктивной вершиной и всеми остальными вершинами принимается равной нулю.

Математическая модель закрытой транспортной задачи в сетевой форме формулируется следующим образом.

Требуется построить план перевозок $X = \|x_d\|$ в некоторой сети, для которого суммарные затраты на перевозку груза будут минимальными:

$$z(X) = \sum_{d \in D} c_d x_d \rightarrow \min, \quad (2.15)$$

где $x_d = x_{ij}$ – величина поставки груза по ребру $d = (i, j)$;

D – множество ребер сети;

при ограничениях – для любой вершины сети разность выходящего и входящего потоков грузов равна ее мощности:

$$-\sum_{d \in D_i^+} x_d + \sum_{d \in D_i^-} x_d = a_i, \quad i \in I, \quad (2.16)$$

где D_i^+ – множество ребер, входящих в вершину i ,

D_i^- – множество ребер, исходящих из вершины i ;

условия неотрицательности, исключающие обратные перевозки:

$$x_d \geq 0, \quad d \in D. \quad (2.17)$$

Транспортную задачу в матричной форме можно представить в сетевом виде, приняв мощности вершин равными запасам поставщиков со знаком «+» и спросом потребителей со знаком «-».

Все вершины разделятся на два непересекающихся подмножества («поставщики» и «потребители»), ребра будут вести только из вершин первого подмножества в вершины второго: от поставщиков к потребителям (рисунок 2.4).

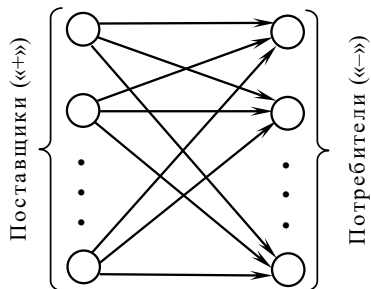


Рисунок 2.4 – Пример сети, представляющей матричную транспортную задачу в сетевой форме

2.3.2 Решение транспортной задачи в сетевой форме методом потенциалов

Решение транспортной задачи в сетевой форме состоит из двух этапов: построение начального базисного плана перевозок и доведение его до оптимального с помощью метода потенциалов.

Построение начального базисного плана, а также его преобразование будем производить непосредственно на транспортной сети. Поставки груза из вершины в вершину обозначаются стрелками (нагруженными ребрами) с указанием объемов поставок. План строится по возможным маршрутам перевозок.

Для транспортной задачи в сетевой форме начальный базисный план строится таким образом, чтобы он удовлетворял следующим требованиям:

1) все запасы грузов должны быть распределены, а потребности удовлетворены;

2) в каждую вершину сети должна входить или выходить из нее хотя бы одна стрелка;

3) общее количество стрелок должно быть на единицу меньше числа вершин;

4) стрелки не должны образовывать замкнутый контур.

Базисный план перевозок однородного груза на сети всегда содержит общее количество нагруженных ребер на единицу меньше числа вершин. В противном случае построенный план будет *вырожденным*. Для преодоления вырожденности вводят необходимое до $(n - 1)$ количества стрелок с нулевыми поставками с условием, что они не будут образовывать замкнутый контур с другими стрелками сети. Направление вводимых стрелок берется произвольным.

Рассмотрим *метод потенциалов* для транспортной задачи в сетевой форме.

Идея метода состоит в том, что полученный начальный базисный план перевозок путем постепенного его улучшения доводится до оптимального.

Условие оптимальности плана перевозок можно сформулировать следующим образом. Для того чтобы допустимый базисный план был оптимальным, необходимо и достаточно, чтобы для каждой вершины существовало число v_i , называемое потенциалом, такое, что для всех ребер $d = (i, j)$:

$$v_j - v_i = c_{ij}, \text{ если } x_{ij} > 0; \quad (2.18)$$

$$v_j - v_i \leq c_{ij}, \text{ если } x_{ij} = 0. \quad (2.19)$$

Для имеющегося плана перевозок строится система потенциалов вершин сети. Для этого выбирается произвольная вершина i , потенциал которой полагается равным произвольному значению (например, нулю). Затем, двигаясь по нагруженным ребрам, определяются потенциалы остальных вершин, руководствуясь правилом:

а) если стрелка направлена из вершины i в вершину j (рисунок 2.5, а), то к потенциалу этой вершины прибавляется величина c_{ij} , являющаяся стоимостью перевозки единицы груза из вершины i в вершину j :

$$v_j = v_i + c_{ij}; \quad (2.20)$$

б) если стрелка направлена к вершине i (рисунок 2.5, б), то величина c_{ij} вычитается из потенциала вершины i :

$$v_j = v_i - c_{ij}. \quad (2.21)$$

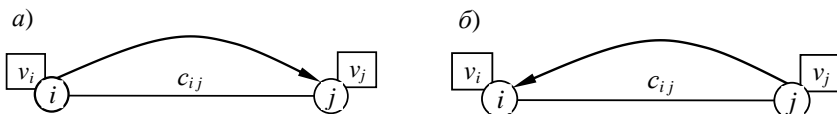


Рисунок 2.5 – Определение потенциалов вершин

Для проверки условия оптимальности необходимо определить оценки ненагруженных ребер (т. е. ребер без стрелок). Для этого из большего потенциала вершин рассматриваемого ребра вычитаем меньший, а разность вычитаем из показателя c_{ij} , соответствующего данному ребру:

$$\Delta_{ij} = c_{ij} - [\max(v_i, v_j) - \min(v_i, v_j)]. \quad (2.22)$$

где v_i, v_j – потенциалы соответственно вершин i и j ребра (i, j) .

Если для всех ненагруженных ребер

$$\Delta_{ij} \geq 0, \quad (2.23)$$

то построенный план является *оптимальным*, и, следовательно, процесс решения задачи завершен; в противном случае – переходим к построению улучшенного плана перевозок.

Для этого среди ненагруженных ребер, имеющих отрицательные оценки $\Delta_{ij} < 0$, необходимо выбрать ребро с наибольшей по абсолютной величине оценкой и к этому ребру подрисовать новую стрелку, направленную от вершины с меньшим потенциалом к вершине с большим потенциалом. В результате будет получен замкнутый контур из стрелок. Построенный замкнутый контур является аналогом цикла в методе потенциалов для транспортной задачи в матричной форме.

Для определения объема поставки вновь «загружаемого» ребра рассматриваются все стрелки образованного контура, имеющие направление, противоположное направлению добавленной стрелки, и среди них находится стрелка с наименьшей величиной поставки λ .

Изменение объемов перевозок по ребрам полученного замкнутого контура производится следующим образом. На стрелках, имеющих то же направление, что и новая стрелка, объем перевозок увеличивается на выбранную величину λ . На стрелках, имеющих противоположное направление, объем перевозок уменьшается на выбранную величину λ . Объемы поставок, не входящих в контур, сохраняются неизменными. Стрелка, по которой была выбрана величина λ , ликвидируется. Общее число стрелок сети остается прежним: на единицу меньше числа вершин.

Затем вновь проверяем условие оптимальности полученного плана перевозок. При необходимости повторяем процедуру улучшения плана.

Сформулируем **алгоритм решения ТЗ в сетевой форме** методом потенциалов.

1 Устанавливаем тип модели транспортной задачи. Если задача является задачей закрытого типа, то перейти к пункту 2 алгоритма, в противном случае привести ее к задаче закрытого типа.

2 Строим начальный базисный план, который должен удовлетворять условиям 1–4 (с. 52).

3 Вычисляем потенциалы вершин по формулам (2.20), (2.21).

4 Проверяем выполнение условий оптимальности. Для этого по формуле (2.20) определяем оценки ненагруженных ребер. Если для всех ненагруженных ребер $\Delta_{ij} \geq 0$, то план является оптимальным, и, следовательно, процесс решения задачи завершен; $z^* = z(X^*)$. В противном случае – переходим к следующему этапу.

5 К ребру с наибольшей по абсолютной величине оценкой Δ_{ij} подрисовываем новую стрелку, в результате чего будет получен замкнутый контур. Перераспределяем груз по контуру и получаем новый улучшенный план перевозок. Переходим к пункту 3 алгоритма.

Альтернативные решения. Если оценки ненагруженных ребер $\Delta_{ij} = 0$, то полученный оптимальный план перевозок не является единственным. Для получения альтернативного оптимального решения транспортной задачи в сетевой форме необходимо к ребру с нулевой оценкой подрисовать новую стрелку и перераспределить груз по полученному замкнутому контуру.

2.3.3 Пример решения транспортной задачи в сетевой форме

Пример 2.5. Постановка задачи. Задана сеть автомобильных дорог; количество однородного груза у поставщиков $a_i > 0$ и потребителей $a_i < 0$ по транспортным пунктам сети; значения критерия оптимальности – тарифы c_{ij} на перевозку единицы груза по всем участкам сети.

Требуется построить план перевозок однородного груза, при котором суммарные затраты на перевозку груза будут минимальными.

Условие задачи схематически представлено на рисунке 2.6.

Для каждой вершины над чертой указан ее номер, под чертой – запасы груза у поставщиков (со знаком «+») и спрос потребителей (со знаком «-»). Около каждого ребра указаны тарифы на перевозку единицы груза между пунктами сети.

Поставки груза из вершины в вершину будем обозначать стрелками с указанием объемов поставок груза.

Установим тип модели транспортной задачи.

$$\sum_{i=1}^7 a_i = -60 + 0 + 100 - 100 + 80 + 40 - 60 = 0.$$

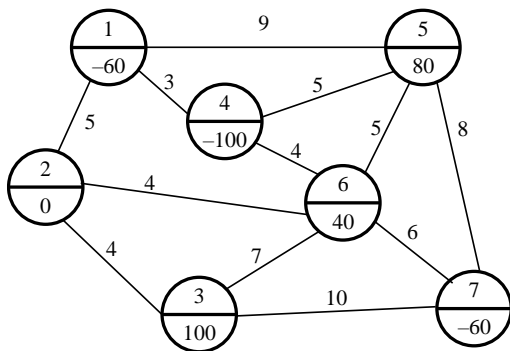


Рисунок 2.6 – Условие транспортной задачи

Это означает, что данная задача является задачей закрытого типа.

Построим начальный план перевозок произвольным образом. Возможный план приведен на рисунке 2.7.

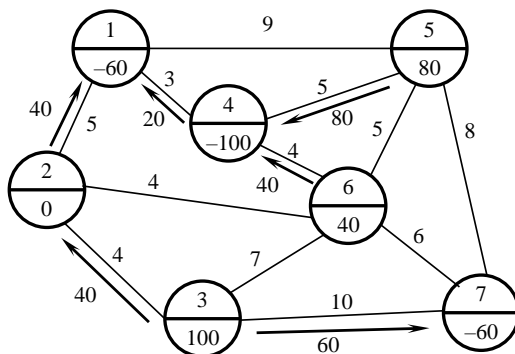


Рисунок 2.7 – Начальный план перевозок

Построенный план перевозок удовлетворяет условиям 1–4 (с. 52) и, следовательно, является базисным.

Значение целевой функции для начального плана перевозок составит

$$z(X^0) = 5 \cdot 40 + 4 \cdot 40 + 3 \cdot 20 + 10 \cdot 60 + 4 \cdot 40 + 5 \cdot 80 = 1580 \text{ ден. ед.}$$

Вычислим потенциалы всех вершин (рисунок 2.8).

Для этого одной из вершин присваиваем произвольное значение потенциала, например $v_3 = 0$. Значение потенциалов укажем около каждой вершины, и для большей наглядности потенциалы будем заключать в рамки. Затем, двигаясь по нагруженным ребрам, определяем потенциалы остальных вершин.

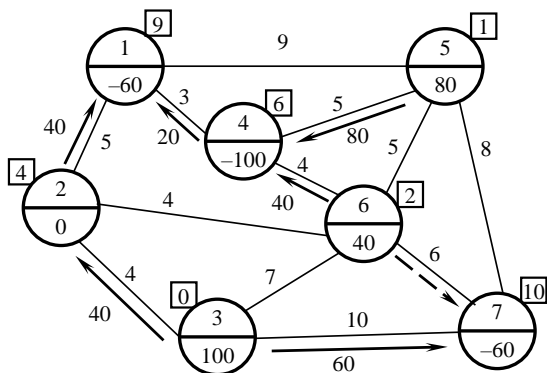


Рисунок 2.8 – Улучшение начального плана перевозок

Используя формулы (2.20), (2.21), получим

$$\begin{aligned} v_2 &= 0 + 4 = 4; & v_5 &= 6 - 5 = 1; \\ v_1 &= 4 + 5 = 9; & v_6 &= 6 - 4 = 2; \\ v_4 &= 9 - 3 = 6; & v_7 &= 0 + 10 = 10. \end{aligned}$$

Проверим выполнение условия оптимальности. Для всех ненагруженных ребер находим оценки Δ_{ij} . Для этого из большего потенциала вершин рассматриваемого ребра вычитаем меньший, а разность вычитаем из показателя c_{ij} , соответствующего данному ребру:

$$\begin{aligned} \Delta_{15} &= 9 - (9 - 1) = 1; & \Delta_{56} &= 5 - (2 - 1) = 4; \\ \Delta_{26} &= 4 - (4 - 2) = 2; & \Delta_{57} &= 8 - (10 - 1) = -1; \\ \Delta_{36} &= 7 - (2 - 0) = 5; & \Delta_{67} &= 6 - (10 - 2) = -2. \end{aligned}$$

Так как ребра (5; 7) и (6; 7) имеют отрицательные оценки, то построенный план не является оптимальным и требует улучшения.

Из ребер (5; 7) и (6; 7), имеющих отрицательные оценки Δ_{ij} , выбираем ребро (6; 7), так как ему соответствует наибольшее по абсолютной величине значение оценки. К ребру (6; 7) подрисовываем пунктирную стрелку, направленную от вершины с меньшим потенциалом к вершине с большим потенциалом. Как видно из рисунка 3.6, при добавлении этой стрелки на сети образовался замкнутый контур 6–7–3–2–1–4–6.

По полученному замкнутому контуру переместим часть груза. Величина перемещаемого груза выбирается равной минимальной поставке на стрелках, имеющих направление, противоположное направлению стрелки (6; 7). Этот минимум достигается на ребре (1; 4) и равен $\lambda = \min(60, 20, 40) = 20$.

Изменим объем перевозок по ребрам полученного замкнутого контура: на стрелках, имеющих то же направление, что и новая стрелка (6; 7), объем перевозок увеличиваем на величину $\lambda = 20$; на стрелках, имеющих противо-

положительное направление, уменьшаем на $\lambda = 20$. Стрелка (1; 4), соответствующая минимальной поставке, ликвидируется. Общее число стрелок сети остается прежним – на единицу меньше числа вершин.

Таким образом, на рисунке 2.9 получили новый план перевозок:

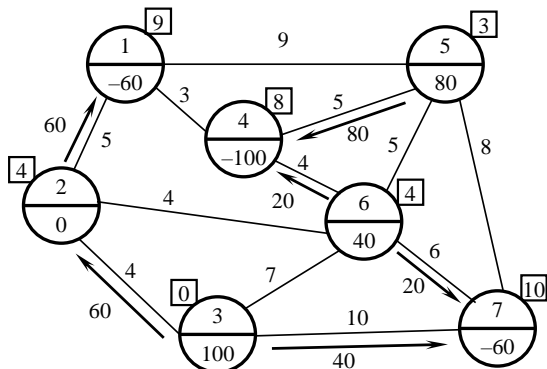


Рисунок 2.9 – Улучшенный методом потенциалов план перевозок

Вычислим значение целевой функции для улучшенного плана перевозок:

$$z(X^1) = 5 \cdot 60 + 4 \cdot 60 + 10 \cdot 40 + 5 \cdot 80 + 4 \cdot 20 + 6 \cdot 20 = 1540 \text{ ден. ед.}$$

Вычислим новые значения потенциалов. Примем $v_3 = 0$.

$$v_2 = 0 + 4 = 4;$$

$$v_6 = 10 - 6 = 4;$$

$$v_1 = 4 + 5 = 9;$$

$$v_4 = 4 + 4 = 8;$$

$$v_7 = 0 + 10 = 10;$$

$$v_5 = 8 - 5 = 3.$$

Проверим выполнение условия оптимальности для ненагруженных ребер:

$$\Delta_{14} = 3 - (9 - 8) = 2;$$

$$\Delta_{36} = 7 - (4 - 0) = 3;$$

$$\Delta_{15} = 9 - (9 - 3) = 3;$$

$$\Delta_{56} = 5 - (4 - 3) = 4;$$

$$\Delta_{26} = 4 - (4 - 4) = 4;$$

$$\Delta_{57} = 8 - (10 - 3) = 1.$$

Построенный план перевозок является оптимальным, потому что удовлетворяет условию (2.23), и единственным, так как нет оценок ненагруженных ребер, равных нулю.

Таким образом, на рисунке 2.9 представлен оптимальный план перевозок однородного груза, при котором суммарные затраты будут минимальными и составят 1540 ден. ед.

3 ЗАДАЧИ ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ

3.1 Основные понятия

Дискретное программирование – раздел математического программирования, изучающий экстремальные задачи, в которых значения переменных принимают конечное или счетное множество.

Целочисленное программирование является частным случаем дискретного. Задачи оптимизации, переменные которых принимают лишь целые значения, называется *задачей целочисленного программирования*. В математической модели задачи целочисленного программирования как целевая функция, так и функции в системе ограничений могут быть линейными, нелинейными и смешанными. Ограничимся случаем, когда целевая функция и система ограничений задачи являются линейными.

Можно выделить следующие основные классы задач дискретного программирования:

- 1) транспортная задача и ее модификации;
- 2) задачи с неделимостью (задача о контейнерных перевозках или задача о рюкзаке);
- 3) экстремальные комбинаторные задачи (задача о коммивояжере, о покрытии, линейный и нелинейный варианты задач о назначениях);
- 4) модели с булевыми переменными (задача о назначениях);
- 5) задачи с разрывными целевыми функциями (транспортная задача с фиксированными доплатами);
- 6) задачи на неклассических областях.

В свою очередь каждый тип моделей подразделяется на линейные и нелинейные, статические и динамические, детерминированные и стохастические и т. д. Наиболее изучен класс задач целочисленного программирования детерминированного типа, в частности, детерминированная задача целочисленного линейного программирования.

Большинство целочисленных и комбинаторных типов задач, таких как задача с неделимостью, задача о коммивояжере, задача календарного планирования принадлежат к разряду так называемых «трудно решаемых задач». Это означает, что вычислительная сложность алгоритма их точного

решения, т. е. зависимость числа элементарных операций (операций сложения или сравнения), необходимых для получения точного решения, от размерности задачи n , является показательной (порядка 2^n), т. е. сравнимой с трудоёмкостью полного перебора вариантов. В качестве n , например, для задачи с неделимостями служит число целочисленных переменных и число ограничений, для задачи коммивояжера – число городов или узлов графа маршрутов, для задачи календарного планирования – число деталей и число станков. Такие задачи называют еще NP -трудными или NP -полными. Получение их точного решения не может быть гарантировано, хотя для некоторых задач данного типа существуют эффективные методы, позволяющие находить точное решение даже при больших размерностях.

Задачи с вычислительной сложностью, определяемой полиномиальной зависимостью от n , называются «эффективно решаемыми задачами». К такому типу задач принадлежат задачи транспортного типа и линейные задачи о назначениях.

Общая постановка задачи целочисленного программирования отличается от общей постановки задачи линейного программирования лишь наличием дополнительного ограничения. Этим ограничением является требование целочисленности, в соответствии с которым значения всех или части переменных модели в оптимальном решении являются целыми неотрицательными числами. При этом, если требование целочисленности распространяется на все переменные, то задачу целочисленного программирования называют *полностью целочисленной задачей*. Если же требование целочисленности относится лишь к части переменных, то задачу называют *частично целочисленной*. Если при этом целевая функция и функции, входящие в ограничения, линейные, то говорят, что данная задача является *задачей целочисленного линейного программирования*. Задачу линейного программирования, отличающуюся от рассматриваемой задачи целочисленного программирования лишь отсутствием требования целочисленности, называют *задачей с ослабленными ограничениями*, соответствующей задаче целочисленного программирования.

3.2 Математическая модель задачи целочисленного линейного программирования

В общем виде математическая модель задачи целочисленного линейного программирования имеет вид

$$z = \sum_{j=1}^n c_j x_j \rightarrow \max (\min) \quad (3.1)$$

при следующих ограничениях:

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = \overline{1, m}, \quad (3.2)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (3.3)$$

$$x_j - \text{целые}, \quad j = 1, 2, \dots, n_1 \leq n. \quad (3.4)$$

Если $n_1 < n$, то задача называется частично целочисленной, если $n_1 = n$ – полностью целочисленной. Среди задач рассматриваемого класса выделяются задачи с *булевыми*, принимающих значения ноль или единица:

$$x_{ij} \in \{0; 1\}, \quad j = 1, 2, \dots, n_1 \leq n. \quad (3.5)$$

Линейная функция (3.1) называется целевой функцией; множество планов X , удовлетворяющих системе ограничений (3.2)–(3.4), называется множеством допустимых решений Ω , $X \in \Omega$. Допустимый план $X \in \Omega$, доставляющий целевой функции (3.1) экстремальное значение, называется оптимальным (X^*).

3.3 Методы решения задач целочисленного линейного программирования

3.3.1 Суть методов решения задач целочисленного линейного программирования

Решение задачи целочисленного программирования, казалось бы, может быть найдено методами решения задач линейного программирования, приведенными выше, с отброшенными условиями целочисленности и последующими округлениями нецелых переменных в ответе. Однако такое округление может привести к решению, далекому от оптимального. Попытка решить задачу с отброшенным условием целочисленности и последующим округлением полученного оптимального плана до ближайших целых значений не всегда состоятельна. С практической точки зрения подобный подход допустим в тех случаях, когда значения переменных, образующих оптимальное решение исходной задачи, достаточно велики и погрешностями округления можно пренебречь.

В первом приближении методы целочисленной оптимизации можно разделить на две основные группы: точные и приближенные.

К точным относятся методы отсечения и комбинаторные:

- 1) метод отсечений (отсекающих плоскостей) Гомори;
- 2) метод ветвей и границ.

Это универсальные методы целочисленного программирования.

Кроме универсальных, имеется много специальных точных методов, учитывающих специфику задачи. Однако точные методы имеют слабую сходимость.

Многие экспериментальные и прикладные задачи не удалось решить точными методами за десятки и сотни тысяч итераций, хотя их конечность теоретически доказана.

Трудности машинной реализации точных методов привели к появлению различного рода приближенных или эвристических методов, построенных на использовании особенностей конкретной задачи.

Среди *приближенных* методов наметились два направления:

- 1) разработка детерминированных эвристических алгоритмов, учитывающих специфику задачи;
- 2) использование случайного поиска в сочетании с локальной оптимизацией.

Эвристические и метаэвристические методы не обеспечивают получения точного решения и могут использоваться в случаях, когда применение предыдущих методов невозможно или не приводит к успеху. К тому же эти методы можно использовать для решения задач любой сложности.

3.3.2 Метод отсечений (отсекающих плоскостей) Гомори

Метод отсечений (отсекающих плоскостей) Гомори применим к задачам, где требование целочисленности налагаются на все без исключения переменные. Метод трудоемок для задач большой размерности.

Рассмотрим суть метода Гомори на примере задачи с двумя переменными, которую можно представить графически (рисунок 3.1).

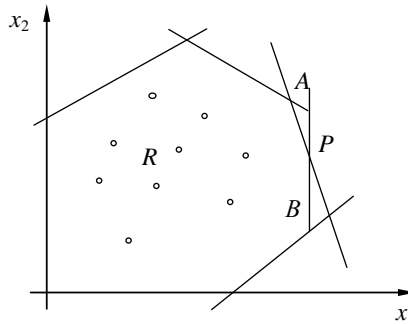


Рисунок 3.1 – Графическая интерпретация метода Гомори

1 Ограничения задачи дают область допустимых решений в виде многогранника решений.

2 Вершины многогранника решений могут быть точками экстремума, но не отвечают требованию целочисленности. Пусть точка P – точка экстремума, не отвечающая требованию целочисленности.

3 Внутри многогранника решений и на его границах имеется множество точек R , соответствующее планам с целочисленными значениями переменных.

4 Делается отсечение (прямая AB), что соответствует вводу еще одного неравенства в систему ограничений. Это отсечение гарантирует, что оптимальный нецелочисленный план (точка P) не будет в последующих расчетах допустимым. В то же время множество целых точек не сужается.

Алгоритм Гомори конечен и дает точное решение.

Сформулируем **алгоритм метода Гомори** для решения задач целочисленного программирования.

1 Решается задача (3.1)–(3.3) симплексным или графическим методом до получения оптимального плана без учета требования целочисленности.

2 Полученное оптимальное решение, если оно существует, проверяется на целочисленность. Если условие целочисленности выполняется по всем переменным, то оптимальное решение задачи (3.1)–(3.4) совпадает с оптимальным решением задачи (3.1)–(3.3). Если это условие не выполняется хотя бы по одной переменной, то переходят к третьему этапу. Если задача (3.1)–(3.3) оказывается неразрешимой, то задача (3.1)–(3.4) тоже не имеет решения.

3 Строится дополнительное ограничение, отсекающее часть области, в которой содержится оптимальное решение задачи (3.1)–(3.3) и не содержится ни одного допустимого решения задачи (3.1)–(3.4).

4 Последний этап предусматривает возвращение к задаче линейного программирования с отброшенными условиями целочисленности, но с расширенной системой ограничений, в которую включено дополнительное ограничение, полученное на третьем этапе. К расширенной системе ограничений вновь применяется симплексная процедура. Если найденное таким образом решение будет опять нецелым, то формируется новое дополнительное ограничение и процесс вычислений повторяется.

Алгоритм Гомори позволяет за конечное число шагов прийти к оптимальному целочисленному решению, если оно существует.

Рассмотрим пример решения задачи целочисленного линейного программирования методом Гомори.

Пример 3.1.

$$\begin{aligned} z(X) &= 3x_1 + x_2 \rightarrow \max, \\ 3x_1 + 4x_2 &\leq 24, \\ 3x_1 - x_2 &\leq 6, \\ x_1 \geq 0, \quad x_2 &\geq 0, \\ x_1, x_2 &\in Z. \end{aligned}$$

По методу Гомори для определения оптимального плана исходной задачи решим вначале задачу линейного программирования симплексным методом без учета условия целочисленности переменных:

	БП	С _Б	x_1	x_2	x_3	x_4	Решение
0	x_3	0	3	4	1	0	24
	x_4	0	3	-1	0	1	6
	z-строка		-3	-1	0	0	0

	БП	С _Б	x_1	x_2	x_3	x_4	Решение
1	x_3	0	0	5	1	-1	18
	x_1	3	1	-0,33	0	0,33	2
	z-строка		0	-2	0	1	6

	БП	С _Б	x_1	x_2	x_3	x_4	Решение
2	x_2	1	0	1	0,2	-0,2	3,6
	x_1	3	1	0	0,07	0,27	3,2
	z-строка		0	0	0,4	0,6	13,2

В результате получаем оптимальный нецелочисленный план

$$X_{\text{нц}}^0 = (3,2; 3,6); z(X_{\text{нц}}^0) = 13,2.$$

В полученном плане две компоненты нецелые.

Построим дополнительное линейное ограничение. Находим:

$$\max \{ \{ x_1^0 \}, \{ x_2^0 \} \} = \max \{ \{ 3,2 \}, \{ 3,6 \} \} = 0,6 = \{ x_2^0 \}.$$

По строке x_2^0 строим дополнительное ограничение:

$$0,2 x_3 - 0,2 x_4 \geq 0,6,$$

В полученное неравенство введем дополнительную переменную x_5 :

$$0,2 x_3 - 0,2 x_4 - x_5 = 0,6,$$

Умножив на (-1), получим ограничение:

$$-0,2 x_3 + 0,2 x_4 + x_5 = -0,6,$$

Вводится новая строка и столбец для x_5 :

	БП	С _Б	x_1	x_2	x_3	x_4	x_5	Решение
0	x_2	1	0	1	0,2	-0,2	0	3,6
	x_1	3	1	0	0,067	0,267	0	3,2
	x_5	0	0	0	-0,2	0,2	1	-0,6
	z-строка		0	0	0,4	0,6	0	13,2

	БП	C_B	x_1	x_2	x_3	x_4	x_5	Решение
1	x_2	1	0	1	0	0	1	3
	x_1	3	1	0	0	0,34	0,34	3
	x_3	0	0	0	1	-1	-5	3
	z -строка		0	0	0	1	2	12

В результате получено оптимальное целочисленное решение исходной задачи:

$$X_{ц}^* = (3; 3); z(X_{ц}^*) = -12.$$

3.3.3 Метод ветвей и границ решения задач целочисленного линейного программирования

Метод ветвей и границ относится к группе комбинаторных методов. Комбинаторные методы исходят из конечности числа допустимых планов задачи и заменяют полный перебор всех планов их частичным направленным перебором. Комбинаторные методы в значительно меньшей степени подвержены в процессе вычислений влиянию ошибок округления, поэтому являются более предпочтительными по сравнению с методами отсечения. Метод ветвей и границ – один из наиболее эффективных методов решения задач комбинаторного типа.

Рассмотрим общую задачу целочисленного программирования

$$z = f(X) \rightarrow \max, \quad (3.6)$$

$$X \in \Omega, \quad (3.7)$$

где Ω – конечное множество допустимых планов.

Сформулируем **алгоритм метода ветвей и границ** для решения задач целочисленного программирования:

1 Находим верхнюю границу (оценку) функции $f(X)$, $X \in \Omega$, т. е. такое число $\varphi^0(\Omega)$, что для любых $X \in \Omega$:

$$f(X) \leq \varphi^0(\Omega).$$

Если при этом удастся найти такой план X^0 задачи (3.6)–(3.7), для которого выполняется равенство

$$f(X^0) = \varphi^0(\Omega),$$

то X^0 – оптимальный план задачи (3.6)–(3.7).

2 Если оптимальный план не найден, то некоторым способом разбиваем множество Ω на конечное число непересекающихся подмножеств Ω_r^1 :

$$\Omega = \bigcup_{r=1}^{r_1} \Omega_r^1, \quad \bigcup_{r=1}^{r_1} \Omega_r^1 = \emptyset.$$

и находим для каждого из этих подмножеств верхнюю границу $\varphi_1(\Omega_r^1)$ ($r=1, \dots, r_1$).

Если при этом удастся найти такой план $x_r \in \Omega_r^1$ ($1 \leq r \leq r_1$), что выполняется соотношение

$$f(x_r^1) = \varphi(\Omega_r^1) \geq \varphi(\Omega_r), \quad (r=1, \dots, r_1),$$

то x^1 – оптимальный план задачи (3.6)–(3.7).

Если же такой план не найден, то выбираем подмножество Ω_r^1 с наибольшей верхней границей (перспективное подмножество) и разбиваем его на несколько непересекающихся подмножеств Ω_s^2 ($s=1, \dots, s_1$).

Для каждого нового подмножества находим верхнюю границу $\varphi(\Omega_s^2)$. Если будет найден такой план x_k^2 , что

$$f(x_k^2) = \varphi(\Omega_k^2) \geq \varphi(\Omega_s^2),$$

то x_k^2 – оптимальный план задачи.

Если оптимальный план не найден, то дальнейшему ветвлению подвергается подмножество с наибольшей верхней границей и т. д.

Процесс продолжается до получения оптимального плана. Способы ветвления и нахождения верхних границ выбираются для каждой конкретной задачи целочисленного программирования.

Процесс сопровождается построением дерева ветвления (рисунок 3.2).

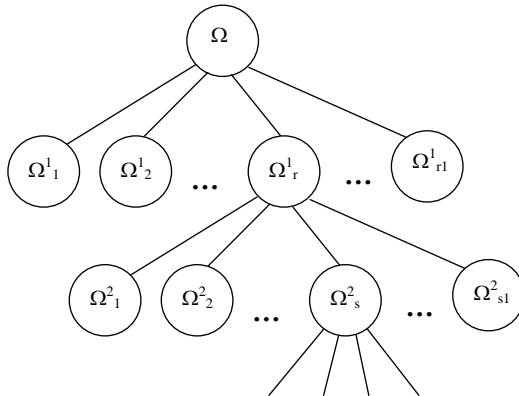


Рисунок 3.2 – Дерево ветвления

Рассмотрим пример решения задачи целочисленного линейного программирования методом ветвей и границ.

Пример 3.2.

$$\begin{aligned} z(X) &= 2x_1 + x_2 \rightarrow \max, \\ 6x_1 - 4x_2 &\leq 24, \\ x_1 + 2x_2 &\leq 10, \\ x_1 \geq 0, \quad x_2 &\geq 0, \\ x_1, x_2 &\in Z. \end{aligned}$$

Находим множество допустимых решений Ω – многоугольник $OABC$ (рисунок 3.3).

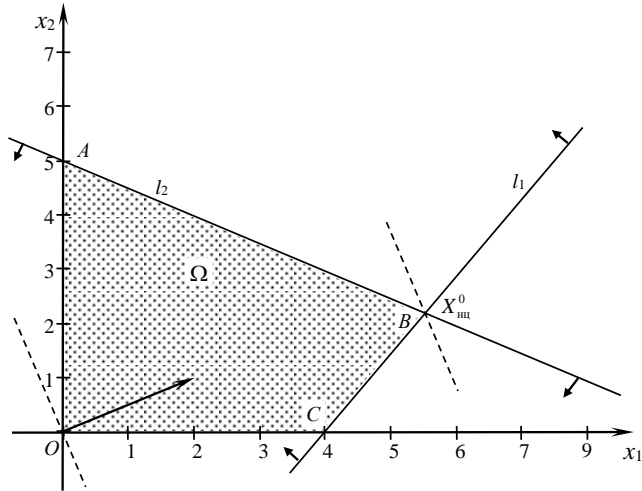


Рисунок 3.3 – Графическое решение

Из решения системы уравнений

$$\begin{cases} 6x_1 - 4x_2 = 24, \\ x_1 + 2x_2 = 10, \end{cases}$$

получим оптимальный нецелочисленный план

$$X_{шт}^0 = (5,5; 2,25).$$

По методу ветвей и границ в качестве оценки множества Ω принимаем значение целевой функции на оптимальном целочисленном плане, т. е.

$$\zeta(\Omega) = z(X_{шт}^0) = 13,25.$$

Выбрав нецелую переменную x_1 , разбиваем множество Ω на два непересекающихся подмножества:

$$A_1 = \{X \in \Omega, x_1 \leq [x_1^0] = [5,5] = 5\},$$

$$A_2 = \{X \in \Omega, x_1 \geq [x_1^0] + 1 = [5,5] + 1 = 6\}.$$

Таким образом, имеем две подзадачи:

$$z(X) = 2x_1 + x_2 \rightarrow \max,$$

$$6x_1 - 4x_2 \leq 24,$$

$$x_1 + 2x_2 \leq 10,$$

$$x_1 \leq 5,$$

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

$$x_1, x_2 \in \mathbb{Z}.$$

$$z(X) = 2x_1 + x_2 \rightarrow \max,$$

$$6x_1 - 4x_2 \leq 24,$$

$$x_1 + 2x_2 \leq 10,$$

$$x_1 \geq 6,$$

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

$$x_1, x_2 \in \mathbb{Z}.$$

Решение каждой из подзадач представлено на рисунке 3.4.

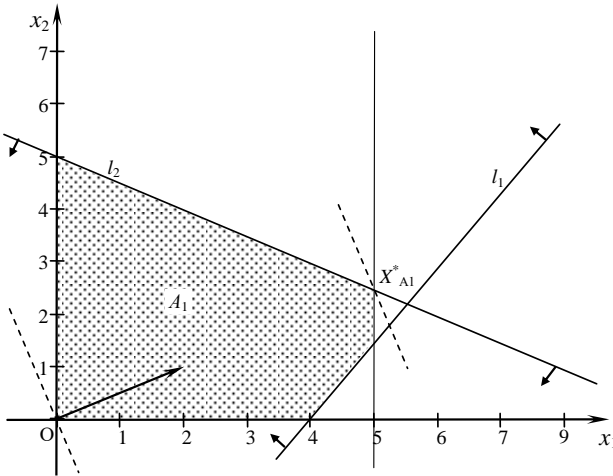


Рисунок 3.4 – Графическое решение

Для первой из подзадач имеем:

$$\begin{cases} x_1 + 2x_2 = 10, \\ x_1 = 5, \end{cases}$$

$$X_{A_1}^* = (5; 2,5); \zeta(A_1) = 12,5.$$

Для второй: $X_{A_2}^* = \emptyset$.

Поскольку $X_{A_1}^* = (5; 2,5)$ – оптимальный нецелочисленный план, то разбиваем подмножество A_1 на два непересекающихся подмножества B_1 и B_2 следующим образом:

$$B_1 = \{X \in A_1, x_2 \leq [x_2^*] = [2,5] = 2\},$$

$$B_2 = \{X \in A_1, x_2 \geq [x_2^*] + 1 = [2,5] + 1 = 3\}.$$

Таким образом, имеем две подзадачи:

$$\begin{aligned} z(X) &= 2x_1 + x_2 \rightarrow \max, \\ 6x_1 - 4x_2 &\leq 24, \\ x_1 + 2x_2 &\leq 10, \\ x_1 &\leq 5, \\ x_2 &\leq 2, \\ x_1 \geq 0, \quad x_2 &\geq 0, \\ x_1, x_2 &\in Z. \end{aligned}$$

$$\begin{aligned} z(X) &= 2x_1 + x_2 \rightarrow \max, \\ 6x_1 - 4x_2 &\leq 24, \\ x_1 + 2x_2 &\leq 10, \\ x_1 &\leq 5, \\ x_2 &\geq 3, \\ x_1 \geq 0, \quad x_2 &\geq 0, \\ x_1, x_2 &\in Z. \end{aligned}$$

Решение каждой из подзадач представлено на рисунке 3.5.

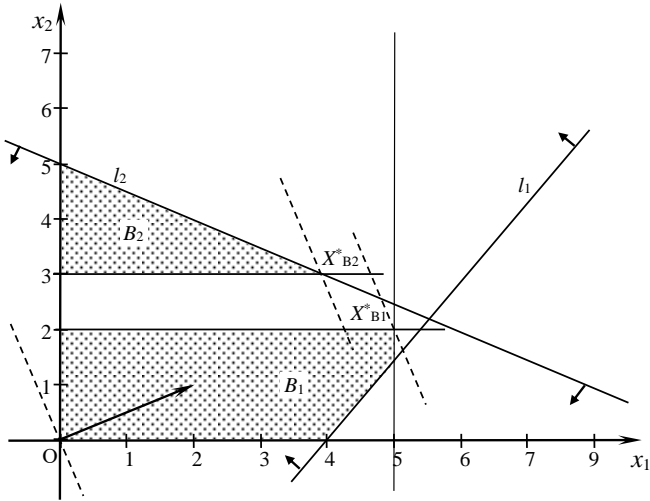


Рисунок 3.5 – Графическое решение

Для первой из подзадач имеем:

$$\begin{cases} x_1 = 5, \\ x_2 = 2, \end{cases}$$

$$X_{B_1}^* = (5; 2); \quad \zeta(B_1) = 12,$$

для второй –

$$\begin{cases} x_1 + 2x_2 = 10, \\ x_2 = 3, \end{cases}$$

$$X_{B_2}^* = (4; 3); \quad \zeta(B_2) = 11.$$

Изобразим графически решение задачи.

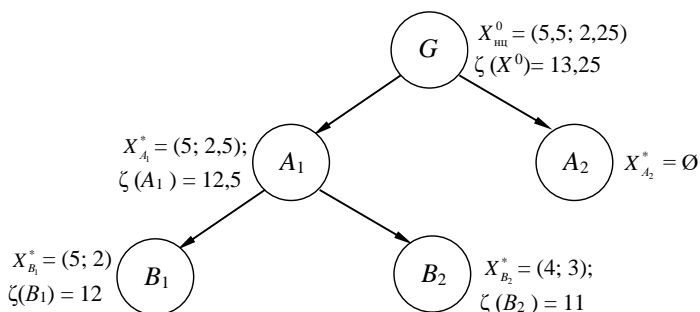


Рисунок 3.6 – Дерево ветвления

Проводя анализ решения, заключаем, что оптимальный целочисленный план найден:

$$X_{B_1}^* = (5; 2); \quad \zeta(B_1) = 12.$$

3.4 Задача о коммивояжере

3.4.1 Математическая модель задачи о коммивояжере и методы ее решения

Постановка задачи. Имеется n пунктов транспортной сети, в которых должен побывать коммивояжер. Задана матрица стоимостей c_{ij} ($i, j = \overline{1, n}$) на переход между пунктами транспортной сети. Коммивояжер должен выйти из одного из пунктов, посетить только один раз каждый из n пунктов и вернуться в исходный пункт.

Необходимо найти такой маршрут передвижения коммивояжера между пунктами транспортной сети, чтобы суммарная стоимость была минимальной.

Обозначим неизвестные

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер переезжает из пункта } i \text{ в пункт } j; \\ 0 & \text{в противном случае.} \end{cases}$$

Требуется найти $X = \|x_{ij}\|$, чтобы суммарная стоимость была минимальной:

$$z(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (3.6)$$

при следующих ограничениях:

– коммивояжер въезжает в каждый пункт только один раз:

$$\sum_{j=1}^n x_{ij} = 1, \quad j = \overline{1, n}, \quad (3.7)$$

– коммивояжер выезжает из каждого пункта только один раз:

$$\sum_{i=1}^n x_{ij} = 1, \quad i = \overline{1, n}, \quad (3.8)$$

– условие целочисленности

$$x_{ij} \in \{0; 1\}, \quad i, j = \overline{1, n}, \quad (3.9)$$

– устранения подциклов:

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i, j = \overline{1, n}, \quad i \neq j, \quad (3.10)$$

где переменные u могут принимать произвольные действительные значения.

Условие (3.10) обеспечивает замкнутость цикла только для n пунктов транспортной сети.

Задача о коммивояжере (3.6)–(3.10) в математической постановке эквивалентна задаче упорядочения конечного числа работ на оборудовании при учете потерь от переналадок. Во многих случаях длительность переналадок оборудования перед выполнением некоторой работы нельзя считать не зависящей от характера предшествующей работы. Иногда разброс длительностей переналадок становится основным критерием оценки расписания.

Таким образом, возникает задача отыскания такой последовательности, в которой суммарные потери от переналадок оборудования будут минимальны. Для задачи упорядочения каждая работа соответствует пункту транспортной сети, а длительность переналадки – стоимости на переход между пунктами транспортной сети. Матрица C может быть несимметричной, так как затраты, связанные с переналадкой с i -й работы на j -ю, как правило, отличны от затрат при переходе от j -й работы к i -й.

Существует несколько точных методов решения задачи о коммивояжере: ветвей и границ, динамического программирования и др.

Методом ветвей и границ с использованием современных компьютеров можно решать задачи коммивояжера для $n \leq 40$, динамического программирования – для $n \leq 17$. В связи с малой эффективностью точных методов получили широкое применение эвристические.

В настоящее время существует более ста приближенных методов решения задачи коммивояжера, среди которых наиболее прост метод *ближайшего соседа*. Он реализует требование включать в искомый замкнутый контур вершину, ближайшую к только что найденной. Алгоритм «ближайшего соседа» состоит в последовательном добавлении к начальной вершине следующей ближайшей к ней и т. д. Отметим, что если ближайший сосед для некоторой вершины уже вошел в контур, то берется следующая по близости вершина и т. д. Метод очень прост, однако степень приближения к оптимальному решению существенно зависит от выбора начальной вершины. Поэтому алгоритм целесообразно применять, начиная с каждой вершины, и затем выбрать замкнутый контур наименьшей длины.

3.4.2 Решение задачи о коммивояжере методом ветвей и границ (алгоритм Литтла)

Если в задаче о коммивояжере (3.6)–(3.10) считать пункты транспортной сети вершинами графа, а коммуникации (i, j) – его дугами, то требование нахождения минимального пути, проходящего один раз через каждый пункт, и возвращения в него можно рассматривать как нахождение на графе контура минимальной длины.

Замкнутый контур, проходящий один раз через каждую вершину графа, называется *гамильтоновым контуром*. Таким образом, задача коммивояжера состоит в нахождении на графе гамильтонова контура минимальной длины.

Рассмотрим *алгоритм Литтла* для нахождения минимального гамильтонова контура на графе с n вершинами.

Если между вершинами i и j нет дуги, то ставится символ ∞ . Этот же символ ставится на главной диагонали, что означает запрет на возвращение в вершину, через которую уже проходил контур. Основная идея метода состоит в том, что вначале строят нижнюю границу длин множества гамильтоновых контуров Ω^0 . Затем множество контуров Ω^0 разбивается на два подмножества таким образом, чтобы первое подмножество Ω_{ij}^1 состояло из гамильтоновых контуров, содержащих некоторую дугу (i, j) , а другое подмножество Ω_{ij}^1 не содержало этой дуги. Для каждого из подмножеств определяются нижние границы по тому же правилу, что и для первоначального множества гамильтоновых контуров. Полученные нижние границы подмножеств Ω_{ij}^1 и Ω_{ij}^1 оказываются не меньше нижней границы для всего множества гамильтоновых контуров, т. е.

$$\varphi(\Omega^0) \leq \varphi(\Omega_{ij}^1) \equiv \varphi_{ij}^1,$$

$$\varphi(\Omega^0) \leq \varphi(\Omega_{ij}^1) \equiv \varphi_{ij}^1.$$

Сравнивая нижние границы φ_{ij}^1 и φ_{ij}^1 подмножеств Ω_{ij}^1 и Ω_{ij}^1 , можно выделить среди них то, которое с большей вероятностью содержит гамильтонов контур минимальной длины. Затем одно из подмножеств Ω_{ij}^1 или Ω_{ij}^1 по аналогичному правилу разбивается на два новых Ω_{ij}^2 и Ω_{ij}^2 . Для них снова отыскиваются нижние границы φ_{ij}^2 и φ_{ij}^2 и т. д. Процесс ветвления продолжается до тех пор, пока не отыщется единственный гамильтонов контур. Его называют *первым рекордом*. Затем просматривают оборванные вет-

ви. Если их нижние границы больше длины первого рекорда, то задача решена: если же есть такие, для которых нижние границы меньше, чем длина первого рекорда, то подмножество с наименьшей нижней границей подвергается дальнейшему ветвлению, пока не убеждаются, что оно не содержит лучшего гамильтонова контура. Если же такой найдется, то анализ оборванных ветвей продолжается относительно нового значения длины контура. Его называют *вторым рекордом*. Процесс решения заканчивается, когда будут проанализированы все подмножества.

Для практической реализации идеи метода ветвей и границ применительно к задаче коммивояжера нужно указать прием определения нижних границ подмножеств и разбиения множества гамильтоновых контуров на подмножества (ветвление). Определение нижних границ базируется на следующем утверждении.

Если к элементам любого ряда матрицы задачи коммивояжера (строке или столбцу) прибавить или вычесть из них некоторое число, то от этого оптимальность плана не изменится. Длина же любого гамильтонова контура изменится на данную величину. Иначе: если план X^* – оптимальный план задачи (3.6)–(3.10), то он также оптимален для задачи с функцией цели z' и с матрицей

$$C' = [c_{ij} \pm u_i \pm v_j] \quad (i, j = 1, \dots, n), \quad X \in \Omega.$$

Опираясь на это утверждение, вычтем из каждой строки матрицы коммивояжера C число u_i , равное минимальному элементу этой строки. Перейдем к матрице C' с элементами

$$c'_{ij} = c_{ij} - \min_j c_{ij}.$$

Вычтем затем из каждого j -го столбца матрицы C' число v_j , равное минимальному элементу этого столбца. Получим матрицу C'' с элементами

$$c''_{ij} = c'_{ij} - \min_j c'_{ij} - \min_i c'_{ij}.$$

Матрица C'' называется приведенной по строкам и столбцам. В приведенной матрице C'' в каждой строке и каждом столбце содержится хотя бы один нуль. Все элементы матрицы C'' неотрицательны: $c''_{ij} \geq 0$.

Величина $\sum_{i=1}^n u_i + \sum_{j=1}^n v_j = \varphi_0$ называется *константой приведения*.

Если дана задача коммивояжера с матрицей $C = [c_{ij}]$, то константа приведения этой матрицы φ_0 определяет нижнюю границу гамильтоновых контуров $\varphi(\Omega^0) = \varphi_0$.

Рассмотрим вопрос о разбиении множества гамильтоновых контуров на подмножества. Это делается с помощью некоторой перспективной дуги (i_0, j_0) . Первое подмножество $\Omega^1_{i_0, j_0}$ составляет гамильтонов контур, содержа

щий перспективную дугу (i_0, j_0) , второе подмножество $\Omega_{i_0 j_0}^1$ – не содержит дуги (i_0, j_0) .

Включение дуги (i_0, j_0) в контур приводит к автоматическому сокращению матрицы C на строку i_0 и столбец j_0 . Так как гамильтонов контур не может одновременно содержать дуги (i_0, j_0) и (j_0, i_0) , то, включив (i_0, j_0) в контур, следует (j_0, i_0) исключить, т. е. путь (j_0, i_0) «блокировать бесконечностью» ($c_{j_0 i_0} = \infty$). Сократив размер матрицы, можно улучшить нижнюю границу подмножества гамильтоновых контуров.

Для выделения претендентов на включение в множество дуг, по которым производится ветвление, рассмотрим в приведенной матрице C'' все элементы, равные нулю ($c''_{ij} = 0$). Найдем степени θ_{ij} нулевых элементов этой матрицы. Пусть $c''_{i_0 j_0} = 0$. Степень нулевого элемента равна сумме минимальных элементов в строке i_0 и столбце j_0 при блокировании перехода (i_0, j_0) бесконечностью, т. е.

$$\theta_{i_0 j_0} = \min_{j \neq j_0} c_{i_0 j} + \min_{i \neq i_0} c_{i j_0}.$$

С наибольшей вероятностью искомому гамильтонову контуру принадлежат дуги с максимальной степенью нуля. Пусть $\max_{c''_{ij}=0} \theta_{ij} = \theta_{i_0 j_0}$. Дуга (i_0, j_0)

включается в множество $\Omega_{i_0 j_0}^1$ и является запретной для множества $\Omega_{i_0 j_0}^1$.

Для получения матрицы контуров $\Omega_{i_0 j_0}^1$, включающих дугу (i_0, j_0) , вычеркиваем в матрице C'' строку i_0 и столбец j_0 , а чтобы не допустить образования негамильтонова контура, заменим симметричный относительно главной диагонали элемент (j_0, i_0) на ∞ .

Множество гамильтоновых контуров $\Omega_{i_0 j_0}^1$, не включающих дугу (i_0, j_0) , получаем путем замены элемента $c''_{i_0 j_0}$ в матрице C'' на ∞ .

Далее приводим матрицы подмножеств $\Omega_{i_0 j_0}^1$ и $\Omega_{i_0 j_0}^1$ и определяем их нижние границы. Пусть $h_{i_0 j_0}^1$ и $h_{i_0 j_0}^1$ – константы приведения матриц множеств $\Omega_{i_0 j_0}^1$ и $\Omega_{i_0 j_0}^1$. Тогда :

$$\varphi_{i_0 j_0}^1 = \varphi(\Omega_{i_0 j_0}^1) = \varphi(\Omega^0) + h_{i_0 j_0}^1,$$

$$\varphi_{i_0 j_0}^1 = \varphi(\Omega_{i_0 j_0}^1) = \varphi(\Omega^0) + h_{i_0 j_0}^1.$$

Производим ветвление подмножества с меньшей нижней границей и т. д. Найдя, таким образом, хотя бы один гамильтонов контур (первый рекорд), сравним его длину с нижними границами оборванных ветвей. Процесс ветвления сопровождается построением дерева ветвления. Если длина контура не превышает нижних границ оборванных ветвей дерева, то задача решена. В противном случае, ветвление продолжается с подмножества, имеющего наименьшую нижнюю границу.

В случае, если каждая дуга графа связана с получением, например, прибыли, задачу коммивояжера нужно решать на максимум. Задачу на максимум легко свести к алгоритму Литгла для решения задачи коммивояжера на минимум. Для этого найдем максимальный элемент в каждом столбце:

$$l_j = \max_i c_{ij}.$$

Построим матрицу

$$C'_{ij} = [c'_{ij}] = [\max_i c_{ij} - c_{ij}].$$

Имеем

$$z' = \sum_{j=1}^n l_j - z.$$

Отсюда следует, что z' достигает минимума для того же плана, для которого z достигает максимума. В случае решения задачи коммивояжера на максимум в исходной матрице по главной диагонали ставится символ ∞ . Этот же приписывается дугам, по которым нет перемещения.

Пример 3.3. Пусть задано $n = 5$ пунктов транспортной сети, в которых должен побывать коммивояжер. Задана матрица стоимостей c_{ij} ($i, j = 1, 5$) на переход между пунктами транспортной сети (таблица 3.1). Коммивояжер должен выйти из одного из пунктов, посетить только один раз каждый из пяти пунктов и вернуться в исходный пункт. Необходимо найти такой маршрут передвижения коммивояжера между пунктами транспортной сети, чтобы суммарная стоимость была минимальной.

Таблица 3.1 – Матрица стоимостей

$i \backslash j$	1	2	3	4	5
1	∞	60	80	30	12
2	20	∞	36	14	18
3	40	60	∞	2	20
4	50	20	70	∞	10
5	18	16	14	12	∞

Изобразим последовательность передвижения коммивояжера между пунктами транспортной сети в виде графа (рисунок 3.7), каждая вершина

которого будет соответствовать определенному пункту, а каждая дуга – передвижению коммивояжера с одного пункта в другой. Каждой дуге сопоставим число, характеризующее стоимость перемещения.

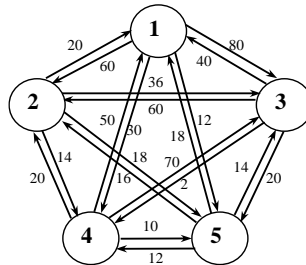


Рисунок 3.7 – Условие задачи о коммивояжере

Решение задачи сводится к поиску на графе гамильтонова контура наименьшей продолжительности, т.е. по существу к решению задачи коммивояжера. Решим задачу о коммивояжере методом ветвей и границ.

1 Присоединим справа к матрице (см. таблицу 3.1) столбец $u_i = \min_j c_{ij}$ (таблица 3.2). Вычтем элементы u_i из соответствующих элементов матрицы C и получим матрицу C' , приведенную по строкам (таблица 3.3): $C' = [c'_{ij}] = [c_{ij} - \min_j c_{ij}]$.

Таблица 3.2 – Матрица C

$i \backslash j$	1	2	3	4	5	u_i
1	∞	60	80	30	12	12
2	20	∞	36	14	18	14
3	40	60	∞	2	20	2
4	50	20	70	∞	10	10
5	18	16	14	12	∞	12
$\sum_{i=1}^n u_i =$						50

Таблица 3.3 – Матрица C'

$i \backslash j$	1	2	3	4	5	
1	∞	48	68	18	0	
2	6	∞	22	0	4	
3	38	58	∞	0	18	
4	40	10	60	∞	0	
5	6	4	2	0	∞	
v_j	6	4	2	0	0	$\sum_{j=1}^n v_j = 12$

2 Присоединим снизу к матрице (см. таблицу 3.3) строку

$$v_j = \min_j c'_{ij}.$$

Вычтя элементы v_j из соответствующих элементов столбцов матрицы C' , получим матрицу

$$C'' = [c''_{ij}] = [c'_{ij} - \min_i c'_{ij}].$$

Такая матрица называется приведенной по строкам и столбцам и обозначена $\tilde{\Omega}^0$. В каждой строке и каждом столбце такой матрицы имеется хотя бы один нуль (таблица 3.4).

Таблица 3.4 – Матрица гамильтоновых контуров $\tilde{\Omega}^0$

$i \backslash j$	1	2	3	4	5
1	∞	44	66	18	0^{18}
2	0^0	∞	20	0^0	4
3	32	54	∞	0^{18}	18
4	34	6	58	∞	0^6
5	0^0	0^6	0^{20}	0^0	∞

3 Найдем константу приведения:

$$\varphi_0 = \sum_{i=1}^n u_i + \sum_{j=1}^n v_j = 50 + 12 = 62.$$

Она определяет нижнюю границу множества всех гамильтоновых контуров, т. е.

$$\varphi_0 = \varphi(\Omega^0) \leq z(X), X \in \Omega.$$

4 Найдем степени нулей для приведенной по строкам и столбцам матрицы $\tilde{\Omega}^0$ (см. таблицу 3.4). Для этого нуль в матрице $\tilde{\Omega}^0$, степень которого отыскивается, мысленно заменяем на ∞ и находим сумму минимальных элементов строки и столбца, где он находится.

5 Выбираем дугу (i_0, j_0) , для которой степень нулевого элемента наибольшая. В нашем случае она равна 20 и соответствует дуге (5, 3). Таким образом, претендентом на включение в гамильтонов контур является дуга (5, 3).

6 Разбиваем множество всех гамильтоновых контуров $\tilde{\Omega}^0$ на два: Ω_{53}^1 и Ω_{53}^1 . Матрицу Ω_{53}^1 , содержащую дугу (5, 3), получаем из таблицы 3.4 путем вычеркивания строки 5 и столбца 3. Чтобы не допустить образования негамильтонова контура, заменяем элемент c''_{53} на ∞ (таблица 3.5).

Таблица 3.5 – Матрица гамильтоновых контуров Ω_{53}^1

$i \backslash j$	1	2	4	5
1	∞	44	18	0
2	0	∞	0	4
3	32	54	0	∞
4	34	6	∞	0
		6		

7 Выполняем дополнительное приведение матрицы контуров Ω_{53}^1 . Находим $h_{53}^1 = 6$ (таблица 3.6). Следовательно, нижняя граница множества Ω_{53}^1 равна $\varphi(\tilde{\Omega}_{53}^1) = 62 + 6 = 68$.

Таблица 3.6 – Матрица гамильтоновых контуров $\tilde{\Omega}_{53}^1$

$i \backslash j$	1	2	4	5
1	∞	38	18	0^{18}
2	0^{32}	∞	0^0	4
3	32	48	0^{32}	∞
4	34	0^{38}	∞	0^0

8 Матрицу гамильтоновых контуров Ω_{53}^1 получим из таблицы 3.4 путем замены элемента c_{53}'' на ∞ (таблица 3.7).

Таблица 3.7 – Матрица гамильтоновых контуров Ω_{53}^1

$i \backslash j$	1	2	3	4	5
1	∞	44	66	18	0
2	0	∞	20	0	4
3	32	54	∞	0	18
4	34	6	58	∞	0
5	0	0	∞	0	∞
			20		

9 Находим дополнительную константу приведения для множества контуров Ω_{53}^1 : $h_{53}^1 = 20$.

Следовательно, нижняя граница этого множества $\varphi(\tilde{\Omega}_{53}^1) = 62 + 20 = 82$ (таблица 3.8).

Таблица 3.8 – Матрица гамильтоновых контуров $\tilde{\Omega}_{53}^1$

$i \backslash j$	1	2	3	4	5
1	∞	44	46	18	0
2	0	∞	0	0	4
3	32	54	∞	0	18
4	34	6	38	∞	0
5	0	0	∞	0	∞

10 Сравниваем нижние границы подмножеств $\tilde{\Omega}_{53}^1$ и Ω_{53}^1 . Так как $\varphi(\tilde{\Omega}_{53}^1) = 68 < \varphi(\Omega_{53}^1) = 82$, дальнейшему ветвлению подвергаем подмножество $\tilde{\Omega}_{53}^1$. Процесс отыскания оптимального плана сопровождается построением дерева ветвления (рисунок 3.8).

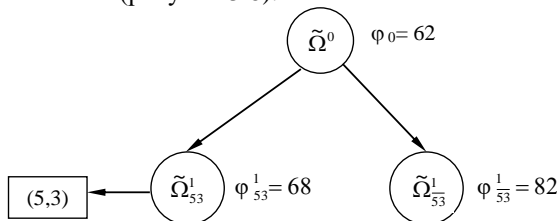


Рисунок 3.8 – Дерево ветвления

На рисунке 3.9 представлено ветвление по дуге (5, 3).

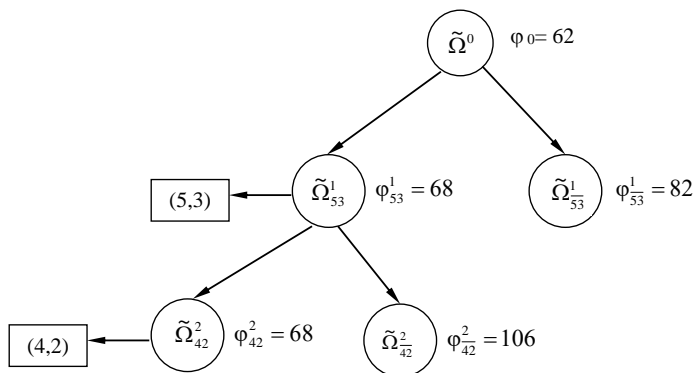


Рисунок 3.9 – Дерево ветвления по дуге (5, 3)

Далее переходим к ветвлению подмножества $\tilde{\Omega}_{53}^1$. Находим степени нулей этой матрицы (см. таблицу 3.6). Претендентом на включение в гамиль-

тонов контур является дуга (4, 2) с наибольшей степенью нуля. Разбиваем множество $\tilde{\Omega}_{53}^1$ на два подмножества Ω_{42}^2 и Ω_{42}^2 (таблицы 3.9 и 3.10).

Таблица 3.9 – Матрица гамильтоновых контуров $\tilde{\Omega}_{42}^2$

$i \backslash j$	1	4	5
1	∞	18	0^{22}
2	0^{36}	∞	4
3	32	0^{50}	∞

Таблица 3.10 – Матрица гамильтоновых контуров $\tilde{\Omega}_{42}^2$

$i \backslash j$	1	2	4	5
1	∞	38	18	0
2	0	∞	0	4
3	32	48	0	∞
4	34	0	∞	0
		38		

Определяем константы приведения для этих матриц: $h_{42}^2 = 0$, $h_{42}^2 = 38$. Следовательно, $\varphi_{42}^2 = 68$, $\varphi_{42}^2 = 106$. На рисунке 3.10 представлено ветвление с использованием дуги (4, 2).

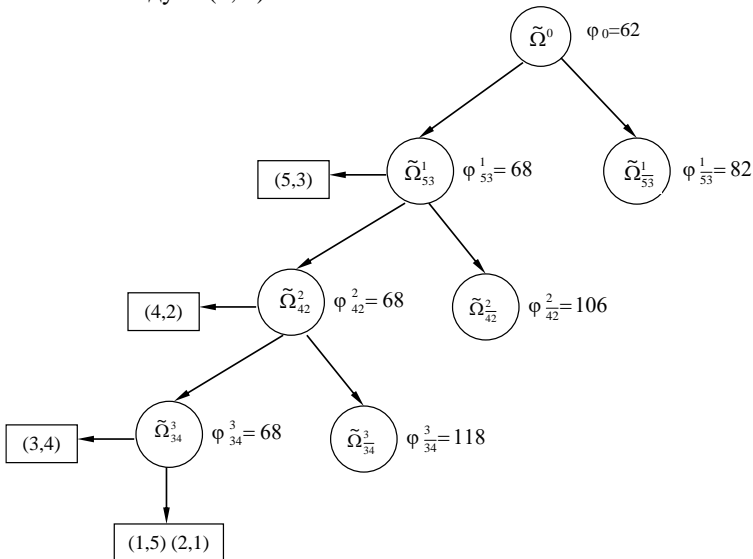


Рисунок 3.10 – Полное дерево ветвления

Так как $\varphi_{42}^2 < \varphi_{42}^2$, ветвлению подлежит подмножество $\tilde{\Omega}_{42}^2$ (см. таблицу 3.9). Вычисляем степени его нулей. Претендентом на включение в гамильтонов контур является дуга (3, 4). Разбиваем множество $\tilde{\Omega}_{42}^2$ на подмножества Ω_{34}^3 и Ω_{34}^3 . Их матрицы представлены в таблицах 3.11 и 3.12.

Таблица 3.11 – Матрица гамильтоновых контуров $\tilde{\Omega}_{34}^3$

$i \backslash j$	1	5
1	∞	0
2	0	4

Таблица 3.12 – Матрица гамильтоновых контуров $\tilde{\Omega}_{34}^3$

$i \backslash j$	1	4	5
1	∞	18	0
2	0	∞	4
3	32	∞	∞

Так как $\varphi_{34}^3 = 68 < \varphi_{34}^3 = 118$, ветвлению нужно подвергнуть подмножество $\tilde{\Omega}_{34}^3$. Но его матрица имеет размерность 2×2 . Поэтому в гамильтонов контур следует включить дуги, соответствующие в матрице подмножества $\tilde{\Omega}_{34}^3$ нулевым элементам, т. е. дуги (1, 5) и (2, 1). На рисунке 5 представлено полное дерево ветвлений. Определим полученный гамильтонов контур:

$$\Gamma_1 = \{(5,3), (4,2), (3,4), (1,5), (2,1)\} = (5 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 5).$$

Его длина

$$z(\Gamma_1) = c_{53} + c_{34} + c_{42} + c_{21} + c_{15} = 14 + 20 + 2 + 12 + 20 = 68.$$

Так как нижние границы оборванных ветвей больше длины первого реркорда $z(\Gamma^*) = 68$ (см. рисунок 3.10), этот контур имеет наименьшую длину.

На рисунке 3.11 представлен замкнутый маршрут минимальной длины.

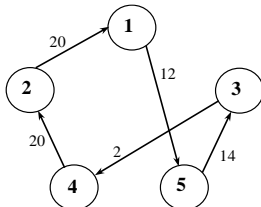


Рисунок 3.11 – Замкнутый маршрут минимальной длины

3.5 Задача о назначениях (задача выбора)

3.5.1 Решение задачи о назначениях венгерским методом

Постановка задачи о назначениях. Имеется n исполнителей A_1, A_2, \dots, A_n и n видов работ B_1, B_2, \dots, B_n . Пусть c_{ij} ($i, j = \overline{1, n}$) – параметры эффективности выполнения исполнителем A_i работы B_j (таблица 3.13).

Ставится задача отыскания такой расстановки исполнителей на работы, чтобы суммарная эффективность выполнения работ была наилучшей при условии, что каждый исполнитель может быть назначен только на одну работу и за каждой работой должен быть закреплен только один исполнитель.

Таблица 3.13 – Показатели эффективности выполнения работ

Исполнители A_i	Работы B_j			
	1	2	...	n
1	c_{11}	c_{12}	...	c_{1n}
2	c_{21}	c_{22}	...	c_{2n}
...
n	c_{n1}	c_{n2}	...	c_{nn}

Составим математическую модель задачи.

Требуется найти план назначения исполнителей на работы $X = \{x_{ij}\}$, при котором суммарная эффективность выполнения работ будет оптимальной:

$$z(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max (\min) \quad (3.11)$$

при ограничениях:

– каждый исполнитель назначается только на одну работу:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n}, \quad (3.12)$$

– каждая работа выполняется только одним исполнителем:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n}, \quad (3.13)$$

условие целочисленности

$$x_{ij} \in \{0; 1\}, \quad i, j = \overline{1, n}, \quad (3.14)$$

где переменные $x_{ij} = 1$, если j -й вид работ выполняется i -м исполнителем, $x_{ij} = 0$ – в остальных случаях.

Если задача о назначениях ставится при условии получения максимального эффекта (3.15), то ее сводят к задаче на минимум (3.16):

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max, X \in \Omega, \quad (3.15)$$

где Ω – область допустимых решений.

Пусть дана матрица эффективности $C = \| \| c_{ij} \| \|$. В каждом столбце находим максимальный элемент $l_j = \max_i c_{ij}$ и строим новую матрицу $C' = \| \| l_j - c_{ij} \| \|$.

В этом случае целевая функция имеет вид:

$$z' = \sum_{i=1}^n \sum_{j=1}^n (l_j - c_{ij}) x_{ij} \rightarrow \min, x \in \Omega, \quad (3.16)$$

$$z' = \sum_{i=1}^n \sum_{j=1}^n (l_j - c_{ij}) x_{ij} = \sum_{j=1}^n l_j \left(\sum_{i=1}^n x_{ij} \right) - \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} = \sum_{j=1}^n l_j - z.$$

Функция z' достигает минимума при условии, что z достигает максимума, $X \in \Omega$.

Задача о назначениях открытого типа. Возникает тогда, когда количество исполнителей n не равно количеству работ m . Если $m < n$, то вводят $n - m$ фиктивных работ. Считается, что с назначением на фиктивные работы исполнителей не связаны затраты, то есть соответствующие элементы матрицы стоимостей равны нулю.

В случае $m > n$, вводят $m - n$ фиктивных исполнителей. Соответствующие элементы c_{ij} стоимостей можно полагать очень большим (M). Если запрещается выполнение какой-либо работы каким-либо исполнителем, то и в этом случае соответствующий элемент c_{ij} матрицы стоимостей можно полагать очень большим (M). Таким образом, задача преобразуется к задаче закрытого типа.

Пусть, например, количество исполнителей n превышает количество работ m . Введем дополнительные фиктивные работы с индексами $\overline{j} = m + 1, n$. Коэффициенты таблицы назначений c_{ij} , $i = \overline{1, n}$, $j = \overline{m + 1, n}$ положим равными нулю. В этом случае получаем задачу закрытого типа. Если в оптимальном плане этой задачи $x_{ij} = 1$, то исполнитель i назначается на выполнение фиктивной работы, то есть остается без работы. Заметим, что оптимальное значение целевой функции исходной задачи совпадает с оптимальным значением задачи, приведенной к закрытому типу. Таким образом, эффективность назначений в результате такого преобразования не меняется.

Задача о назначениях является *частным случаем транспортной задачи*, в которой исполнители соответствуют пунктам отправления, а работы – пунктам назначения. В данном случае все величины спроса и предложения равны 1. Стоимость «транспортировки» исполнителя i на работу j равна c_{ij} . Задачу о назначениях можно эффективно решить точно так же, как и транспортную задачу. Применительно к задаче о назначениях симплексный метод не эффективен, так как любое ее допустимое базисное решение является вырожденным.

Специфические особенности задачи о назначениях, где все величины спроса и предложения равны 1, позволили разработать эффективный метод ее решения, известный как *венгерский метод*.

Если говорить о связи венгерского метода с общими методами линейного программирования, то наиболее близко он примыкает к методам, опирающимся на теорию двойственности. По существу этот метод является уточнением метода последовательного сокращения невязок применительно к транспортной задаче.

Венгерский метод решения задачи о назначениях.

Введём ряд определений: задачи целочисленного линейного программирования называются *эквивалентными*, если их оптимизирующие наборы совпадают. Преобразование, переводящее задачу целочисленного линейного программирования в эквивалентную ей, будем называть *эквивалентным*. Систему нулевых элементов матрицы, обладающую тем свойством, что никакая пара из них не лежит в одной строке или в одном столбце, будем называть *системой независимых нулей*.

Алгоритм венгерского метода состоит в преобразовании исходной задачи целочисленного линейного программирования в эквивалентную ей задачу минимизации, матрица $C = \|c_{ij}\|$, ($i, j = 1, n$) которой содержит неотрицательные элементы.

Если в матрице C имеется система из n независимых нулей, то решением задачи будет матрица X с единичными элементами на местах, соответствующих независимым нулям, и с остальными нулевыми элементами. Алгоритм венгерского метода включает один подготовительный и четыре основных этапа и не более чем $(n - 2)$ последовательно проводимых итераций. Каждая итерация связана с некоторыми эквивалентными преобразованиями матрицы, полученной в результате проведения предыдущей итерации, и выбором максимального числа независимых нулей. Окончательным результатом итерации является увеличение числа независимых нулей, имеющих в ее начале, как минимум на единицу. Как только количество независимых нулей становится равным n , проблема выбора оказывается решенной: оптимальное решение определяется системой независимых нулей в последней из матриц, эквивалентных C .

Результатом решения задачи о назначениях является матрица $X^* = \|x_{ij}\|$, ($i, j = \overline{1, n}$), компоненты которой являются целыми числами.

Оптимальный план задачи о назначениях можно представить в виде квадратной матрицы назначений, в каждой строке и в каждом столбце которой находится ровно одна единица. Такую матрицу иногда называют *матрицей перестановок*. Значение целевой функции, соответствующее оптимальному плану, называют *эффективностью назначений*.

Алгоритм решения задачи о назначениях венгерским методом.

Подготовительный этап

Если исходная задача была задачей на максимум, то она должна быть преобразована в задачу минимизации (формула (3.16)) следующим эквивалентным преобразованием матрицы C . Необходимо определить максимальный элемент в каждом столбце исходной матрицы и вычесть из него все элементы соответствующего столбца – получаем новую матрицу C' .

Основные этапы

Этап 1. Получение нулей в каждой строке. В исходной матрице стоимостей (C) или в преобразованной (C'), в зависимости от условия задачи, определяем в каждой строке минимальную стоимость и вычитаем ее из других элементов строки.

Этап 2. Получение нулей в каждом столбце. В матрице, полученной на первом этапе, найдем в каждом столбце минимальную стоимость и вычитаем ее из других элементов столбца.

Этап 3. Проверка решения на оптимальность. Оптимальным назначениям будут соответствовать нулевые элементы. Находим строку, содержащую наименьшее число нулей, выделяем один из них «*» и зачёркиваем все остальные нули этой строки и столбца, содержащих выделенный нуль. Аналогичные операции последовательно выполняем для всех строк. Если получена система независимых нулей (число выделенных нулей равно n), то решение является оптимальным, иначе переходим к следующему этапу.

Этап 4. Если после выполнения третьего этапа описанного алгоритма не получено оптимальное решение, то требуется перейти к выполнению первой итерации (решение содержит не более чем $(n - 2)$ последовательно проводимых итераций) и выполнить следующие действия:

1) *поиск минимального набора строк и столбцов, содержащих нули.* Требуется выделить (вычеркнуть) в последней матрице строки и столбцы, которые содержат все нулевые элементы так, чтобы их количество было минимальным;

2) *перестановка некоторых нулей:* находим наименьший невыделенный (не вычеркнутый) элемент в матрице и вычитаем его из остальных невыделенных элементов и прибавляем к элементам, стоящим на пересечении выделенных строк и столбцов (строим новую матрицу) и переходим к третьему этапу алгоритма.

Пример 3.4. Постановка задачи. Имеются пять исполнителей A_1, \dots, A_5 и четыре вида работ B_1, \dots, B_4 . Известна себестоимость c_{ij} ден. ед. ($i = \overline{1, 5}$; $j = \overline{1, 4}$) выполнения i -м исполнителем j -го вида работ (таблица 3.14).

Таблица 3.14 – Исходные данные

Исполнители	Работы			
	B_1	B_2	B_3	B_4
A_1	40	80	60	90
A_2	40	70	40	100
A_3	80	80	–	90
A_4	60	100	60	70
A_5	70	90	70	50

Требуется так распределить исполнителей по работам, чтобы обеспечить выполнение всех работ с минимальными суммарными затратами. Известно, что исполнитель A_3 не выполняет работу B_3 . Какой исполнитель будет простаивать?

Определим тип задачи о назначениях. Количество исполнителей равно пяти, а количество работ – четырем, следовательно, данная задача является задачей открытого типа. Преобразуем ее к задаче закрытого типа, введем фиктивную работу B_5 .

Построим математическую модель задачи. Обозначим через x_{ij} – назначение i -го исполнителя на j -й вид работ.

Математическая модель задачи примет вид:

Найти план назначений

$$X = \|x_{ij}\| = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \end{pmatrix},$$

чтобы суммарные затраты были минимальными:

$$z = \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_{ij} \rightarrow \min ,$$

при следующих ограничениях:

– каждый исполнитель назначается только на одну работу:

$$\sum_{j=1}^5 x_{ij} = 1, \quad i = \overline{1, 5};$$

– каждая работа выполняется только одним исполнителем:

$$\sum_{i=1}^5 x_{ij} = 1, \quad j = \overline{1, 5};$$

условие целочисленности:

$$x_{ij} \in \{0; 1\}, \quad i, j = \overline{1, 5},$$

где переменные $x_{ij} = 1$, если j -й вид работ выполняется i -м исполнителем, $x_{ij} = 0$ в остальных случаях.

Стоимость выполнения фиктивной работы B_5 исполнителями равна нулю, то есть элементы $c_{15}, c_{25}, c_{35}, c_{45}, c_{55}$ матрицы стоимостей равны нулю. Так как исполнитель A_3 не выполняет работу B_3 , то элемент c_{33} матрицы стоимостей можно полагать очень большим числом M .

Получим матрицу стоимостей выполнения работ:

$$C = \| c_{ij} \| = \begin{pmatrix} 40 & 80 & 60 & 90 & 0 \\ 40 & 70 & 40 & 100 & 0 \\ 80 & 80 & M & 90 & 0 \\ 60 & 100 & 60 & 70 & 0 \\ 70 & 90 & 70 & 50 & 0 \end{pmatrix}.$$

Этап 1. *Получение нулей в каждой строке.*

Находим минимальный элемент в каждой строке исходной таблицы 3.15.

Таблица 3.15 – Стоимости выполнения работ

	B_1	B_2	B_3	B_4	B_5	min
A_1	40	80	60	90	0	0
A_2	40	70	40	100	0	0
A_3	80	80	M	90	0	0
A_4	60	100	60	70	0	0
A_5	70	90	70	50	0	0

Затем вычитаем минимальные элементы из всех элементов соответствующих строк. Результаты вычислений записываем в таблицу 3.16.

Таблица 3.16 – Получение нулей в каждой строке

	B_1	B_2	B_3	B_4	B_5
A_1	40	80	60	90	0
A_2	40	70	40	100	0
A_3	80	80	M	90	0
A_4	60	100	60	70	0
A_5	70	90	70	50	0
min	40	70	40	50	0

Этап 2. *Получение нулей в каждом столбце.*

Находим минимальный элемент в каждом столбце таблицы 3.16. Вычитаем найденный элемент из каждого элемента соответствующего столбца. В результате получаем следующую таблицу 3.17.

Таблица 3.17 – Получение нулей в каждом столбце

	B_1	B_2	B_3	B_4	B_5
A_1	0	10	20	40	0
A_2	0	0	0	50	0
A_3	40	10	$M - 40$	40	0
A_4	20	30	20	20	0
A_5	30	20	30	0	0

Этап 3. *Проверка решения на оптимальность.* Находим строки, содержащие наименьшее число нулей – это строка три и четыре, выбираем одну из них, например, четвертую и выделяем один из нулей этой строки «*» и зачеркиваем все остальные нули этой строки и столбца, содержащих выделенный нуль (таблица 3.18). Следующими строками, содержащими наименьшее число нулей, являются первая и пятая, из которых выбираем одну и выполняем аналогичные операции, и так далее последовательно для всех строк.

Таблица 3.18 – Получение нулей в каждой строке

	B_1	B_2	B_3	B_4	B_5
A_1	0*	10	20	40	0
A_2	0	0*	0	50	0
A_3	40	10	$M - 40$	40	0
A_4	20	30	20	20	0*
A_5	30	20	30	0*	0

Так как количество выделенных «*» нулей равно 4 (система нулей не является независимой), следовательно, решение не оптимально.

Этап 4. *Выполнение 1 итерации.* Находим минимальный набор строк и столбцов, содержащих нули (таблица 3.19).

Таблица 3.19 – Получение минимального набора строк и столбцов, содержащих нули

	B_1	B_2	B_3	B_4	B_5
A_1	0	10	20	40	0
A_2	0	0	0	50	0
A_3	40	10	$M - 40$	40	0
A_4	20	30	20	20	0
A_5	30	20	30	0	0

Находим наименьший невыделенный элемент, который находится на пересечении третьей строки и второго столбца, равный 10 и вычитаем его из остальных невыделенных элементов и прибавляем к элементам, стоящим на пересечении выделенных строк (первая, вторая и пятая) и столбцов (пятый). В результате получаем следующую таблицу 3.20.

Таблица 3.20 – Перестановка нулей

	B_1	B_2	B_3	B_4	B_5
A_1	0	10	20	50	10
A_2	0	0	0	60	10
A_3	30	0	$M - 50$	30	0
A_4	10	20	10	10	0
A_5	30	20	30	0	10

Этап 3. Проверка решения на оптимальность (таблица 3.21).

Таблица 3.21 – Оптимальное решение

	B_1	B_2	B_3	B_4	B_5
A_1	0*	10	20	50	10
A_2	0	0	0*	60	10
A_3	30	0*	$M - 50$	30	0
A_4	10	20	10	10	0*
A_5	30	20	30	0*	10

Получена система независимых нулей (количество выделенных нулей равно 5). В результате выполнения одной итерации получено оптимальное решение.

Следовательно, оптимальный план назначения исполнителей на работы:

$$X^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Значение целевой функции при оптимальном плане

$$z^* = z(X^*) = 40 \cdot 1 + 40 \cdot 1 + 80 \cdot 1 + 0 \cdot 1 + 50 \cdot 1 = 210 \text{ ден. ед.}$$

Для того чтобы затраты были минимальными, рекомендуется придерживаться полученного оптимального плана назначений: первый исполнитель назначается на первый вид работ, второй – на третий, третий – на второй, пятый – на четвертый. Четвертый исполнитель будет простаивать. В этом случае затраты составят 210 ден. ед.

3.5.2 Применение задачи о назначениях для оптимизации расписания общественного транспорта на дублирующих участках

Методика оптимизации расписания городского общественного транспорта путем выравнивания интервалов времени между следующими друг за другом маршрутными транспортными средствами разных маршрутов на совместных (дублирующих) участках включает следующие этапы.

На *первом* этапе проводится анализ городской сети общественного транспорта и определяется множество дублирующих участков (ДУ), их протяженность, частота движения маршрутных транспортных средств разных маршрутов на ДУ.

Второй этап заключается в выборе дублирующего участка для оптимизации расписания и определении его характеристик. Определяется оптимальный интервал времени между прибытиями на базовый остановочный пункт (ОП) маршрутных транспортных средств всех маршрутов I_{Mk}^* и маршрутов дублирующего участка I_D^* определяется по формулам:

$$I_{Mk}^* = \frac{T}{N_{Mk}}, \quad (3.17)$$

$$I_D^* = \frac{T}{N_D}, \quad (3.18)$$

где T – период времени планирования; N_{Mk} – количество рейсов, выполняемых на k -м маршруте в период планирования T ; N_D – количество рейсов, выполняемых на маршрутах дублирующего участка.

На *третьем* этапе для базового остановочного пункта по дублирующему участку D для исходного расписания в период планирования T формируется двумерная булева матрица назначений (таблица 3.22). В матрице отражается назначение j -го маршрута на i -й временной интервал:

$x_{ij} = 1$, если на i -й временной интервал назначен j -й маршрут, $x_{ij} = 0$ в противном случае;

$x_{ij} = 1^*$ – обязательное назначение, которое определяет фиксированное время прибытия маршрутного транспортного средства на базовый остановочный пункт.

Для исходного расписания в период планирования определяются характеристики матрицы назначений дублирующего участка D (таблица 3.22):

– $|I_D^* - I_i|$ – величина отклонения интервалов между следующими друг за другом маршрутными транспортными средствами от оптимальной величины дублирующего участка;

– $|I_{Mk}^* - I_i|$ – величина отклонения интервалов между следующими друг за другом маршрутными транспортными средствами от оптимальной величины k -го маршрута.

Таблица 3.22 – Матрица назначений дублирующего участка D

Время при-бытия	M_1	M_2	...	M_k	K_D	$ I_D^* - I_i $	$ I_{M_1}^* - I_i $...	$ I_{M_k}^* - I_i $
t_1	x_{11}	x_{12}	...	x_{1k}	K_{D1}	$ I_D^* - I_1 $	$ I_{M_1}^* - I_1 $...	$ I_{M_k}^* - I_1 $
t_2	x_{21}	x_{22}	...	x_{2k}	K_{D2}	$ I_D^* - I_2 $	$ I_{M_1}^* - I_2 $...	$ I_{M_k}^* - I_2 $
...
t_i	x_{i1}	x_{i2}	...	x_{ik}	K_{Di}	$ I_D^* - I_i $	$ I_{M_1}^* - I_i $...	$ I_{M_k}^* - I_i $
...
t_n	x_{n1}	x_{n2}	...	x_{nk}	K_{Dn}	$ I_D^* - I_n $	$ I_{M_1}^* - I_n $...	$ I_{M_k}^* - I_n $
Сумма	$\sum_{i=1}^n x_{i1}$	$\sum_{i=1}^n x_{i2}$...	$\sum_{i=1}^n x_{ik}$	$\sum_{i=1}^n K_{Di}$	$\sum_{i=1}^n I_D^* - I_i $	$\sum_{i=1}^n I_{M_1}^* - I_i $...	$\sum_{i=1}^n I_{M_k}^* - I_i $

Рассчитывается коэффициент загрузки остановочного пункта маршрутными транспортными средствами, равный количеству транспортных средств, прибывающих на остановочный пункт:

$$K_{Di} = \sum_{j=1}^k x_{ij}. \quad (3.19)$$

На четвертом этапе для исходного расписания в плановый период определяется значение целевой функции $D^0(I)$ для дублирующего участка D :

$$D^0(I) = \sum_{i=1}^{N_D} |I_D^* - I_i^0| + \sum_{i=1}^{N_{M_1}} |I_{M_1}^* - I_i^0| + \dots + \sum_{i=1}^{N_{M_k}} |I_{M_k}^* - I_i^0| \rightarrow \min. \quad (3.20)$$

Значение целевой функции $D^0(I)$ определяется как минимизация средней величины отклонения интервалов между следующими друг за другом маршрутными транспортными средствами от оптимальной величины.

Пятый этап заключается в выравнивании интервалов времени между следующими друг за другом маршрутными транспортными средствами на дублирующем участке.

Для рассматриваемого дублирующего участка D выбирается маршрут M_l с минимальным количеством транспортных средств.

Для выбранного маршрута M_l в базовом остановочном пункте производится выравнивание расписания движения маршрутных транспортных средств путем корректировки времени прибытия t_i . При этом должны учитываться следующие ограничения:

- расписание выравненных маршрутов M_{l-1} на дублирующем участке;

- оптимальный интервал времени $I_{M_i}^*$ между прибытиями на остановочный пункт транспортных средств маршрута MD_i ;
- обязательные назначения для маршрута MD_i ;
- коэффициент загрузки остановочного пункта K_{D_i} , величина которого для дублирующего участка должна быть равна 1;
- возможность отправки транспортного средства с конечного пункта для выравненных маршрутов (сравнение времени прибытия и отправления транспортного средства с конечного остановочного пункта; проверка соблюдения режима труда и отдыха водителей).

Шестой этап состоит в определении характеристик матрицы назначений (см. таблицу 3.22) и значения целевой функции $D_r^*(I)$ дублирующего участка D_r для планового периода T после оптимизации:

$$D_r^*(I) = \sum_{i=1}^{N_D} |I_{D_r}^* - I_i| + \sum_{i=1}^{N_{M1}} |I_{MD1}^* - I_i| + \dots + \sum_{i=1}^{N_{Mk}} |I_{MDk}^* - I_i| \rightarrow \min. \quad (3.21)$$

Исходя из минимальной величины $D_r^*(I)$ и ограничений, производится перебор матриц назначений и выбирается оптимальное расписание прибытия маршрутных транспортных средств рассматриваемого дублирующего участка D_r на базовый остановочный пункт SB_r . Далее производится расчет времени движения через остальные остановочные пункты маршрута относительно базового пункта и переход к следующему дублирующему участку.

На последнем этапе производится оценка эффективности оптимизации расписания по времени ожидания пассажирами прибытия маршрутных транспортных средств и равномерности их движения на дублирующих участках, а также эффективность оптимизации расписания движения общественного транспорта на дублирующих участках в целом.

Из матриц назначений для каждого планового периода T по дублирующим участкам определяются:

- количество периодов времени, для которых на остановочный пункт прибывают два и более транспортных средств дублирующего участка ($K_D \geq 2$);
- время ожидания пассажирами транспортных средств маршрутов дублирующего участка T_W ;
- величины отклонения интервалов между следующими друг за другом маршрутными транспортными средствами от оптимальной величины для дублирующих участков до оптимизации $D_r^0(I)$ и после оптимизации $D_r^*(I)$.

Эффективность оптимизации расписания движения общественного транспорта на дублирующих участках:

$$F_r^* = D_r^0(I) - D_r^*(I). \quad (3.22)$$

Приведенная методика оптимизации расписания общественного транспорта на дублирующих участках апробирована на существующей транспортной сети города Гомеля. В настоящее время в городе Гомеле перевозка пассажиров осуществляется по 83 регулярным автобусным маршрутам. Было выявлено шесть протяженных дублирующих участков, на которых предусмотрено движение автобусов трех и более маршрутов (таблица 3.23).

Таблица 3.23 – Параметры дублирующих участков

Дублирующие участки	Маршруты	Количество ОП	Сумма количества маршрутов и ОП	Длина ДУ, км
D_1 – «Институт „Гомельпроект” – Улица Огоренко»	№ 17, 18, 34	13	$3 + 13 = 16$	7,5
D_2 – «Вокзал – Первая школа»	№ 35, 55, 58	12	$3 + 12 = 15$	8,47
D_3 – «Вокзал – Горэлектротранспорт»	№ 10, 19, 43	10	$3 + 10 = 13$	4,94
D_4 – «Вокзал – Кинотеатр „Октябрь”»	№ 20, 21, 40, 52	8	$4 + 8 = 12$	4,47
D_5 – «Медгородок – Технический университет им. П. О. Сухого»	№ 16, 17, 26, 33	8	$4 + 8 = 12$	3,39
D_6 – «Вокзал – Дворец культуры „Гомсельмаш”»	№ 6, 8, 8А, 9	7	$4 + 7 = 11$	3,93

Дублирующие участки были ранжированы в порядке убывания суммы остановочных пунктов и количества маршрутов.

Первый дублирующий участок D_1 – «Институт „Гомельпроект” – Улица Огоренко» является общим для маршрутов № 17, № 18 и № 34 на протяжении 13 остановочных пунктов, его длина – 7,5 км. Данный совместный отрезок пути является одним из важнейших в городе Гомеле, так как проходит через наиболее загруженные транспортным потоком улицы в Центральном районе города (Советская и Интернациональная), в их пределах на остановочных пунктах формируется большое количество ожидающих пассажиров, также часто возникают задержки общественного транспорта по причине вынужденной остановки для ожидания возможности подъезда к остановочному пункту из-за одновременного прихода нескольких маршрутных транспортных средств различных маршрутов.

В качестве примера рассмотрим оптимизацию расписания движения на дублирующем участке D_1 «Институт „Гомельпроект” – Улица Огоренко» в час пик для промежутка времени с 7.00 до 8.00 часов.

Рассчитаем основные характеристики для промежутка времени с 7.00 до 8.00 часов по остановочному пункту «Институт „Гомельпроект”»:

– оптимальный интервал времени между прибытиями на остановочный пункт маршрутных транспортных средств дублирующего участка

$$I_{D1}^* = \frac{60}{8+4+6} = 3,33 \text{ мин};$$

– оптимальный интервал времени между прибытиями на остановочный пункт транспортных средств маршрутов № 17, № 18 и № 34:

$$I_{17}^* = 7,5 \text{ мин}; I_{18}^* = 15 \text{ мин}; I_{34}^* = 10 \text{ мин}.$$

По рассчитанному оптимальному интервалу времени между прибытиями на остановочный пункт маршрутных транспортных средств первого дублирующего участка, который равен 3,33 минуты, сдвигаем время прибытия автобусов № 17, № 18 и № 34 таким образом, чтобы разница между их оптимальным и реальным значением была минимальной. При этом учитываются интервалы движения между автобусами одного маршрута для исключения возможности ухудшения уже существующего расписания.

Матрицы назначений для первого дублирующего участка «Институт „Гомельпроект” – Улица Огоренко» в пиковый период времени с 7.00 до 8.00 часов представлены в таблицах 3.24, 3.25.

Таблица 3.24 – Матрица назначений для дублирующего участка D_1 при существующем расписании

Время	D_1 – «Институт „Гомельпроект” – Улица Огоренко»								
	№ 17	№ 18	№ 34	K_{D1}	$ I_{D1}^* - I_i $	T_w	$ I_{17}^* - I_i $	$ I_{18}^* - I_i $	$ I_{34}^* - I_i $
7:00	0	0	1	1	–	–			1
7:04	1	0	0	1	0,67	50	0		
7:09	0	1	0	1	1,67	75		2	
7:11	1	0	0	1	1,33	15	0		
7:12	0	0	1	1	2,33	5			2
7:18	1	0	0	1	2,67	105	0		
7:22	0	1	0	1	0,67	50		2	
7:24	0	0	1	1	1,33	15			2
7:25	1	0	0	1	2,33	5	0		
7:35	1	0	0	1	6,67	275	3		
7:36	0	1	1	2	2,33	5		1	2
7:42	1	0	0	1	2,67	105	0		
7:47	0	0	1	1	1,67	75			1
7:49	1	1	0	2	1,33	15	0	2	
7:56	1	0	0	1	3,67	140	0		
7:59	0	0	1	1	0,33	30			2
Сум- ма	8	4	6	18	31,67	965	3	7	10

Таблица 3.25 – Матрица назначений для дублирующего участка D_1 при оптимизированном расписании

Время	D_1 – «Институт «Гомельпроект» – Улица Огоренко»								
	№ 17	№ 18	№ 34	K_{D1}	$ I_{D1}^* - I_i $	T_w	$ I_{17}^* - I_i $	$ I_{18}^* - I_i $	$ I_{34}^* - I_i $
7:00	0	0	1	1	–	–			1
7:03	1	0	0	1	0,33	30	0		
7:07	0	1	0	1	0,67	50		2	
7:10	1	0	0	1	0,33	30	0		
7:13	0	0	1	1	0,33	30			3
7:17	1	0	0	1	0,67	50	0		
7:20	0	1	0	1	0,33	30		2	
7:23	0	0	1	1	0,33	30			0
7:25	1	0	0	1	1,33	15	1		
7:31	1	0	0	1	2,67	105	1		
7:33	0	0	1	1	1,33	15			0
7:36	0	1	0	1	0,33	30		1	
7:39	1	0	0	1	0,33	30	1		
7:43	0	0	1	1	0,67	50			0
7:46	1	0	0	1	0,33	30	0		
7:49	0	1	0	1	0,33	30		2	
7:53	1	0	0	1	0,67	50	0		
7:56	0	0	1	1	0,33	30			3
Сум-ма	8	4	6	18	11,31	635	3	7	7

В результате оптимизации расписания для периода времени с 7.00 до 8.00 часов на первом дублирующем участке суммарная величина отклонения интервалов между следующими друг за другом автобусами от оптимальной величины снизилась с 32 до 11 минут. Суммарное время ожидания пассажирами транспортных средств сократилось на 30 %.

Проведена оптимизация расписания движения автобусов на шести дублирующих участках в час пик для промежутка времени с 7.00 до 8.00 часов. Оценка эффективности скорректированного расписания по шести дублирующим участкам представлена в таблице 3.26.

Величина эффективности оптимизации расписания движения автобусов по шести дублирующим участкам для промежутка времени с 7.00 до 8.00 часов равна 147 минут.

По результатам проведенной оптимизации можно сделать следующие выводы:

1 Скорректированы интервалы движения для каждого маршрута в отдельности по дублирующим участкам.

Таблица 3.26 – Результаты оптимизации расписания для промежутка времени с 7.00 до 8.00

ДУ	До оптимизации			После оптимизации			$F_r^*(I)$
	$ I_D^* - I_i $	$ I_{MD}^* - I_i $	$D_r^0(I)$	$ I_D^* - I_i $	$ I_{MD}^* - I_i $	$D_r^*(I)$	
D_1	32	20	52	11	17	28	24
D_2	24	34	58	9	28	37	21
D_3	34	38	72	9	33	42	30
D_4	39	32	71	4	29	33	38
D_5	31	44	75	13	45	58	17
D_6	15	27	42	2	23	25	17
Сум- ма	175	195	370	48	175	223	147

2 Величина отклонения интервалов между следующими друг за другом автобусами от оптимальной величины уменьшилась в среднем на 26 % для промежутка времени с 7.00 до 8.00 часов.

3 Увеличена равномерность движения следующих друг за другом автобусов разных маршрутов на дублирующих участках.

4 Величина отклонения интервалов между следующими друг за другом маршрутными транспортными средствами от оптимальной величины уменьшилась в среднем на 36 %.

5 В итоге суммарная величина отклонения интервалов между следующими друг за другом автобусами от оптимальной величины по шести дублирующим участкам уменьшилась в среднем на 40 %.

6 Сократилось время ожидания маршрутных транспортных средства теми пассажирами, перевозка которых возможна несколькими вариантами маршрутов. При этом время ожидания пассажирами транспортных средств сократилось на 28 %.

4 ЗАДАЧИ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

4.1 Постановка задачи нелинейного программирования.

Виды экстремумов

При решении многих прикладных задач наряду с линейными зависимостями между факторами и показателями часто встречаются нелинейные зависимости, в особенности нелинейные целевые функции. В некоторых случаях такие задачи удастся свести к линейным, отказавшись от учета мало значащих факторов или введя дополнительные переменные или ограничения. Однако в ряде случаев избежать нелинейности целевой функции или ограничений не удастся.

Задачи, в которых либо целевая функция, либо ограничения, либо и то, и другое нелинейны, называются задачами *нелинейного программирования*.

В общем виде задача нелинейного программирования формулируется следующим образом.

Найти план $X = (x_1, x_2, \dots, x_n)$, который доставляет экстремум функции

$$z = f(X) \rightarrow \max (\min) \quad (4.1)$$

при системе ограничений

$$\varphi_i(X) \leq b_i, \quad i = \overline{1, m_1}, \quad (4.2)$$

$$\varphi_i(X) \geq b_i, \quad i = \overline{m_1 + 1, m_2}, \quad (4.3)$$

$$\varphi_i(X) = b_i, \quad i = \overline{m_2 + 1, m}, \quad (4.4)$$

где хотя бы одна из функций $f(X)$ или $\varphi_i(X)$ – нелинейны.

На некоторые или все переменные налагается условие неотрицательности.

Функция (4.1) называется *целевой функцией*, а неравенства (4.2)–(4.4) называются *ограничениями задачи*.

Множество точек X , удовлетворяющих системе ограничений (4.2)–(4.4), называется *множеством допустимых решений* Ω , $X \in \Omega$.

В зависимости от наличия или отсутствия ограничений, налагаемых на переменные, а также особенностей целевой функции, различают:

- безусловный абсолютный (глобальный) максимум (минимум);
- безусловный относительный (локальный) максимум (минимум);
- условный абсолютный максимум (минимум);
- условный относительный максимум (минимум).

Оптимальным решением задачи (4.1)–(4.4), или точкой *глобального максимума (минимума)*, называется точка $X^* \in \Omega$, если $f(X^*) \geq f(X)$ ($f(X^*) \leq f(X)$) для всех X , удовлетворяющих системе ограничений (4.2)–(4.4).

Локальным решением задачи (4.1)–(4.4), или точкой *локального максимума (минимума)*, называется точка $X^* \in \Omega$, если существует некоторое число $\varepsilon > 0$, такое, что для всех $X \neq X^*$, удовлетворяющих системе ограничений (4.2)–(4.4), и $\|X - X^*\| \leq \varepsilon$, где $\|X - X^*\| = \sqrt{\sum_{j=1}^n (x_j - x_j^*)^2}$ выполняется

$f(X^*) \geq f(X)$ ($f(X^*) \leq f(X)$).

Отметим, что точка глобального максимума (минимума) является и точкой локального максимума (минимума).

На рисунке 4.1 приведен график нелинейной функции. На отрезке $[x_0, x_6]$ точки x_1, x_3 и x_5 являются точками локального максимума, точка x_3 является точкой глобального максимума, точки x_0, x_2, x_4 и x_6 являются точками локального минимума, точка x_4 является точкой глобального минимума.

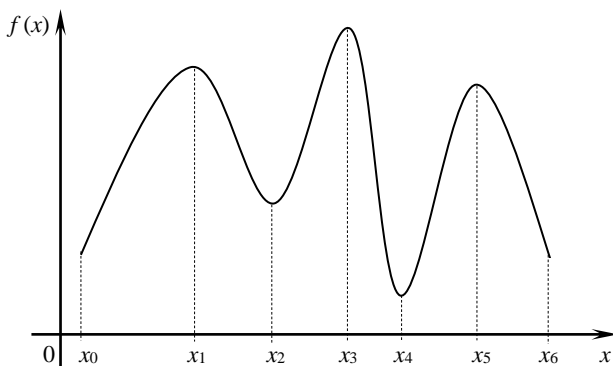


Рисунок 4.1 – График нелинейной функции с глобальными и локальными экстремумами

Если в исходной задаче нелинейного программирования отсутствуют ограничения, то речь идет о задаче *безусловной оптимизации*.

В этом случае точка X^* будет являться точкой *безусловного глобального максимума (минимума)*, если $f(X^*) \geq f(X)$ ($f(X^*) \leq f(X)$) для любых X .

Если неравенство $f(X^*) \geq f(X)$ ($f(X^*) \leq f(X)$) выполняется не для всех X , а только для точек, лежащих в ε -окрестности точки X^* ($\varepsilon > 0$), то говорят о *безусловном локальном максимуме (минимуме)*.

В теории нелинейного программирования особый класс задач составляют задачи, модели которых содержат выпуклые функции.

Функция $f(X)$, определенная на выпуклом множестве Ω , называется *выпуклой вниз*, если для любых точек X' и X'' из этого множества и любого $0 \leq \lambda \leq 1$ выполняется соотношение

$$f(\lambda X' + (1-\lambda)X'') \leq \lambda f(X') + (1-\lambda)f(X''). \quad (4.5)$$

Функция $f(X)$, определенная на выпуклом множестве Ω , называется *выпуклой вверх*, если для любых точек X' и X'' из этого множества и любого $0 \leq \lambda \leq 1$ выполняется соотношение

$$f(\lambda X' + (1-\lambda)X'') \geq \lambda f(X') + (1-\lambda)f(X''). \quad (4.6)$$

Если в соотношениях (4.5) и (4.6) при $0 < \lambda < 1$ и любых допустимых X' и X'' , $X' \neq X''$ имеет место строгое неравенство, то $f(X)$ будет соответственно *строго выпуклой вниз* (*строго выпуклой вверх*).

Геометрическая иллюстрация данных определений для функции одной переменной приведена на рисунках 4.2 и 4.3.

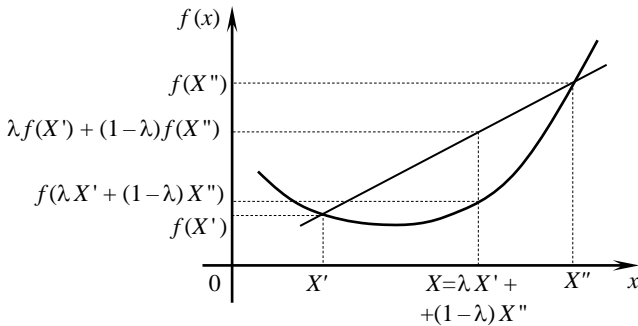


Рисунок 4.2 – График выпуклой вниз функции

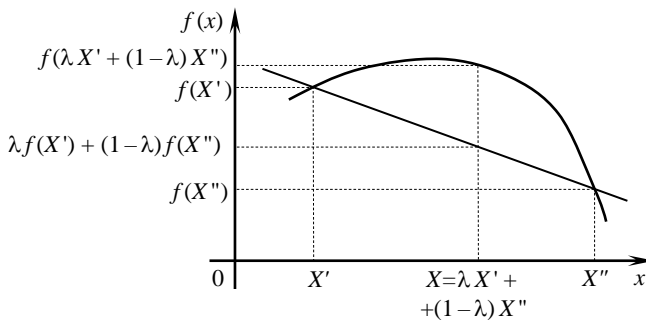


Рисунок 4.3 – График выпуклой вверх функции

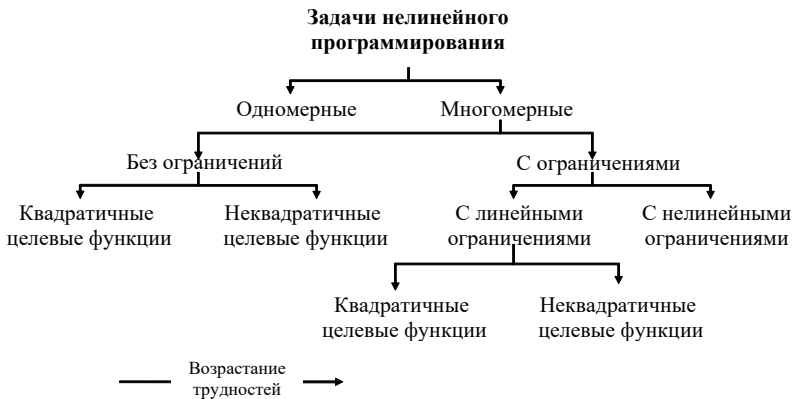
Выпуклая вниз функция $f(X)$ (рисунок 4.2) на отрезке $[X'; X'']$ не может принимать больших значений, чем линейная функция, интерполирующая значения $f(X')$ и $f(X'')$. В свою очередь выпуклая вверх функция $f(X)$ (рисунок 4.3) на отрезке $[X'; X'']$ не может принимать меньших значений, чем линейная функция, интерполирующая значения $f(X')$ и $f(X'')$.

Раздел нелинейного программирования, в котором целевая функция $f(X)$ и функции ограничений $\varphi_i(X)$ являются выпуклыми, называется *выпуклым программированием*.

В теории выпуклого программирования в качестве основной рассматривается задача минимизации выпуклой вниз функции n переменных $z = f(X)$ при ограничениях $\varphi_i(X) \leq 0$ ($i = \overline{1, m}$), $X \geq 0$, где функции $\varphi_i(X)$ предполагаются выпуклыми на выпуклом множестве допустимых решений Ω . Тогда любая точка локального минимума является и точкой глобального минимума.

Если $f(X)$ и $\varphi_i(X)$ являются выпуклыми вверх функциями, то имеем задачу максимизации $f(X)$ при ограничениях $\varphi_i(X) \geq 0$ ($i = \overline{1, m}$), $X \geq 0$. Тогда любая точка локального максимума является и точкой глобального максимума.

Трудности, связанные с решением задач нелинейной оптимизации, зависят от вида функций $f(X)$ и $\varphi_i(X)$, которые могут быть линейными, квадратичными или более сложной структуры и зависеть от одной, двух и более переменных. Условно можно выделить следующие классы задач нелинейного программирования:



Степень сложности задач в приведенной классификации очень разная. Самыми простыми для исследователя являются одномерные задачи нелинейной оптимизации и наиболее сложными – многомерные задачи с нелинейными ограничениями. Такое разнообразие классов задач породило и разнообразие методов решения.

4.2 Одномерная оптимизация

Под *одномерной оптимизацией* понимается задача поиска экстремума у функции одной переменной $f(x)$.

Функция называется *унимодальной* на отрезке $[a, b]$, если на этом отрезке она имеет одну точку экстремума.

Далее будут рассматриваться методы поиска экстремума унимодальных функций. Если это требование не выполняется, то после построения графика целевой функции f можно выделить отрезки (рисунок 4.4), на каждом из которых существует только один минимум (максимум) $f(x)$ и затем применить метод для нахождения экстремума $f(x)$ на выбранном отрезке.

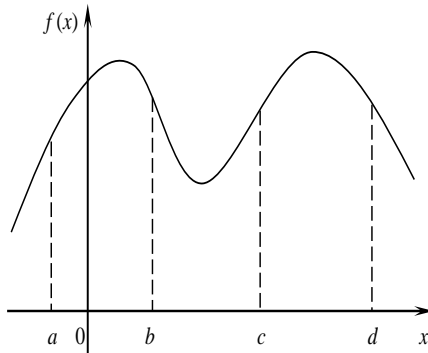


Рисунок 4.4 – График нелинейной целевой функции с локальными экстремумами

Для решения задачи одномерной оптимизации могут применяться следующие методы: аналитические, численные, методы случайного поиска.

Применение *аналитических методов* возможно в случае дифференцируемости функции $f(x)$ и заключается в исследовании уравнения $f'(x) = 0$.

Пусть целевая функция f – гладкая и непрерывная функция. Можно воспользоваться классическим методом нахождения экстремумов функций: решить уравнение $f'(x) = 0$.

В результате будут получены стационарные точки, и, чтобы выделить среди них точки экстремума, необходимо для всех стационарных точек вычислить $f''(x)$.

Если есть точка x^* , для которой $f'(x^*) = 0$ и $f''(x^*) < 0$, то в точке x^* функция $f(x)$ имеет максимум, если же $f''(x^*) > 0$, то функция $f(x)$ имеет в точке x^* минимум.

Если получают, что $f''(x^*) = 0$, то для решения вопроса о существовании экстремума в точке x^* необходимо рассматривать четные производные более высоких порядков.

Реализация численного метода или метода случайного поиска позволяет найти что-то одно: или максимум, или минимум.

Среди *численных* методов находят применение следующие: метод дихотомии (половинного деления); метод «золотого сечения»; метод Фибоначчи; шаговые и др.

Метод дихотомии. Пусть целевая функция $f(x)$ имеет на отрезке $[a, b]$ минимум в точке x^* (рисунок 4.6).

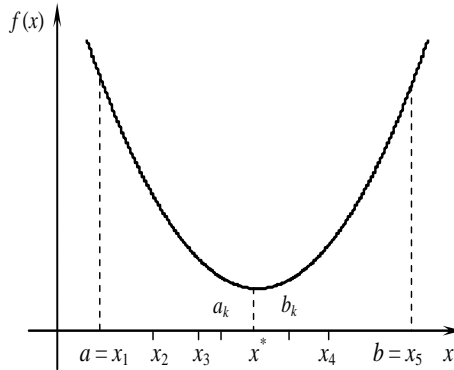


Рисунок 4.6 – Пример функции с одним минимумом

Вычисление абсциссы x^* точки минимума $f(x)$ фактически сводится к вычислению отрезка $[a_n, b_n]$, длина которого $|b_n - a_n| < \varepsilon$, ε – заданная точность вычисления и $x \in [a_n, b_n]$.

Сущность метода дихотомии заключается в следующем:

1) полагаем $x_1 = a$ и $x_5 = b$. Делим отрезок $[a, b]$ точками x_2, x_3 и x_4 на четыре равные части;

2) вычисляем значения функции $f(x)$: $f(x_1), f(x_2), f(x_3), f(x_4), f(x_5)$;

3) среди чисел $f(x_i)$ находим минимальные.

Пусть, например, $f(x_k) = \min \{f(x_i), i = \overline{1, 5}\}$. В силу унимодальности $f(x)$ на отрезке $[a, b]$ точка x^* находится на отрезке $[x_{k-1}, x_{k+1}]$, длина которого по построению вдвое меньше длины отрезка $[a, b]$.

Если всё-таки $|x_{k+1} - x_k| > \varepsilon$, то полагаем $a = x_{k-1}, b = x_{k+1}$ и вновь выполняем все действия, начиная с первого действия.

Если на n -й итерации получено, что локализованный отрезок $|x_{k+1} - x_k| \leq \varepsilon$, то в качестве значения x^* можно, например, принять

$$x^* = \frac{x_{k-1} + x_{k+1}}{2}.$$

Метод «золотого сечения» основан на делении отрезка $[a;b]$ по правилу золотого сечения – когда отношение большего отрезка к меньшему является const.

Прежде всего напомним, что если d – длина отрезка, то золотое сечение отрезка определяется пропорцией $\frac{d}{x} = \frac{x}{d-x}$, т. е. x находится из квадратного уравнения $x^2 + dx - d^2 = 0$, и следовательно, $x_1 \approx 0,38(b-a)$ и $x_2 \approx 0,62(b-a)$ (рисунок 4.7).

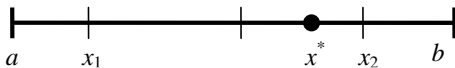


Рисунок 4.7 – Пример выбора точки деления отрезка по методу «золотого сечения»

Пусть целевая функция $f(x)$ имеет на отрезке $[a, b]$ единственный минимум в точке x^* . Необходимо найти абсциссу точки x^* с заданной точностью $\varepsilon = 1$. Метод «золотого сечения» предписывает выполнение следующих действий:

- 1) вычисляем $x_1 = a + 0,38(b-a)$ и $x_2 = a - 0,62(b-a)$.
- 2) вычисляем $f(x_1)$ и $f(x_2)$. При этом могут иметь место следующие случаи:

$x_2 > x_1$ и $f(x_1) > f(x_2)$. В этом случае $x^* \in [x_1, b]$;

$x_2 > x_1$ и $f(x_1) < f(x_2)$. В этом случае $x^* \in [a, x_2]$;

$x_2 > x_1$ и $f(x_1) = f(x_2)$. В этом случае $x^* \in [x_1, x_2]$;

- 3) выбираем отрезок, которому принадлежит точка x^* . Назовём его $[a_1, b_1]$. Длина этого отрезка меньше длины исходного отрезка $[a, b]$, но если только $|b_1 - a_1| \leq \varepsilon$, мы можем считать, что с точностью ε

$$x^* = \frac{b_1 + a_1}{2}.$$

В противном случае действия 1–3 повторяются до тех пор, пока на n -м шаге не будет локализован отрезок $[a_n, b_n]$, для которого будет выполнено требование $|b_n - a_n| \leq \varepsilon$.

Метод Фибоначчи основан на делении отрезка $[a;b]$ с использованием чисел Фибоначчи (последовательность чисел, у которых каждое следующее число равно сумме двух предыдущих).

Пусть F_n – n -е число в последовательности Фибоначчи.

Известно, что предел

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \varphi$$

является пропорцией золотого сечения, поэтому метод золотого сечения может быть трансформирован в метод Фибоначчи, при этом количество итераций в данном методе строго ограничено, а следовательно, задано количество возможных к выполнению функций $f(x)$. По этому методу начальные точки рассчитываются следующим образом:

$$x_1 = a + (b - a) \frac{F_{n-2}}{F_n};$$

$$x_2 = a + (b - a) \frac{F_{n-1}}{F_n}.$$

Далее поиск экстремума выполняется аналогично методу золотого сечения.

Методы случайного поиска основаны на выборе случайным образом некоторых точек из выбранного отрезка и нахождении точки, соответствующей экстремуму целевой функции. Точность определяется числом точек поиска n .

Если ε – точность, то вероятность нахождения при одном испытании при выборе одной точки –

$$P_{\geq 1} = \frac{\varepsilon}{b - a}.$$

Тогда вероятность получения решения хотя бы один раз при выборе n точек будет равна: $P_{\geq 1} = 1 - (1 - P)^n$.

4.3 Многомерная оптимизация

4.3.1 Методы оптимизации функций нескольких переменных

Задачи многомерной оптимизации заключаются в поиске экстремума функции нескольких переменных $z = f(x_1, x_2, x_3, \dots, x_n)$.

Так же как и в случае одномерной оптимизации, выделяют следующие методы многомерной оптимизации: аналитические методы, численные методы и методы случайного поиска.

Рассмотрим некоторые методы безусловной многомерной оптимизации.

Аналитический метод основывается на исследовании частных производных первого и второго порядка. Необходимым условием существования экстремума в точке $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ является равенство 0 в точке X^* частных производных первого порядка:

$$\left\{ \begin{array}{l} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} = 0; \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} = 0; \\ \dots \dots \dots \dots \dots \dots \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} = 0. \end{array} \right.$$

В общем случае решение указанной системы может быть как точкой минимума, так и точкой максимума, а также может не являться точкой экстремума.

Достаточным условием существования экстремума функции является знак второго дифференциала.

Для того чтобы функция $f(x_1, x_2, x_3, \dots, x_n)$ в точке X^* имела максимум, необходимо, чтобы $d^2 f(X^*) < 0$. Для существования минимума достаточно, чтобы $d^2 f(X^*) > 0$.

Для случаев двух переменных второй дифференциал равен:

$$d^2 f(X) = \frac{\partial^2 f}{\partial x_1^2} \cdot dx_1^2 + 2 \frac{\partial^2 f}{\partial x_1 \partial x_2} \cdot dx_1 dx_2 + \frac{\partial^2 f}{\partial x_2^2} \cdot dx_2^2.$$

Определить наличие экстремума в случае трех и более переменных более удобно с помощью критерия Сильвестра. Для этого рассматриваются матрицы:

$$\Delta_2 = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{vmatrix};$$

$$\Delta_3 = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{vmatrix}.$$

Аналогично для случая четырех и более переменных.

Если указанный определитель $\Delta > 0$ в точке X^* , то функция f в точке X^* имеет минимум, а если $\Delta < 0$ – максимум.

Пример 4.1. Исследовать на экстремум функцию:

$$f(X) = x_1^2 + 3x_1 + 2x_2^2 + x_2.$$

Найдем точки экстремума этой функции. Для этого найдем частные производные и приравняем их к 0.

$$\begin{cases} \frac{\partial f}{\partial x_1} = 2x_1 + 3 = 0; \\ \frac{\partial f}{\partial x_2} = 4x_2 + 1 = 0. \end{cases}$$

Решая систему, найдем:

$$x_1^* = -\frac{3}{2}, \quad x_2^* = -\frac{1}{4}.$$

Исследуем знак второго дифференциала в точке $X^* = \left(-\frac{3}{2}; -\frac{1}{4}\right)$:

$$\frac{\partial^2 f}{\partial x_1^2} = 2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 0; \quad \frac{\partial^2 f}{\partial x_2^2} = 4.$$

$$d^2 f(X^*) = 2dx_1^2 + 4dx_2^2 > 0.$$

Так как второй дифференциал $d^2 z$ в точке X^* больше 0, то точка X^* является точкой минимума.

Определим также наличие экстремума функции с помощью критерия Сильвестра:

$$\Delta_2 = \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = 8 > 0.$$

Так как определитель $\Delta_2 > 0$, то в точке $X^* = \left(-\frac{3}{2}; -\frac{1}{4}\right)$ функция f имеет минимум. Минимальное значение функции $f(X^*) = -2,375$.

Среди *численных* методов многомерной оптимизации различают методы нулевого порядка и градиентные (первого и второго порядка).

Методы нулевого порядка основаны на вычислении только значений исследуемой функции.

Среди методов нулевого порядка наиболее известны следующие: метод координатного спуска – поочередная оптимизация параметров вдоль осей координат одним из известных одномерных методов; метод спирального координатного спуска; метод Розенброка; метод Пауэлла и др.

Градиентные методы используют частные производные соответствующего порядка.

4.3.2 Особенности задач нелинейного программирования и графический метод решения

Решение задачи (4.1)–(4.4) сопряжено с определёнными трудностями, порождаемыми нелинейностью функций $f(X)$ и $\varphi_i(X)$.

Чтобы понять, в чем заключаются эти трудности, сопоставим особенности задач линейного и нелинейного программирования.

Рассмотрим задачу нелинейного программирования, содержащую две переменные и записанную в следующем виде:

$$z = f(x_1, x_2) \rightarrow \max (\min); \quad (4.7)$$

$$\varphi_1(x_1, x_2) \leq b_1,$$

$$\varphi_2(x_1, x_2) \leq b_2,$$

...

$$\varphi_m(x_1, x_2) \leq b_m,$$

$$x_1 \geq 0, \quad x_2 \geq 0. \quad (4.8)$$

Как и в случае линейного программирования, для задач нелинейного программирования, содержащих только две переменные, возможна геометрическая интерпретация.

Наиболее существенные отличия задач нелинейного программирования от линейных заключаются в том, что в задачах нелинейного программирования множество допустимых решений Ω может быть невыпуклым, несвязным и иметь бесконечное число крайних точек, а целевая функция может достигать экстремума не только на границе, но и внутри области допустимых решений.

В то время как в задачах линейного программирования область допустимых решений Ω – это выпуклое множество с конечным числом угловых точек, а экстремальное значение линейной целевой функции достигается в одной или нескольких угловых точек области допустимых решений. Кроме того, нелинейная целевая функция может иметь несколько локальных экстремумов, среди которых необходимо найти глобальное решение. В задачах же линейного программирования локальный экстремум целевой функции всегда совпадает с глобальным.

Сформулируем алгоритм **графического метода решения задачи нелинейного программирования** в случае двух переменных:

1 С учетом системы ограничений (4.8) строим множество (область) допустимых решений Ω . Если множество допустимых решений Ω пусто, то задача не имеет решений.

2 Строим линии уровня целевой функции $f(x_1, x_2) = C$ (C – const) при различных значениях параметра C .

3 Определяем направление возрастания (для задачи на максимум) или убывания (для задачи на минимум) линий уровня целевой функции.

4 Определяем точку области допустимых решений, через которую проходит линия уровня целевой функции с максимальным (для задачи на максимум) или минимальным (для задачи на минимум) значением параметра C . Если целевая функция не ограничена сверху (для задачи на максимум) или не ограничена снизу (для задачи на минимум) на множестве допустимых решений Ω , то задача не имеет решений.

5 Определяем оптимальное решение $X^* = (x_1^*; x_2^*)$ и экстремальное значение целевой функции в данной точке $z^* = f(x_1^*; x_2^*)$.

Рассмотрим пример решения задачи нелинейного программирования графическим методом.

Пример 4.2. Графическим методом найти экстремумы функции при заданных ограничениях

$$\begin{aligned}z &= f(x_1, x_2) = x_1^2 + x_2^2 + 4x_1 - 2x_2 \rightarrow \max (\min), \\x_1^2 + x_2^2 + 2x_1 - 14x_2 &\geq -34, \\x_1^2 + x_2^2 - 4x_1 - 6x_2 &\leq 23, \\x_1 + 2x_2 &\geq 10, \\x_1 \geq 0, \quad x_2 &\geq 0.\end{aligned}$$

С учетом системы ограничений построим множество допустимых решений Ω (рисунок 4.8), предварительно выделив полные квадраты относительно переменных x_1 и x_2 в первых двух ограничениях:

$$\begin{aligned}x_1^2 + x_2^2 + 2x_1 - 14x_2 &= (x_1 + 1)^2 + (x_2 - 7)^2 \geq 16, \\x_1^2 + x_2^2 - 4x_1 - 6x_2 &= (x_1 - 2)^2 + (x_2 - 3)^2 \leq 36.\end{aligned}$$

Видим, что область допустимых решений (на рисунке 4.8 она заштрихована) ограничена двумя окружностями, заданными уравнениями:

$$(x_1 + 1)^2 + (x_2 - 7)^2 = 16 \quad \text{и} \quad (x_1 - 2)^2 + (x_2 - 3)^2 = 36,$$

с центрами в точках $O_1(-1; 7)$ и $O_2(2; 3)$ и радиусами, равными, соответственно, 4 и 6, и прямой $x_1 + 2x_2 = 10$.

Строим линии уровня целевой функции $f(x_1, x_2) = C$ ($C > 0$) при различных значениях параметра C , предварительно преобразовав целевую функцию:

$$f(x_1, x_2) = (x_1 + 2)^2 + (x_2 - 1)^2 - 5 = C.$$

Линиями уровня целевой функции будут окружности с центром в точке $O_3(-2; 1)$ и радиусом $\sqrt{C+5}$. Изменяя значение C , будем получать окружности различных радиусов, а следовательно, и различные значения функции, причем чем больше радиус окружности, тем больше значение целевой функции $f(x_1, x_2)$.

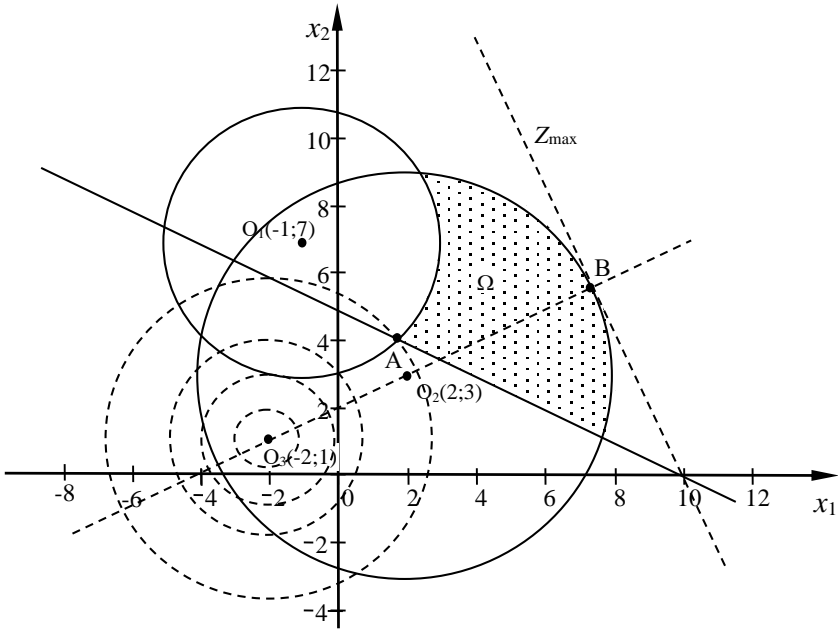


Рисунок 4.8 – Графическое решение задачи

Описав вокруг точки O_3 окружности различных радиусов, определяем, что в точке A функция достигает минимального значения.

Координаты точки A находим путем решения системы уравнений:

$$\begin{cases} x_1^2 + x_2^2 + 2x_1 - 14x_2 = -34, \\ x_1 + 2x_2 = 10. \end{cases}$$

Данная система уравнений имеет два решения, причем одно из решений не удовлетворяет условиям неотрицательности, которое налагается на переменные x_1 и x_2 .

Оптимальное решение, удовлетворяющее системе ограничений:

$$X^*_{\min} = (1,77; 4,12).$$

Вычислим минимальное значение целевой функции $z_{\min} = z(X^*) = 18,92$.

В точке B функция достигает максимального значения. Для нахождения координат точки B воспользуемся тем фактом, что точки O_2 , O_3 и B лежат на одной прямой. Поэтому, составив уравнение прямой O_2O_3 , координаты точки B найдем из системы уравнений этой прямой и уравнения окружности:

$$\begin{cases} (x_1 - 3)^2 + (x_2 - 6)^2 = 16, \\ 0,143x_1 + x_2 = 6,429. \end{cases}$$

Система уравнений имеет два решения, причем одно из решений не удовлетворяет условиям неотрицательности, которое налагается на переменные x_1 и x_2 .

Оптимальное решение, удовлетворяющее системе ограничений:

$$X^*_{\max} = (6,96; 5,44).$$

Вычислим максимальное значение целевой функции $z_{\max} = z(X^*) = 115,57$.

Итак,

$$X^*_{\min} = (1,77; 4,12), \quad z_{\min} = z(X^*) = 18,92;$$

$$X^*_{\max} = (6,96; 5,44), \quad z_{\max} = z(X^*) = 115,57.$$

4.3.3 Метод неопределенных множителей Лагранжа

Метод неопределенных множителей Лагранжа является классическим методом решения задач нелинейного программирования (в частности выпуклого). Рассмотрим метод неопределенных множителей Лагранжа для решения задач выпуклого программирования, в котором и целевая функция $f(X)$, и функции ограничений $\varphi_i(X)$ являются выпуклыми.

Рассмотрим классическую задачу оптимизации:

$$z = f(x_1, x_2, \dots, x_n) \rightarrow \max (\min), \quad (4.9)$$

$$\varphi_i(x_1, x_2, \dots, x_n) = b_i, \quad i = \overline{1, m}. \quad (4.10)$$

Эта задача выделяется из общей задачи нелинейного программирования тем, что среди ограничений (4.10) нет неравенств, нет условий неотрицательности переменных, их дискретности, $m < n$, функции $f(x_1, x_2, \dots, x_n)$ и $\varphi_i(x_1, x_2, \dots, x_n)$ непрерывны и имеют непрерывные частные производные, по крайней мере, второго порядка.

Для решения задачи (4.9)–(4.10) методом неопределенных множителей Лагранжа необходимо:

1 Составить функцию Лагранжа :

$$L(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m) = f(x_1, x_2, \dots, x_n) + \sum_{i=1}^m \lambda_i (b_i - \varphi_i(x_1, x_2, \dots, x_n)), \quad (4.11)$$

где $\lambda_1, \lambda_2, \dots, \lambda_m$ – множители Лагранжа.

2 Найти частные производные функции Лагранжа по переменным величинам x_j ($j = \overline{1, n}$), λ_i ($i = \overline{1, m}$) и приравнять их к нулю:

$$\begin{cases} \frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial \varphi_i}{\partial x_j} = 0, \quad j = \overline{1, n}; \\ \frac{\partial L}{\partial \lambda_i} = b_i - \varphi_i = 0, \quad i = \overline{1, m}. \end{cases} \quad (4.12)$$

3 Решить полученную систему уравнений (4.12) и тем самым определить стационарные точки (т. е. точки, подозрительные на экстремум) функции Лагранжа.

4 Среди стационарных точек функции Лагранжа, взятых без координат λ_i ($i = \overline{1, m}$), выбрать точки, в которых достигается условный экстремум функции $f(X)$, вычислить значения функции в этих точках и выбрать среди них точку с максимальным (минимальным) значением функции $f(X)$.

Необходимое условие экстремума сводится к существованию решения системы уравнений (4.12).

Достаточные условия выводятся, если для функций $f(x_1, x_2, \dots, x_n)$ и $\varphi_i(x_1, x_2, \dots, x_n)$ существуют вторые частные производные и они непрерывны.

Если в стационарной точке второй дифференциал функции Лагранжа $d^2L > 0$, то функция $f(x_1, x_2, \dots, x_n)$ в данной точке имеет условный минимум, если $d^2L < 0$, то условный максимум.

В случае трех и более переменных экстремум функции лучше определять по квадратичной форме.

Для каждой стационарной точки X_i составляется квадратичная форма $\left[\frac{\partial^2 L(X_i)}{\partial X^2} \right]$ и проверяется на знакоопределенность.

Если квадратичная форма положительно определена, то X_i – точка локального минимума, если отрицательно определена, то X_i – точка локального максимума.

Приведем примеры матриц квадратичных форм функций:

а) двух переменных:

$$\frac{\partial^2 L}{\partial X^2} = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 x_2} \\ \frac{\partial^2 L}{\partial x_2 x_1} & \frac{\partial^2 L}{\partial x_2^2} \end{bmatrix};$$

б) трех переменных:

$$\frac{\partial^2 f}{\partial X^2} = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 x_2} & \frac{\partial^2 L}{\partial x_1 x_3} \\ \frac{\partial^2 L}{\partial x_2 x_1} & \frac{\partial^2 L}{\partial x_2^2} & \frac{\partial^2 L}{\partial x_2 x_3} \\ \frac{\partial^2 L}{\partial x_3 x_1} & \frac{\partial^2 L}{\partial x_3 x_2} & \frac{\partial^2 L}{\partial x_3^2} \end{bmatrix};$$

в) n – переменных:

$$\frac{\partial^2 f}{\partial X^2} = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 x_2} & \dots & \frac{\partial^2 L}{\partial x_1 x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 L}{\partial x_n x_1} & \frac{\partial^2 L}{\partial x_n x_2} & \dots & \frac{\partial^2 L}{\partial x_n^2} \end{bmatrix}.$$

Таким образом, идея метода множителей Лагранжа заключается в том, что исходная задача (4.9)–(4.10) отыскания условного экстремума целевой функции $f(x_1, x_2, \dots, x_n)$ сводится к задаче отыскания безусловного экстремума функции Лагранжа $L(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m)$, которая решается приравниванием к нулю частных производных по x_j ($j = \overline{1, n}$) и λ_i ($i = \overline{1, m}$).

Множителям Лагранжа можно придать экономический смысл. Множители Лагранжа λ_i ($i = \overline{1, m}$) показывают, насколько изменится значение целевой функции при изменении правой части i -го ограничения на единицу. Таким образом, если целевую функцию f интерпретировать как доход или стоимость, а b_i как объемы некоторых ресурсов, то множители Лагранжа показывают, как изменится максимальный доход (или минимальная стоимость), если количество ресурса i -го вида увеличится на единицу. Метод множителей Лагранжа распространяется и на случай, когда переменные неотрицательны и некоторые ограничения заданы в форме неравенств.

Пример 4.3. Методом неопределенных множителей Лагранжа найти экстремумы функции при заданных ограничениях:

$$z = 3x_1 + 2x_2 \rightarrow \max (\min),$$

$$x_1^2 + x_2^2 = 9.$$

Составим функцию Лагранжа:

$$L(x_1, x_2, \lambda) = 3x_1 + 2x_2 + \lambda(9 - x_1^2 - x_2^2),$$

Найдем частные производные функции Лагранжа по x_1, x_2, λ , приравняем их нулю и решим полученную систему уравнений:

$$\begin{cases} \frac{\partial L}{\partial x_1} = 3 - 2\lambda x_1 = 0, \\ \frac{\partial L}{\partial x_2} = 2 - 2\lambda x_2 = 0, \\ \frac{\partial L}{\partial \lambda} = 9 - x_1^2 - x_2^2 = 0. \end{cases}$$

Решая полученную систему уравнений, находим:

$$\lambda_1 = \frac{\sqrt{13}}{6}, \quad x_1' = \frac{9}{\sqrt{13}}, \quad x_2' = \frac{6}{\sqrt{13}};$$

$$\lambda_2 = -\frac{\sqrt{13}}{6}, \quad x_1'' = -\frac{9}{\sqrt{13}}, \quad x_2'' = -\frac{6}{\sqrt{13}}.$$

Имеем две стационарные точки $M_1\left(\frac{9}{\sqrt{13}}; \frac{6}{\sqrt{13}}\right)$, $M_2\left(-\frac{9}{\sqrt{13}}; -\frac{6}{\sqrt{13}}\right)$.

Найдем частные производные второго порядка и второй дифференциал функции Лагранжа:

$$\frac{\partial^2 L}{\partial x_1^2} = -2\lambda, \quad \frac{\partial^2 L}{\partial x_2^2} = -2\lambda, \quad \frac{\partial^2 L}{\partial x_1 \partial x_2} = 0,$$

$$d^2 L = \frac{\partial^2 L}{\partial x_1^2} d x_1^2 + 2 \frac{\partial^2 L}{\partial x_1 \partial x_2} d x_1 d x_2 + \frac{\partial^2 L}{\partial x_2^2} d x_2^2,$$

$$d^2 L = -2\lambda d x_1^2 - 2\lambda d x_2^2 = -2\lambda (d x_1^2 + d x_2^2).$$

При $\lambda_1 = \frac{\sqrt{13}}{6}$

$$d^2 L = -2 \cdot \frac{\sqrt{13}}{6} (d x_1^2 + d x_2^2) = -\frac{\sqrt{13}}{3} (d x_1^2 + d x_2^2) < 0,$$

значит, в точке $M_1\left(\frac{9}{\sqrt{13}}; \frac{6}{\sqrt{13}}\right)$ функция $z = 3x_1 + 2x_2$ имеет максимум,

равный $z_{\max} = 3 \cdot \frac{9}{\sqrt{13}} + 2 \cdot \frac{6}{\sqrt{13}} = \frac{39}{\sqrt{13}} = 3\sqrt{13}$.

При $\lambda_2 = -\frac{\sqrt{13}}{6}$

$$d^2 L = -2 \cdot \left(-\frac{\sqrt{13}}{6}\right) (d x_1^2 + d x_2^2) = \frac{\sqrt{13}}{3} (d x_1^2 + d x_2^2) > 0,$$

значит, в точке $M_2\left(-\frac{9}{\sqrt{13}}; -\frac{6}{\sqrt{13}}\right)$ функция $z = 3x_1 + 2x_2$ имеет минимум,

равный $z_{\min} = -3 \cdot \frac{9}{\sqrt{13}} - 2 \cdot \frac{6}{\sqrt{13}} = -\frac{39}{\sqrt{13}} = -3\sqrt{13}$.

Итак,

$$X^*_{\max} = \left(\frac{9}{\sqrt{13}}; \frac{6}{\sqrt{13}}\right), \quad z_{\max} = z(X^*) = 3\sqrt{13};$$

$$X^*_{\min} = \left(-\frac{9}{\sqrt{13}}; -\frac{6}{\sqrt{13}}\right), \quad z_{\min} = z(X^*) = -3\sqrt{13}.$$

4.3.4 Решение задачи о производстве продукции графическим методом и методом неопределенных множителей Лагранжа

Пример 4.4. Постановка задачи. Предприятие может выпускать два вида продукции. Затраты на производство продукции определяются выражением $c_j - l_j x_j$, где x_j – искомый объем производства продукции j -го вида ($j = 1, 2$); c_j – себестоимость продукции j -го вида ($c_1 = 100$ ден. ед., $c_2 = 80$ ден. ед.); l_j – коэффициент снижения затрат ($l_1 = 1, l_2 = 1$). Предприятие располагает ресурсами двух видов в количествах 160 и 180 единиц соответственно. Нормы расхода ресурса первого вида на единицу продукции каждого вида равны 2 и 4 единицы, ресурса второго вида – 3 и 2 единицы соответственно.

Требуется определить объем производства продукции, обеспечивающий минимум суммарных затрат при выполнении ограничений по ресурсам и суммарному объему выпуска продукции, который должен быть не меньше 30 единиц.

Составим математическую модель задачи.

Найдем план $X = (x_1, x_2)$ производства продукции, обеспечивающий минимум суммарных затрат

$$z = f(x_1, x_2) = (100 - x_1) x_1 + (80 - x_2) x_2 \rightarrow \min,$$

при ограничениях:

– на суммарный объем выпуска продукции

$$x_1 + x_2 \geq 30,$$

– ресурсы

$$2 x_1 + 4 x_2 \leq 160,$$

$$3 x_1 + 2 x_2 \leq 180,$$

условие неотрицательности

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

Решим задачу графическим методом.

С учетом системы ограничений построим множество допустимых решений Ω – многоугольник $ABCDE$ (рисунок 4.9).

Преобразуем целевую функцию:

$$z = f(x_1, x_2) = (100 - x_1) x_1 + (80 - x_2) x_2 = -((x_1 - 50)^2 + (x_1 - 40)^2) + 4100.$$

Целевая функция

$$z = f(x_1, x_2) = -((x_1 - 50)^2 + (x_1 - 40)^2) + 4100$$

достигает минимального значения при условии, что функция $f_1 = (x_1 - 50)^2 + (x_1 - 40)^2$ принимает максимальное значение.

Линиями уровня целевой функции будут окружности с центром в точке $O(50;40)$. Проводя окружности различных радиусов, видим, что максимальное значение на множестве допустимых решений функция f_1 принимает в точке $A(0;30)$ (чем больше радиус окружности, тем больше значение функции f_1).

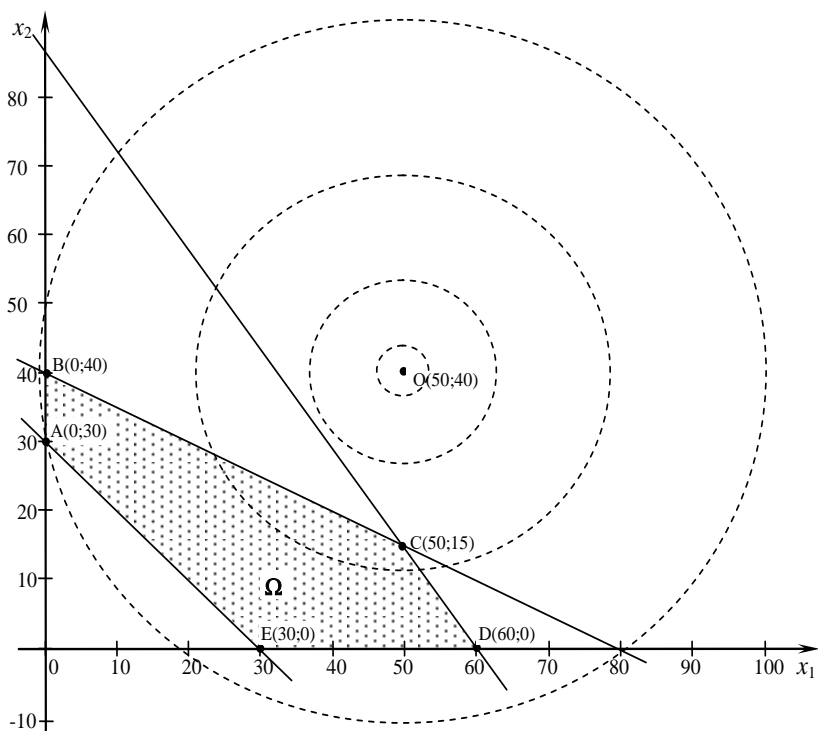


Рисунок 4.9 – Графическое решение задачи

Получено оптимальное решение $X^*_{\min} = (0; 30)$. Вычислим минимальное значение целевой функции

$$z_{\min} = z(X^*) = (100 - 0) \cdot 0 + (80 - 30) \cdot 30 = 1500.$$

Таким образом, если предприятие будет выпускать 30 единиц продукции второго вида, то общие затраты будут минимальными и составят 1500 ден. ед.

Пример 4.5. Постановка задачи. Предприятие может выпускать два вида продукции. Затраты на производство продукции определяются выражением $c_j + l_j x_j$, где x_j – искомый объем производства продукции j -го вида ($j = 1, 2$); c_j – себестоимость продукции j -го вида ($c_1 = 60$ ден. ед., $c_2 = 20$ ден. ед.); l_j – коэффициент изменения затрат ($l_1 = 1/4$; $l_2 = 1$).

Требуется определить объем производства продукции, обеспечивающий минимум суммарных затрат при выполнении ограничений по суммарному объему выпуска продукции, который должен составлять 100 единиц.

Составим математическую модель задачи.

Найдем план $X = (x_1, x_2)$ производства продукции, обеспечивающий минимум суммарных затрат

$$z = f(x_1, x_2) = \left(60 + \frac{1}{4}x_1\right)x_1 + (20 + x_2)x_2 \rightarrow \min,$$

при ограничениях на суммарный объем выпуска продукции

$$x_1 + x_2 = 100.$$

Решим задачу методом неопределенных множителей Лагранжа.

Составим функцию Лагранжа:

$$L(x_1, x_2, \lambda) = 60x_1 + \frac{1}{4}x_1^2 + 20x_2 + x_2^2 + \lambda(100 - x_1 - x_2) \rightarrow \min.$$

Найдем частные производные функции Лагранжа по неизвестным величинам x_1, x_2, λ и приравняем их нулю:

$$\begin{cases} \frac{\partial L}{\partial x_1} = 60 + \frac{1}{2}x_1 - \lambda = 0, \\ \frac{\partial L}{\partial x_2} = 20 + 2x_2 - \lambda = 0, \\ \frac{\partial L}{\partial \lambda} = 100 - x_1 - x_2 = 0. \end{cases}$$

Решая полученную систему линейных уравнений, находим:

$$x_1 = 64, \quad x_2 = 36, \quad \lambda = 92.$$

Найдем частные производные второго порядка и второй дифференциал функции Лагранжа:

$$\frac{\partial^2 L}{\partial x_1^2} = \frac{1}{2}, \quad \frac{\partial^2 L}{\partial x_2^2} = 2, \quad \frac{\partial^2 L}{\partial x_1 \partial x_2} = 0,$$

$$d^2L = \frac{\partial^2 L}{\partial x_1^2} dx_1^2 + 2 \frac{\partial^2 L}{\partial x_1 \partial x_2} dx_1 dx_2 + \frac{\partial^2 L}{\partial x_2^2} dx_2^2 = \frac{1}{2} dx_1^2 + 2 dx_2^2 > 0.$$

Так как $d^2L > 0$, то точка с координатами $X^* = (64; 36)$ является точкой минимума. Вычислим минимальное значение целевой функции

$$z_{\min} = z(X^*) = \left(60 + \frac{1}{4} \cdot 64\right) \cdot 64 + (20 + 36) \cdot 36 = 6880.$$

Таким образом, если предприятие будет выпускать 64 единицы продукции первого вида и 36 единиц продукции второго вида, то общие затраты будут минимальными и составят 6880 ден. ед.

4.4 Градиентные методы

Градиентные методы можно применять для решения любой задачи нелинейного программирования, однако указанные методы позволяют найти локальный экстремум функции.

Рассмотрим применение градиентных методов для решения задач выпуклого программирования, так как в этих задачах всякий локальный экстремум является глобальным.

Если функция $f(X)$ дифференцируема в точке $X = (x_1, x_2, \dots, x_n)$, то *градиентом* функции $f(X)$ называется n -мерный вектор

$$\nabla f(X) = \text{grad } f(X) = \left(\frac{\partial f(X)}{\partial x_1}, \dots, \frac{\partial f(X)}{\partial x_j}, \dots, \frac{\partial f(X)}{\partial x_n} \right). \quad (4.14)$$

Градиент функции в точке направлен по нормали к линии уровня $f(X)$ и показывает направление наискорейшего возрастания функции в этой точке.

Антиградиент $[-\nabla f(X)]$ – противоположный градиенту вектор – показывает направление наискорейшего убывания функции в точке.

Если функция $f(X)$ является выпуклой, то необходимым и достаточным условием существования глобального максимума или минимума в точке X^* является равенство нулю градиента функции в этой точке $\nabla f(X^*) = 0$.

Сущность *градиентных методов* заключается в следующем.

Пусть, например, требуется найти экстремум функции $f(X)$.

1 Выбираем начальную точку X^0 , принадлежащую области определения функции.

2 Определяем направление градиента функции в точке X^0 :

$$\nabla f(X^0) = \left(\frac{\partial f(X^0)}{\partial x_1}, \dots, \frac{\partial f(X^0)}{\partial x_j}, \dots, \frac{\partial f(X^0)}{\partial x_n} \right).$$

3 Выполняем перемещение из точки X^0 на величину, равную некоторому шагу λ , в точку X^1 . Известно, что параметрическое уравнение прямой, проходящей через заданную точку X^0 в заданном направлении $\nabla f(X^0)$, имеет вид

$$x_j^1 = x_j^0 \pm \lambda \nabla f(X^0), \quad j = \overline{1, n}. \quad (4.15)$$

Если определяется максимум функции $f(X)$, то перемещение ведется в направлении градиента, и в этом случае перед λ ставится знак плюс; если ищется минимум функции $f(X)$, то перемещение ведется в направлении антиградиента (перед λ ставится знак минус).

В точке X^1 устанавливаем новое градиентное направление, и процесс вычислений повторяем.

4 Выполнение итерационной процедуры прекращается, если выполняется один из критериев достижения экстремума целевой функции.

Признаком достижения экстремума функции $f(X)$ служит обращение в нуль градиента $\nabla f(X) = 0$.

Но на практике, как правило, ограничиваются приближенным решением и критерием достижения требуемого результата является выполнение условия

$$\sum_{j=1}^n \left(\frac{\partial f(X)}{\partial x_j} \right)^2 \leq \varepsilon, \text{ где } \varepsilon > 0 - \text{наперед заданное число.}$$

Существуют и другие критерии. Например, после каждой итерации можно сравнивать достигнутое значение целевой функции с ее значением в предыдущей точке. Если выполняется $|f(X^{k+1}) - f(X^k)| \leq \varepsilon$, где $\varepsilon > 0$ – наперед заданное число, то поиск экстремума функции прекращается.

Таким образом, основная идея градиентных методов состоит в построении последовательности точек X^0, X^1, \dots, X^n , сходящейся к точке экстремума X^* , т. е. для точек последовательности выполняются условия $f(X^0) < f(X^1) < f(X^2) < \dots < f(X^k) < \dots$ (если определяется максимум функции $f(X)$) и $f(X^0) > f(X^1) > f(X^2) > \dots > f(X^k) > \dots$ (если определяется минимум функции $f(X)$).

Градиентные методы, как правило, позволяют получать точное решение за бесконечное число шагов и только в некоторых случаях – за конечное. В связи с этим градиентные методы относятся к приближенным методам решения.

Среди градиентных методов выделяют: метод градиентного подъема (спуска) и метод наискорейшего подъема (спуска). Указанные методы отличаются различными способами выбора величины шага λ . Рассмотрим эти градиентные методы.

В *методе градиентного подъема (спуска)* движение из точки в точку осуществляется с постоянным шагом λ по формуле

$$x_j^{k+1} = x_j^k \pm \lambda \nabla f(X^k), \quad j = \overline{1, n}. \quad (4.16)$$

Знак плюс ставится при решении задачи на максимум, знак минус – при решении задачи на минимум.

После определения очередного приближения X^{k+1} для задачи на максимум проверяется условие $f(X^{k+1}) > f(X^k)$. Если это условие выполняется, то итерация закончена. Иначе, уменьшаем величину шага λ , например до $\lambda/2$, и производим вычисление очередного приближения. Для задачи на минимум проверяем, соответственно, условие $f(X^{k+1}) < f(X^k)$.

Процесс вычисления прекращается, если выполняется один из критериев достижения экстремума целевой функции.

Пример 4.6. Для производства продукции предприятие закупает сырье двух видов в объемах x_1 и x_2 тонн. Расходы на хранение сырья на складе определяются функцией

$$z = f(x_1, x_2) = 4x_1^2 - 4x_1 + x_2^2 - 8x_2 - 2x_1x_2 + 40.$$

Требуется определить объемы сырья, расходы на хранение которого будут минимальны.

Примем точность вычислений $\varepsilon = 0,5$.

В качестве начальной возьмем точку $X^0 = (3, 5)$, шаг λ примем равным 0,05.

Вычисляем значение функции в точке X^0 :

$$f(X^0) = 4 \cdot 3^2 - 4 \cdot 3 + 5^2 - 8 \cdot 5 - 8 \cdot 5 - 2 \cdot 3 \cdot 5 + 40 = 19.$$

Находим градиент функции в точке X^0 :

$$\frac{\partial f}{\partial x_1} = 8x_1 - 2x_2 - 4; \quad \frac{\partial f}{\partial x_2} = 2x_2 - 2x_1 - 8.$$

$$\frac{\partial f(X^0)}{\partial x_1} = 10; \quad \frac{\partial f(X^0)}{\partial x_2} = -4.$$

Определяем координаты точки $X^1(x_1^1, x_2^1)$:

$$x_1^1 = x_1^0 - \lambda \nabla f(X^0) = 3 - 0,05 \cdot 10 = 2,5;$$

$$x_2^1 = x_2^0 - \lambda \nabla f(X^0) = 5 - 0,05 \cdot (-4) = 5,2;$$

$$\frac{\partial f(X^1)}{\partial x_1} = 5,6; \quad \frac{\partial f(X^1)}{\partial x_2} = -2,6; \quad f(X^1) = 14,44.$$

Разность $|f(X^1) - f(X^0)| = |14,44 - 19| = 4,56 > \varepsilon$, следовательно, выполняем вторую итерацию и переходим к точке $X^2(x_1^2, x_2^2)$:

$$x_1^2 = x_1^1 - \lambda \nabla f(X^1) = 2,5 - 0,05 \cdot 5,6 = 2,22;$$

$$x_2^2 = x_2^1 - \lambda \nabla f(X^1) = 5,2 - 0,05 \cdot (-2,6) = 5,33;$$

$$\frac{\partial f(X^2)}{\partial x_1} = 3,1; \quad \frac{\partial f(X^2)}{\partial x_2} = -1,78; \quad f(X^2) = 12,93;$$

$$|f(X^2) - f(X^1)| = |12,93 - 14,44| = 1,49 > \varepsilon.$$

Выполняем третью итерацию и переходим к точке $X^3(x_1^3, x_2^3)$:

$$x_1^3 = x_1^2 - \lambda \nabla f(X^2) = 2,22 - 0,05 \cdot 3,1 = 2,07;$$

$$x_2^3 = x_2^2 - \lambda \nabla f(X^2) = 5,33 - 0,05 \cdot (-1,78) = 5,42;$$

$$\frac{\partial f(X^3)}{\partial x_1} = 1,72; \quad \frac{\partial f(X^3)}{\partial x_2} = -1,30; \quad f(X^3) = 12,43;$$

$$|f(X^3) - f(X^2)| = |12,43 - 12,93| = 0,5 = \varepsilon.$$

Учитывая, что требуемая точность результата достигнута, процесс вычисления можно прекратить.

Минимальные расходы на хранение при объемах сырья $x_1 \approx 2,07$ и $x_2 \approx 5,42$ составят 12,43 ден. ед.

При решении таких задач выбор величины шага λ – достаточно сложная и важная задача. Если длина шага большая, то можно перескочить экстремум и появляется опасность заикливания. Если длина шага малая, то процесс поиска экстремума может оказаться очень медленным.

Трудности, связанные с выбором длины шага, преодолеваются в методе наискорейшего спуска (подъема), когда в каждой итерации вычисляется оптимальное значение λ .

В *методе наискорейшего подъема (спуска)* движение из точки в точку осуществляется по формуле

$$x_j^{k+1} = x_j^k \pm \lambda_k \nabla f(X^k), \quad j = \overline{1, n}. \quad (4.17)$$

Знак плюс ставится при решении задачи на максимум, знак минус – при решении задачи на минимум.

Оптимальное значение величины шага λ_k на каждой итерации вычисляется исходя из следующего соотношения для вводимой функции $\varphi(\lambda_k)$:

$$\varphi(\lambda_k) = f(X^{k+1}) = f(X^k \pm \lambda_k \nabla f(X^k)) \rightarrow \max(\min). \quad (4.18)$$

Величина λ_k , при которой достигается наибольшее (наименьшее) значение функции $\varphi(\lambda_k)$, может быть определена из необходимого условия экстремума функции $\varphi(\lambda_k)$:

$$\frac{d\varphi(\lambda_k)}{d\lambda_k} = 0. \quad (4.19)$$

Процесс вычисления прекращается, если выполняется один из критериев достижения экстремума целевой функции.

Градиентные методы можно также применять при нахождении экстремумов нелинейной целевой функции при ограничениях. Начальная точка берется в этом случае из ограничений. После получения очередной точки идет проверка по ограничениям.

5 ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

5.1 Основные понятия динамического программирования

Под **динамическим программированием** (ДП) понимается класс методов, предназначенных для нахождения оптимального управления многошаговыми процессами. Предметом исследования этого раздела являются разнообразные управляемые системы, процесс управления которыми может быть разделен на шаги. При этом предполагается, что эффективность управления системой в целом определяется эффективностями управлений, используемых на каждом из шагов.

Основная идея динамического программирования состоит в переходе от исходной многомерной задачи поиска оптимального управления системой в целом к последовательности взаимосвязанных задач меньшей размерности поиска оптимального управления для каждого из шагов.

Приведем несколько примеров задач, для решения которых может быть использован метод динамического программирования.

1 *Разнообразные задачи распределения ресурсов и перспективного планирования.* Например, на реконструкцию и модернизацию n предприятий отрасли выделены средства в объеме C . Для каждого из рассматриваемых предприятий известна функция $g_i(x)$, ($i = 1, 2, \dots, n$), характеризующая ожидаемую прибыль от реконструкции этого предприятия в зависимости от объема выделенных ему средств x . Требуется так распределить имеющиеся средства C между предприятиями, чтобы суммарная прибыль от реконструкции всех производств была максимальной.

2 *Задача определения наиболее экономичного маршрута на сети.* Пусть задана транспортная сеть, отражающая возможные варианты доставки груза из пункта A в пункт B . Для каждого из участков этого пути известны стоимости доставки по нему единицы груза. Требуется определить наиболее экономичный маршрут доставки груза из пункта A в пункт B .

Подобным образом могут быть сформулированы задачи определения наиболее экономичного варианта прокладки участка железнодорожного пути (или автомобильной дороги) между пунктами A и B .

3 *Задача о ремонте и замене оборудования.* Известно, что любое оборудование в процессе его эксплуатации устаревает морально и физически. С увеличением возраста оборудования снижается доход от его эксплуатации

и возрастают расходы, связанные с его содержанием и ремонтом. В некоторый момент времени может оказаться более выгодным произвести замену устаревшего оборудования новым. Требуется определить оптимальные сроки ремонта и замены имеющегося оборудования, позволяющие максимизировать суммарный доход от его эксплуатации в течение планируемого периода.

Сформулируем *характерные черты задач динамического программирования*:

- имеется управляемая система S , в которой происходят некоторые физические, технические, технологические, экономические или др. процессы, которые могут рассматриваться как многошаговые;

- на каждом шаге i система может находиться в одном из состояний x_i из множества допустимых состояний X_i для этого шага;

- для каждого шага i определено множество допустимых управлений U_i , под воздействием которых система переходит из состояний множества X_{i-1} в одно из состояний множества X_i ;

- эффективность использования управления $u_i \in U_i$ при нахождении системы в состоянии $x_{i-1} \in X_{i-1}$ оценивается значением целевой функции на i -м шаге: $z_i(x_{i-1}, u_i)$;

- важнейшей предпосылкой применения алгоритма динамического программирования является *свойство отсутствия последействия* в исследуемой системе. Это свойство состоит в том, что на каждом шаге состояние системы S зависит только от состояния системы перед совершением этого шага и от управления, выбранного на этом шаге. Символически свойство отсутствия последействия может быть записано следующим образом: $x_i = f(x_{i-1}, u_i)$. Иначе говоря, предыстория процесса не сказывается на его протекании в будущем;

- в большинстве задач ДП предполагается, что целевая функция z , характеризующая эффективность управления системой в целом, является *аддитивной* функцией, т. е. она может быть представлена в виде суммы эффективностей управления, достигнутых при совершении каждого из шагов:

$$z(u_1, u_2, \dots, u_n) = \sum_{i=1}^n z_i(u_i);$$

- в зависимости от типа задачи может быть задано начальное или конечное состояние системы либо оба этих состояния; иногда задаются области начальных и конечных состояний системы S ;

- требуется выбрать из множества допустимых управлений такое управление системой S , которое переводит ее из начального состояния в конечное и при этом обращает в экстремум целевую функцию.

5.2 Принцип оптимальности Беллмана

Процесс поиска оптимального решения задачи ДП состоит из двух этапов, называемых соответственно *условной оптимизацией* и *безусловной оптимизацией*.

В процессе первого этапа (*условной оптимизации*) для каждого шага оптимизируемого процесса определяется множество условно-оптимальных управлений. Вычислительная процедура условной оптимизации базируется на *принципе оптимальности Беллмана*, который может быть сформулирован следующим образом: каково бы ни было состояние системы в результате $i - 1$ шагов, управление на i -м шаге должно выбираться таким образом, чтобы это управление в совокупности с оптимальными управлениями на всех последующих шагах доставляло экстремум целевой функции.

В соответствии с этим принципом расчеты на этапе условной оптимизации осуществляются в направлении от последнего шага к первому. Сначала определяется множество оптимальных управлений для последнего, n -го шага; затем определяется множество условно-оптимальных управлений для $(n - 1)$ -го шага, доставляющих экстремум целевой функции на двух последних шагах; после этого определяются условно-оптимальные управления для $(n - 2)$ -го шага, оптимизирующие рассматриваемый процесс на трех последних шагах и т. д.

В процессе второго этапа (*безусловной оптимизации*) составляется искомое оптимальное управление системой в целом из найденных в процессе первого этапа шаговых условно-оптимальных управлений.

При решении задач ДП используются следующие обозначения:

X_i – множество возможных состояний системы в конце i -го шага;

X_{i-1} – множество возможных состояний системы перед совершением i -го шага;

тогда X_0 – множество начальных состояний;

X_n – множество конечных состояний системы;

U_i – множество допустимых управлений, которые могут быть выбраны при совершении i -го шага;

$z_i(x_{i-1}, u_i)$ – значение целевой функции на i -м шаге, при условии, что перед совершением этого шага система S находилась в состоянии x_{i-1} и на i -м шаге было выбрано управление u_i ;

$F_i(x_{i-1}, u_i)$ – условно-оптимальное значение целевой функции при совершении шагов от i -го до n -го включительно (при условии, что перед совершением i -го шага система находилась в состоянии x_{i-1} и на i -м шаге было выбрано управление u_i);

$F_{i+1}(x_i)$ – условно-оптимальное значение целевой функции при совершении шагов от $i + 1$ -го до n -го включительно.

С учетом введенных обозначений принцип оптимальности Беллмана может быть записан следующим образом:

$$F_i(x_{i-1}, u_i) = \underset{u_i}{\text{extr}}(z_i(x_{i-1}, u_i) + F_{i+1}(x_i)). \quad (5.1)$$

Соотношение (5.1) называют *основным функциональным соотношением динамического программирования*.

5.3 Решение задачи о выборе маршрута методом динамического программирования

Для иллюстрации описанной вычислительной процедуры метода динамического программирования рассмотрим задачу о выборе маршрута.

Пример 5.1. На данной сети дорог (рисунок 5.1) указаны стоимости доставки единицы груза из пункта в пункт. Найти наиболее экономный маршрут перевозки груза из пункта 1 в пункт 10 и суммарные затраты.

Концепция динамического программирования предполагает, что процессу принятия решения в оптимизационной задаче может быть придан шаговый характер.

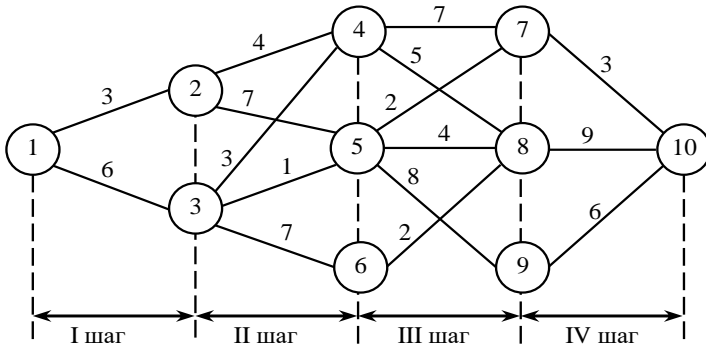


Рисунок 5.1 – Схема возможных вариантов доставки груза

С этой целью разобьем все пункты сети на группы (таблица 5.1). К группе I отнесем пункт 1, к группе II – пункты, в которые можно попасть непосредственно из пункта 1 (такowymi будут 2 и 3), к группе III отнесем пункты, в которые можно попасть непосредственно из любого пункта группы II (такowymi будут 4, 5 и 6), и т. д.

В результате движение транспорта с грузом из пункта 1 в пункт 10 можно рассматривать как четырехшаговый процесс: на первом шаге транспорт перемещается из пункта 1 в какой-то пункт группы II, на втором шаге – из пункта группы II в пункт группы III и т. д.

Таблица 5.1 – Разбиение пунктов сети на группы

I	II	III	IV	V
1	2	4	7	10
	3	5	8	
		6	9	

После разбиения пунктов сети на группы формирование наиболее экономного маршрута может быть реализовано за четыре шага.

В рассматриваемой задаче в качестве физической системы выступает транспорт с грузом, перемещающийся из начального состояния c_1 (из пункта 1) в конечное состояние c_{10} (пункт 10), и сеть дорог. За состояние c_i , системы перед i -м шагом естественно принять местонахождение транспорта с грузом в пункте, в котором он пребывает перед этим шагом.

Управление u_i , на i -м шаге состоит в выборе дороги (i, j) , по которой следует направлять груз из данного пункта в соседний в общем направлении на пункт 10. Состояние в конце шага определяется номером пункта, в который будет доставлен груз в результате сделанного выбора (принятого управления), а значение z_i целевой функции на i -м шаге – это затраты на перевозку единицы груза из данного пункта в выбранный соседний пункт.

Этап 1. В соответствии с вычислительной схемой метода динамического программирования условную оптимизацию начнем с анализа четвертого шага.

Возможные состояния, в которых транспорт с грузом может оказаться перед четвертым шагом, зависят от управлений на предшествующих шагах и соответствуют его местонахождению либо в пункте 7, либо в пункте 8, либо в пункте 9. Обозначим эти состояния соответственно c_7, c_8, c_9 . Они образуют множество возможных состояний на начало четвертого шага. Это будет множество x_3 . Из каждого указанного состояния можно перейти в конечное состояние c_{10} единственным путем: из пункта 7 в пункт 10 груз может быть доставлен только дорогой $(7, 10)$, из 8 в 10 – дорогой $(8, 10)$, из 9 в 10 – дорогой $(9, 10)$.

Решения о доставке груза по названным дорогам и являются управлениями на четвертом шаге, соответствующими указанным состояниям. Итак, множество u_4 управлений на четвертом шаге состоит из элементов: $(7, 10)$, $(8, 10)$ и $(9, 10)$.

Шаг 1. Условно-оптимальные затраты на этом шаге в общем случае выражаются основным функциональным уравнением, записанным для четвертого шага, если в равенстве (5.2) положить $N = 4$:

$$F_4(x_3) = \min_{u_4} (z_4(x_3, u_4)). \quad (5.2)$$

Шаг 2. В данном случае для каждого состояния целевая функция z_4 явля-

ется однозначной (каждому состоянию соответствует единственное управление), поэтому

$$\min_{u_4}(z_4(x_3, u_4)) = z_4(x_3, u_4), \text{ т. е. } F_4(x_3) = z_4(x_3, u_4).$$

Как видно из рисунка 5.1, функция $F_4(x_3)$ в зависимости от состояний x_3 принимает значения 3, 9 и 6, а управления (7, 10), (8, 10), (9, 10) будут условно-оптимальными соответственно для состояний c_7, c_8, c_9 . Этим завершается первый этап условной оптимизации.

Для большей наглядности результаты проведенных рассуждений оформим в виде таблицы 5.2.

Таблица 5.2 – Результаты расчетов для четвертого шага: $i = 4$

x_3	u_4	x_4	F_4
c_7	(7, 10)	c_{10}	3*
c_8	(8, 10)	c_{10}	9
c_9	(9, 10)	c_{10}	6

Этап 2, шаг 3. Переходя ко второму этапу условной оптимизации – анализу третьего шага, запишем функциональное уравнение для этого шага, которое получится из равенства (5.1) при $i = 3$:

$$F_3(x_2) = \min_{u_3}(z_3(x_2, u_3) + F_4(x_3)). \quad (5.3)$$

получаем:

$$F_3(x_2) = \min_{(4,7), (4,8)} (7 + 3, 5 + 9) = 10.$$

Из этого равенства видно, что минимальные затраты в 10 ден. ед. соответствуют выбору дороги (4, 7).

Итак, для состояния c_4 условно-оптимальным управлением будет выбор дороги (4, 7), при этом условно-оптимальные транспортные затраты на участке от пункта 4 до конечного пункта 10 составят 10 ден. ед.

Аналогичным образом для состояния c_5 находим

$$F_3(x_2) = \min_{(5,7), (5,8), (5,9)} (2 + 3, 4 + 9, 8 + 6) = 5.$$

Откуда следует, что условно-оптимальным управлением будет выбор дороги (5, 7), обеспечивающий условно-оптимальные затраты в 5 ден. ед.

Для состояния c_6 получаем условно-оптимальные затраты $F_3(x_2) = \min_{(6,8)} (2 + 9) = 11$, которые соответствуют условно-оптимальному

управлению (6, 8) (выбору дороги (6, 8)).

В компактном виде приведенные выкладки записаны в форме таблицы 5.3. Этим завершается второй этап условной оптимизации.

Шаг 5. $i \neq 1$.

Таблица 5.3 – Результаты расчетов для третьего шага: $i = 3$

x_2	u_3	x_3	z_3	F_4	z_3+F_4	F_3
c_4	(4, 7)	c_7	7	3	10	10
	(4, 8)	c_8	5	9	14	
c_5	(5, 7)	c_7	2	3	5	5*
	(5, 8)	c_8	4	9	13	
	(5, 9)	c_9	8	6	14	
c_6	(6, 8)	c_8	2	9	11	11

Этап 3, шаг 3. Третий этап условной оптимизации – анализ второго шага – осуществляется совершенно аналогично второму этапу. Функциональное уравнение для второго шага запишется в следующей форме:

$$F_2(x_1) = \min_{u_2}(z_2(x_1, u_2) + F_3(x_2)). \quad (5.4)$$

Как видно из рисунка 5.1, здесь множество x_1 , состояний образуют элементы c_2 и c_3 , а множество управлений – элементы (2, 4), (2, 5), (3, 4), (3, 5), (3, 6). Значения целевой функции $z_2(x_1, u_2)$ заданы на рисунке 5.1, а значения $F_3(x_2)$ получены на предыдущем этапе условной оптимизации (таблица 5.3).

Шаг 4. Результаты вычислений приведены в таблице 5.4. Этим завершается третий этап условной оптимизации.

Таблица 5.4 – Результаты расчетов для второго шага: $i = 2$

x_1	u_2	x_2	z_2	F_3	z_2+F_3	F_2
c_2	(2, 4)	c_4	4	10	14	12
	(2, 5)	c_5	7	5	12	
c_3	(3, 4)	c_4	3	10	13	6*
	(3, 5)	c_5	1	5	6	
	(3, 6)	c_6	7	11	18	

Шаг 5. $i \neq 1$.

Этап 4, шаг 3. Заключительным этапом процедуры условной оптимизации является анализ первого шага.

Особенность этого шага состоит в том, что в его начале состояние c_1 системы определено однозначно: транспорт с грузом находится в пункте 1, так что множество x_0 состояний содержит единственный элемент c_1 , а множество u_1 управлений – два элемента: (1, 2) и (1, 3). Функциональное уравнение для этого шага имеет вид

$$F_1(x_0) = \min_{u_1}(z_1(x_0, u_1) + F_2(x_1)). \quad (5.5)$$

Шаг 4. Результаты вычислений приведены в таблице 5.5.

Таблица 5.5 – Результаты расчетов для первого шага: $i = 1$

x_0	u_1	x_1	z_1	F_2	z_1+F_2	F_1
c_1	(1, 2)	c_2	3	12	15	
	(1, 3)	c_3	6	6	12	12*

Шаг 5. $i = 1$.

Как видно из таблицы 5.5, условно-оптимальным управлением для этого шага является выбор дороги (1, 3), обеспечивающий минимальные суммарные затраты $F_1(x_0, u_1) = F_1(1, (1, 3)) = 12$ на всем пути от пункта 1 до пункта 10. Что же касается структуры этого пути (наиболее экономного маршрута), то она выяснится в процессе процедуры безусловной оптимизации.

Этап безусловной оптимизации.

При безусловной оптимизации остается пройти еще раз весь оптимизируемый процесс, но уже в прямом направлении, начиная с первого и кончая четвертым шагом, и «прочитать» искомое оптимальное управление (наиболее экономный маршрут), которое будет составлено из найденных ранее (при условной оптимизации) шаговых условно-оптимальных управлений (дорог между отдельными пунктами сети). Из таблицы 5.5 видно, что из начального пункта 1 (состояние c_1) груз следует направлять по дороге (1, 3), ибо именно этому условно-оптимальному шаговому управлению соответствуют минимальные затраты (см. последний столбец таблицы 5.5). В результате груз окажется в пункте 3 (состояние c_3). Переходя к таблице 5.4, замечаем, что из пункта 3 груз необходимо доставлять дорогой (3, 5) в пункт 5 (состояние c_5) (см. последний столбец таблицы 5.4). Из таблицы 5.3 видно, что далее груз должен перевозиться по дороге (5, 7) до пункта 7 (состояние c_7), откуда, как это следует из таблицы 5.2, он направляется дорогой (7, 10) в конечный пункт 10 (состояние c_{10}).

Таким образом, наиболее экономный маршрут пролегает через пункты 1, 3, 5, 7 и 10, при этом транспортные расходы минимизируются и составляют 12 ден. ед. на единицу груза.

При использовании метода динамического программирования часто получают побочные результаты, связанные с рассматриваемой задачей и нередко имеющие не менее важное значение, чем основной результат. В данном случае, кроме оптимального маршрута доставки груза из пункта 1 в пункт 10, информация, содержащаяся в таблицах 9–12, позволяет находить наиболее экономный маршрут в пункт 10 из любого другого пункта данной сети дорог. Находятся эти маршруты так же, как и сформированный выше маршрут 1–3–5–7–10. Так, например, оптимальный маршрут из пункта 2 в пункт 10 пройдет через пункты 5 и 7.

Заметим в заключение решения задачи, что в рассматриваемом примере проиллюстрирована одна из главных особенностей метода динамического

программирования: при выборе решения на каждом шаге необходимо руководствоваться интересами всего оптимизируемого процесса, а не только интересами данного шага. Именно этим объясняется включение в наиболее экономный маршрут дороги (1, 3), удельные транспортные затраты на которой в 2 раза выше, чем на дороге (1, 2). Эта «жертва» на первом шаге принесена с тем, чтобы минимизировать общие затраты на всем четырехшаговом маршруте.

5.4 Решение задачи о распределении средств на расширение производства методом динамического программирования

Широкий класс составляют задачи, в которых речь идет о наиболее целесообразном распределении во времени тех или иных ресурсов (денежных средств, рабочей силы, сырья и т. п.). Рассмотрим пример задачи такого рода.

Производственное объединение группе предприятий выделяет дополнительные средства на реконструкцию и модернизацию производства. По каждому из n предприятий известен возможный прирост $g_i(x)$ ($i = \overline{1, n}$) выпуска продукции в зависимости от выделенной ему суммы x . Требуется так распределить между предприятиями средства c , чтобы общий прирост $f_n(c)$ выпуска продукции был максимальным.

Физической системой S является производственное объединение, а в качестве шага следует понимать назначение той или иной суммы средств конкретному предприятию.

При $n = 1$ предположим, что все имеющиеся средства выделяются на реконструкцию и модернизацию одного предприятия. Обозначим через $f_1(x)$ максимально возможный прирост выпуска продукции на этом предприятии, соответствующий выделенной сумме x . Каждому значению x отвечает вполне определенное (единственное) значение $g_1(x)$ выпуска, поэтому можно записать, что

$$f_1(x) = \max(g_1(x)) = g_1(x). \quad (5.6)$$

Пусть теперь $n = 2$, т. е. средства распределяются между двумя предприятиями. Если второму предприятию выделена сумма x , то прирост продукции на нем составит $g_2(x)$. Оставшиеся другому предприятию средства $(c - x)$ в зависимости от величины x (а значит, и $c - x$) позволят увеличить прирост выпуска продукции до максимально возможного значения $f_1(c - x)$. При этом условии общий прирост выпуска продукции на двух предприятиях

$$g_2(x) + f_1(c - x). \quad (5.7)$$

Оптимальному значению $f_2(c)$ прироста продукции при распределении суммы c между двумя предприятиями соответствует такое x , при котором сумма, заданная формулой (5.7), максимальна.

Это можно выразить записью

$$f_2(c) = \max_{0 \leq x \leq c} (g_2(x) + f_1(c - x)). \quad (5.8)$$

Значение $f_3(c)$ можно вычислить, если известны значения $f_2(c)$, и т. д.

Функциональное уравнение Беллмана для рассматриваемой задачи запишется в следующем виде:

$$f_n(c) = \max_{0 \leq x \leq c} (g_n(x) + f_{n-1}(c - x)). \quad (5.9)$$

Итак, максимальный прирост выпуска продукции на n предприятиях определяется как максимум суммы прироста выпуска на n -м предприятии и прироста выпуска на остальных $n - 1$ предприятиях при условии, что оставшиеся после n -го предприятия средства распределяются между остальными предприятиями оптимально.

Имея функциональные уравнения, можем последовательно найти сначала f_1 , затем f_2, f_3, \dots и, наконец, f_{n-1} и f_n для различных значений распределяемой суммы средств.

Для отыскания оптимального распределения средств прежде всего найдем величину $x_n^*(c)$ ассигнований n -му предприятию, которая позволяет достичь полученного нами максимального значения f_n прироста продукции. По величине оставшихся средств $c - x_n^*(c)$ и уже известному нам значению f_{n-1} устанавливаем $x_{n-1}^*(c)$ – величину ассигнований $(n - 1)$ -му предприятию и т. д., и, наконец, находим $x_2^*(c)$ и $x_1^*(c)$.

Пример 5.2. Пусть имеются четыре предприятия, между которыми распределяется 100 тыс. ден. ед. Значения $g_i(x)$ прироста выпуска продукции на предприятиях в зависимости от выделенной суммы x приведены в таблице 5.6. Составить план распределения средств, максимизирующий общий прирост выпуска продукции.

Таблица 5.6 – Исходные данные

Средства c , тыс. ден. ед.	Предприятие			
	№ 1	№ 2	№ 3	№ 4
	Прирост выпуска продукции на предприятиях, $g_i(x)$, тыс. ден. ед.			
	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$
20	10	12	11	16
40	31	26	36	37
60	42	36	45	46
80	62	54	60	63
100	76	78	77	80

Условная оптимизация.

Этап 1. Пусть $n=1$. В соответствии с формулой (5.6) в зависимости от начальной суммы c получаем с учетом таблицы 5.6 значения $f_1(c)$, помещенные в таблицу 5.7.

Таблица 5.7 – Вычисление $f_1(c)$

$x_1^*(c)$	$f_1(c)$
20	10
40	31
60	42
80	62
100	76

Этап 2. Предположим теперь, что средства вкладываются в два предприятия. Тогда в соответствии с формулой (5.8)

$$f_2(c) = \max_{0 \leq x \leq c} (g_2(x) + f_1(c - x)).$$

Найдем значения функции для всех допустимых комбинаций c и x . Для упрощения расчетов значения x будем принимать кратными 20 тыс. ден. ед. и для большей наглядности записи оформлять в виде таблиц. Каждому шагу будет соответствовать своя таблица. Рассматриваемому шагу соответствует таблица 5.8.

Для каждого значения (20, 40, 60, 80, 100) начальной суммы c распределяемых средств в таблице 5.8 предусмотрена отдельная строка, а для каждого возможного значения x (0, 20, 40, 60, 80, 100) распределяемой суммы – столбец. Некоторые клетки таблицы останутся незаполненными, так как соответствуют недопустимым сочетаниям c и x . Такой, например, будет клетка, отвечающая строке $c=40$ и столбцу $x=80$, ибо при наличии 40 тыс. ден. ед. естественно отпадает вариант, при котором одному из предприятий выделяется 80 тыс. ден. ед.

Таблица 5.8 – Вычисление $f_2(c)$

$c \backslash x$	0	20	40	60	80	100	$f_2(c)$	$x_2^*(c)$
20	0+10	12+0					12	20
40	0+31	12+10	26+0				31	0
60	0+42	12+31	26+10	36+0			43	20
80	0+62	12+42	26+31	36+10	54+0		62	0
100	0+76	12+63	26+42	36+31	54+10	78+0	78	100

В каждую клетку таблицы будем вписывать значение суммы $g_2(x) + f_1(c - x)$. Первое слагаемое берем из условий задачи (см. таблицу 5.6), второе – из таблицы 5.7.

Так, например, при распределении начальной суммы $c = 80$ тыс. ден. ед. одним из вариантов может быть следующий: второму предприятию выделяется 60 тыс. ден. ед. ($x = 60$), тогда первому – $80 - 60 = 20$ тыс. ден. ед. При таком распределении первоначальной суммы на втором предприятии будет обеспечен прирост продукции на сумму в 36 тыс. ден. ед. (см. таблицу 5.6), на первом – 10 тыс. ден. ед. (см. таблицу 5.7).

Общий прирост составит $(36 + 10)$ тыс. ден. ед., что и записано в соответствующей клетке таблицы 5.8. В двух последних столбцах таблицы проставлены максимальный по строке прирост продукции (в столбце $f_2(c)$) и соответствующая ему оптимальная сумма средств, выделенная второму предприятию (в столбце $x_2^*(c)$). Так, при начальной сумме $c = 60$ тыс. ден. ед. максимальный прирост выпуска продукции составляет 43 тыс. ден. ед. ($12 + 31$), и это достигается выделением второму предприятию 20, а первому – $60 - 20 = 40$ тыс. ден. ед.

Этап 3. Расчет значений $f_3(c)$ приведен в таблице 5.9. Здесь использована формула, получающаяся из (5.9) при $n = 3$:

$$f_3(c) = \max_{0 \leq x \leq c} (g_3(x) + f_2(c - x)).$$

Первое слагаемое взято из таблицы 5.6, второе – из таблицы 5.8.

Таблица 5.9 – Вычисление $f_3(c)$

$c \backslash x$	0	20	40	60	80	100	$f_3(c)$	$x_3^*(c)$
20	0 + 12	11 + 0					12	0
40	0 + 31	11 + 12	36 + 0				36	40
60	0 + 43	11 + 31	36 + 12	45 + 0			48	40
80	0 + 62	11 + 43	36 + 31	45 + 12	60 + 0		67	40
100	0 + 78	11 + 62	36 + 43	45 + 31	60 + 12	77 + 0	79	40

Этап 4. Аналогичным образом находятся значения $f_4(c)$ (таблица 5.10). При расчётах использована формула

$$f_4(c) = \max_{0 \leq x \leq c} (g_4(x) + f_3(c - x)),$$

для которой первое слагаемое взято из таблицы 5.6, второе – из таблицы 5.9.

Таблица 5.10 – Вычисление $f_4(c)$

$c \backslash x$	0	20	40	60	80	100	$f_4(c)$	$x_4^*(c)$
20	0 + 12	16 + 0					16	20
40	0 + 36	16 + 12	37 + 0				37	40
60	0 + 48	16 + 36	37 + 12	46 + 0			52	20
80	0 + 67	16 + 48	37 + 36	46 + 12	63 + 0		73	40
100	0 + 79	16 + 67	37 + 48	46 + 36	63 + 12	80 + 0	85	40

Результаты решения задачи приведены в таблице (таблица 5.11), составленной на основе расчетных таблиц, начиная с таблицы 5.7.

Таблица 5.11 – Итоговые результаты решения задачи

c	$x_1^*(c)$	$f_1(c)$	$x_2^*(c)$	$f_2(c)$	$x_3^*(c)$	$f_3(c)$	$x_4^*(c)$	$f_4(c)$
20	20	10	20	12	0	12	20	16
40	40	31	0	31	40	36	40	37
60	60	42	20	43	40	48	20	52
80	80	62	0	62	40	67	40	73
100	100	76	100	78	40	79	40	85

Безусловная оптимизация.

Таблица 5.11 позволяет единообразно решать целый ряд задач. Например, из таблицы 5.11 видно, что наибольший прирост выпуска продукции, который могут дать четыре предприятия при распределении между ними 100 тыс. ден. ед. ($c = 100$), составляет 85 тыс. ден. ед. ($f_4(100) = 85$). При этом четвертому предприятию должно быть выделено 40 тыс. ден. ед. ($x_4^*(100) = 40$), а остальным трем – $100 - 40 = 60$ тыс. ден. ед. Из той же таблицы видно, что оптимальное распределение оставшихся 60 тыс. ден. ед. ($c = 60$) между тремя предприятиями обеспечит общий прирост продукции на них на сумму 48 тыс. ден. ед. ($f_3(60) = 48$) при условии, что третьему предприятию будет выделено 40 тыс. ден. ед. ($x_3^*(60) = 40$), а остальным двум – $60 - 40 = 20$ тыс. ден. ед. Оставшиеся 20 тыс. ден. ед. при оптимальном распределении между двумя предприятиями дадут прирост продукции на сумму в 12 тыс. ден. ед. ($f_2(20) = 12$). При этом второму предприятию нужно ассигновать 20 тыс. ден. ед. ($x_2^*(20) = 20$), а на долю первого средств не останется ($20 - 20 = 0$).

Итак, максимальный прирост выпуска продукции на четырех предприятиях при распределении между ними 100 тыс. ден. ед. составляет 85 тыс. ден. ед. и будет получен, если первому предприятию средств не выделять, второму выделить 20 тыс. ден. ед., а третьему и четвертому – по 40 тыс. ден. ед.

Предположим теперь, что 100 тыс. ден. ед. нужно распределить оптимально между тремя предприятиями. Из таблицы 5.9 находим: $f_3(100) = 79$ тыс. ден. ед.; прирост продукции на такую сумму может быть получен при $x_3^*(100) = 40$, т. е. если третьему предприятию ассигновать 40 тыс. ден. ед., а двум другим – $100 - 40 = 60$ тыс. ден. ед. Эти средства при оптимальном их распределении между двумя другими предприятиями обеспечат прирост выпуска продукции на сумму $f_2(60) = 43$ тыс. ден. ед. Но это возможно лишь в случае, если $x_2^*(60) = 20$, т. е. если второму предприятию будет выделено 20 тыс. ден. ед. Из таблицы 5.7 видно, что оставшиеся

$60 - 20 = 40$ тыс. ден. ед. следует выделить первому предприятию, так как $f_1(40) = 31$ при $x_1^*(40) = 40$.

При распределении 80 тыс. ден. ед. между тремя предприятиями: $x_1^* = 40$, $x_2^* = 0$, $x_3^* = 40$, при этом $f_3(80) = 67$.

5.5 Решение задачи о ремонте и замене оборудования методом динамического программирования

В процессе работы оборудование ежегодно дает прибыль, требует эксплуатационных затрат и имеет остаточную стоимость. Эти характеристики зависят от возраста оборудования. В любом году оборудование можно сохранить в прежнем виде или отремонтировать (продать по остаточной цене и приобрести новое).

В случае сохранения оборудования возрастают эксплуатационные расходы и понижается производительность. При ремонте нужны значительные дополнительные капитальные вложения.

Задача состоит в определении оптимальной стратегии ремонта в плановом периоде, с тем чтобы суммарная прибыль за этот период была максимальной.

Введем следующие обозначения:

- $r(t)$ – стоимость продукции, производимой за год на единице оборудования возраста t лет;
- $u(t)$ – расходы, связанные с эксплуатацией этого оборудования;
- $s(t)$ – остаточная стоимость оборудования возраста t лет;
- p – стоимость ремонта оборудования;
- T – продолжительность планового периода; $t = 0, 1, 2, \dots, T$ – номер текущего года.

Рассмотрим интервалы (годы) планового периода в последовательности от конца к началу.

Введем функцию условно-оптимальных значений функции цели $F_k(t)$. Эта функция показывает максимальную прибыль, получаемую от оборудования возраста t лет за последние k лет планового периода.

Здесь возраст оборудования рассматривается в направлении естественного хода времени. Например, $t = 0$ соответствует использованию отремонтированного оборудования.

Временные шаги процесса нумеруются в обратном порядке. Например, при $k = 1$ рассматривается последний год планового периода, при $k = 2$ – последние два года и т. д., при $k = T$ – последние T лет, т. е. весь плановый период.

Направления изменения t и k показаны на рисунке 5.2.

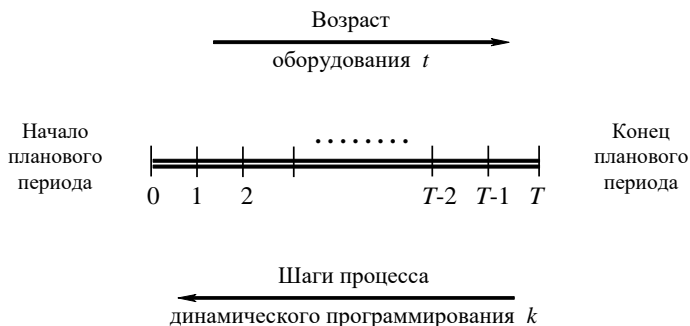


Рисунок 5.2 – Плановый период

Физической системой S является оборудование. Ее состояние характеризуется возрастом. Вектор управления – это решение в момент $t = 0, 1, 2, \dots$ о сохранении или ремонте оборудования.

Для нахождения оптимальной политики ремонта следует проанализировать, согласно принципу оптимальности, процесс от конца к началу. Для этого сделаем предположение о состоянии оборудования на начало последнего года ($k = 1$).

Пусть оборудование имеет возраст t лет. В начале T -го года имеется две возможности:

- 1) сохранить оборудование на T -й год, тогда прибыль за последний год составит $r(t) - u(t)$;
- 2) произвести ремонт оборудования, тогда прибыль за последний год будет равна $s(t) - p + r(0) - u(0)$, где $r(0)$ – стоимость продукции, выпущенной на отремонтированном оборудовании за первый год; $u(0)$ – эксплуатационные расходы в этом году.

Здесь целесообразно разворачивать процесс от конца к началу.

Для последнего года ($k = 1$) оптимальной политикой с точки зрения всего процесса будет политика, обеспечивающая максимальную прибыль только за последний год.

Учитывая значение прибыли при различном образе действия (ремонт – сохранение), приходим к выводу, что решение о ремонте оборудования возраста t лет следует принять в случае, когда прибыль от отремонтированного оборудования на последнем периоде больше, чем от старого, т. е. при условии

$$s(t) - p + r(0) - u(0) > r(t) - u(t).$$

Если же

$$s(t) - p + r(0) - u(0) \leq r(t) - u(t),$$

то оборудование целесообразно не ремонтировать.

Итак, для последнего года оптимальная политика и максимальная прибыль $F_1(t)$ находятся из условия

$$F_1(t) = \max_t \begin{cases} r(t) - u(t) & \text{(сохранение)} \\ s(t) - p + r(0) - u(0) & \text{(ремонт)}. \end{cases}$$

Пусть $k = 2$, т. е. рассмотрим прибыль за два последних года.

Делаем предположение о возможном состоянии t оборудования на начало предпоследнего года.

Если в начале этого года принять решение о сохранении оборудования, то к концу года будет получена прибыль $r(t) - u(t)$. На начало последнего года оборудование перейдет в состояние $t + 1$ и при оптимальной политике в последнем году оно принесет прибыль, равную $F_1(t + 1)$.

Таким образом, общая прибыль за два года составит $r(t) - u(t) + F_1(t + 1)$. Если же в начале предпоследнего года будет принято решение о ремонте оборудования, то прибыль за предпоследний год составит $s(t) - p + r(0) - u(0)$. Поскольку произведен капитальный ремонт старого оборудования, то на начало последнего года оно будет в состоянии $t = 1$. Следовательно, общая прибыль за последние два года при оптимальной политике в последнем году составит:

$$s(t) - p + r(0) - u(0) + F_1(t + 1).$$

Условно оптимальной в последние два года будет политика, доставляющая максимальную прибыль,

$$F_2(t) = \max_t \begin{cases} r(t) - u(t) + F_1(t + 1) & \text{(сохранение)} \\ s(t) - p + r(0) - u(0) + F_1(1) & \text{(ремонт)}. \end{cases}$$

Аналогично находим выражения для условно-оптимальной прибыли за три последних года, четыре и т. д. Общее функциональное уравнение примет вид:

$$F_k(t) = \max_t \begin{cases} r(t) - u(t) + F_{k-1}(t + 1) & \text{(сохранение)} \\ s(t) - p + r(0) - u(0) + F_{k-1}(1) & \text{(ремонт)}. \end{cases}$$

При $k = T$ получим $\max z = F_T(t_0)$, причем

$$F_T(t_0) = \max_t \begin{cases} r(t_0) - u(t_0) + F_{T-1}(t_0 + 1) & \text{(сохранение)} \\ s(t_0) - p + r(0) - u(0) + F_{T-1}(1) & \text{(ремонт)}. \end{cases}$$

Таким образом, разворачивая весь процесс от конца к началу, получаем, что максимальная прибыль за плановый период T составит $F_T(t_0)$. Так как начальное состояние t_0 известно, из выражения для $F_T(t_0)$ находим оптимальное решение в начале первого года, потом вытекающее из него оптимальное решение для второго года и т. д.

Задача о капитальном ремонте старого оборудования была упрощена.

При проведении капитального ремонта необходимо учесть, что в понятие «состояние» системы надо включить время последнего ремонта оборудования. Тогда функция $F_k(t_1, t_2)$ выражает прибыль за последние k лет планового периода при условии, что вначале имелось оборудование возраста t_1 , прошедшее капитальный ремонт после t_2 лет службы. Характеристики r , s и u также будут функциями двух переменных t_1 и t_2 .

Пример 5.3. Разработать оптимальную политику замены оборудования при условии:

1) стоимость $r(t)$ продукции, производимой с использованием оборудования за год, и расходы $u(t)$, связанные с эксплуатацией оборудования, заданы в таблице 5.12;

2) ликвидационная стоимость оборудования не зависит от ее возраста и равна 4 усл. ед.;

3) цена нового оборудования со временем не меняется и равна 30 усл. ед.;

4) продолжительность планового периода 12 лет.

Таким образом, $s(t) = 4$; $p = 30$; $T = 12$ лет.

Таблица 5.12 – Исходные данные

r, u \ t	0	1	2	3	4	5	6	7	8	9	10	11	12
$r(t)$	60	60	58	58	58	56	56	54	54	52	48	46	46
$u(t)$	20	20	24	26	28	30	32	34	36	38	40	44	46

Для исходных данных запишем функциональные уравнения для $F_1(t)$ и $F_k(t)$:

$$\begin{aligned}
 F_1(t) &= \max_t \left\{ \begin{array}{l} r(t) - u(t) \\ s(t) - p + r(0) - u(0) \end{array} \right\} = \max_t \left\{ \begin{array}{l} r(t) - u(t) \\ 4 - 30 + 60 - 20 \end{array} \right\} = \\
 &= \max_t \left\{ \begin{array}{ll} r(t) - u(t) & \text{(сохранение)} \\ 14 & \text{(замена);} \end{array} \right. \quad (5.10)
 \end{aligned}$$

$$F_k(t) = \max_t \left\{ \begin{array}{ll} r(t) - u(t) + F_{k-1}(t+1) & \text{(сохранение)} \\ 14 + F_{k-1}(1) & \text{(замена).} \end{array} \right. \quad (5.11)$$

Используя формулы (5.10)–(5.11), будем последовательно вычислять значения максимальной прибыли $F_k(t)$ и записывать их в таблицу 5.13.

Первую строку таблицы 5.13 получим, придавая параметру t в равенстве (5.10) значения 0, 1, ..., 12 и используя исходные данные таблицы 5.12.

$$F_1(0) = \max \left\{ \frac{r(0) - u(0)}{14} \right\} = \max \left\{ \frac{60 - 20}{14} \right\} = 40 \text{ (сохранение);}$$

$$F_1(1) = \max \left\{ \frac{r(1) - u(1)}{14} \right\} = \max \left\{ \frac{60 - 20}{14} \right\} = 40 \text{ (сохранение);}$$

$$F_1(2) = \max \left\{ \frac{r(2) - u(2)}{14} \right\} = \max \left\{ \frac{58 - 24}{14} \right\} = 34 \text{ (сохранение);}$$

...

$$F_1(9) = \max \left\{ \frac{r(9) - u(9)}{14} \right\} = \max \left\{ \frac{52 - 38}{14} \right\} = 14 \text{ (сохранение);}$$

$$F_1(10) = \max \left\{ \frac{r(10) - u(10)}{14} \right\} = \max \left\{ \frac{48 - 40}{14} \right\} = 14 \text{ (замена);}$$

...

Таблица 5.13 – Итоговые результаты решения задачи

$F_k \backslash t$	0	1	2	3	4	5	6	7	8	9	10	11	12
$F_1(t)$	40	40	34	32	30	26	24	20	18	14	14	14	14
$F_2(t)$	80	74	66	62	56	54	54	54	54	54	54	54	54
$F_3(t)$	114	106	96	88	88	88	88	88	88	88	88	88	88
$F_4(t)$	146	136	122	120	120	120	120	120	120	120	120	120	120
$F_5(t)$	176	162	154	152	150	150	150	150	150	150	150	150	150
$F_6(t)$	202	194	186	182	180	176	176	176	176	176	176	176	176
$F_7(t)$	234	226	216	212	208	208	208	208	208	208	208	208	208
$F_8(t)$	266	256	246	240	240	240	240	240	240	240	240	240	240
$F_9(t)$	296	286	274	272	270	270	270	270	270	270	270	270	270
$F_{10}(t)$	332	314	306	302	300	300	300	300	300	300	300	300	300
$F_{11}(t)$	354	346	336	332	330	328	328	328	328	328	328	328	328
$F_{12}(t)$	386	376	366	362	360	360	360	360	360	360	360	360	360

Если прибыль от нового оборудования равна прибыли от старого, то старое оборудование лучше сохранить еще на год.

Из таблицы 5.13 видно, что $r(t) - u(t)$ с ростом t убывает. Поэтому при $t > 9$ оптимальной будет политика замены оборудования.

Чтобы наглядно различать, в результате какой политики получается условно-оптимальное значение прибыли, будем эти значения (до $t = 9$ включительно оптимальной является политика сохранения) разграничивать жирной линией.

Для заполнения второй строки таблицы 5.13 используем формулу

$$F_2(t) = \max_t \left\{ \begin{array}{c} r(t) - u(t) + F_1(t+1) \\ 14 + F_1(t) \end{array} \right\} = \max_t \left\{ \begin{array}{c} r(t) - u(t) + F_1(t+1) \\ 54 \end{array} \right\} \quad \begin{array}{l} \text{(сохранение)} \\ \text{(замена).} \end{array}$$

Придадим параметру t значения 0, 1, 2, ..., 12; $r(t)$ и $u(t)$ возьмем из таблицы 5.12, а значения $F_1(t+1)$ – из первой строки таблицы 5.13. Получим вторую строку:

$$F_2(0) = \max_t \left\{ \begin{array}{c} r(0) - u(0) + F_1(1) \\ 54 \end{array} \right\} = \max_t \left\{ \begin{array}{c} 60 - 20 + 40 \\ 54 \end{array} \right\} = 80 \text{ (сохранение);}$$

$$F_2(1) = \max_t \left\{ \begin{array}{c} r(1) - u(1) + F_1(2) \\ 54 \end{array} \right\} = \max_t \left\{ \begin{array}{c} 60 - 20 + 34 \\ 54 \end{array} \right\} = 74 \text{ (сохранение);}$$

$$F_2(2) = \max_t \left\{ \begin{array}{c} r(2) - u(2) + F_1(3) \\ 54 \end{array} \right\} = \max_t \left\{ \begin{array}{c} 58 - 24 + 32 \\ 54 \end{array} \right\} = 66 \text{ (сохранение);}$$

$$F_2(3) = \max_t \left\{ \begin{array}{c} r(3) - u(3) + F_1(4) \\ 54 \end{array} \right\} = \max_t \left\{ \begin{array}{c} 58 - 26 + 30 \\ 54 \end{array} \right\} = 62 \text{ (сохранение);}$$

...

Для третьей строки расчетные формулы получим из равенства (5.11) при $k = 3$:

$$F_3(t) = \max_t \left\{ \begin{array}{c} r(t) - u(t) + F_2(t+1) \\ 14 + F_2(t) \end{array} \right\} = \max_t \left\{ \begin{array}{c} r(t) - u(t) + F_2(t+1) \\ 88 \end{array} \right\} \quad \begin{array}{l} \text{(сохранение)} \\ \text{(замена).} \end{array}$$

и т. д.

Заполнив таблицу 5.13, ее данные используем для решения поставленной задачи. Таблица 5.13 позволяет решить всё семейство задач, в которое мы погружали исходную.

Пусть, например, в начале планового периода имеем оборудование возраста 6 лет. Рассмотрим «политику замен» в течение 12 лет, доставляющую максимальную прибыль. Информация об этой задаче имеется в таблице 5.13.

Величина максимальной прибыли, которую можно получить за 12 лет при условии, что вначале имелось оборудование возраста 6 лет, находится в таблице 5.13 в клетке, где записано $F_{12}(6)$. Она составляет 360 единиц.

Значение максимальной прибыли $F_{12}(6) = 360$ записано справа от жирной линии, т. е. в области «политики замены». Это значит, что для достижения в течение 12 лет максимальной прибыли в начале первого года оборудование надо заменить. В течение первого года новое оборудование постареет на год, т. е., заменив оборудование и проработав на нем 1 год, за 11 лет до конца планового периода будем иметь оборудование возраста 1 год.

Из таблицы 5.13 берем $F_{11}(1) = 346$. Это значение располагается в области «политики сохранения», т. е. во втором году планового периода надо сохранить оборудование возраста 1 год, и, проработав на нем год, за 10 лет до конца планового периода будем иметь оборудование возраста 2 года.

Определяем $F_{10}(2) = 206$. Оно помещено в области сохранения. Работаем на оборудовании еще год. Теперь до конца планового периода осталось 9 лет, а возраст оборудования станет 3 года. Находим $F_9(3) = 272$. Это область сохранения. Работаем на оборудовании еще год. Его возраст становится равным 4 годам. До конца планового периода остается 16 лет. Определяем $F_8(4) = 240$. Это область замен. Заменяем оборудование на новое. Проработаем на нем в течение четвертого года. Оно постареет на год. До конца планового периода останется 7 лет. Находим $F_7(1) = 113$. Это область сохранения. Продолжая подобные рассуждения, установим, что $F_6(2) = 186$, $F_5(3) = 152$ – область сохранения, $F_4(4) = 120$ будет в области замен. $F_3(1) = 106$ – область сохранения, $F_2(2) = 66$, $F_1(3) = 32$ – область сохранения.

Рассмотренный процесс формирования оптимального плана замены оборудования можно изобразить на рисунке 5.3.

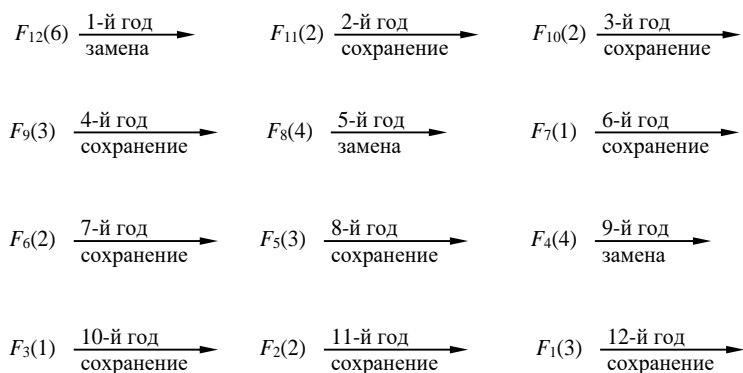


Рисунок 5.3 – Оптимальный план замены оборудования

Таким образом, вместо поиска оптимальной «политики замен» на плановый период в 12 лет погрузили исходную задачу в семейство подобных, когда период меняется от 1 до 12.

Решение ведется по принципу оптимальности для любого состояния системы, независимо от ее предыстории.

Оптимальная «политика замен» является оптимальной на оставшееся число лет. Таблица 5.13 содержит много ценной информации для реализации и других задач. Из нее можно найти оптимальную стратегию замены оборудования с любым начальным состоянием от 0 до 12 лет и на любой плановый период, не превосходящий 12 лет.

Например, найдем «политику замен» на плановый период 10 лет, если вначале имелось оборудование 5-летнего возраста (рисунок 5.4).

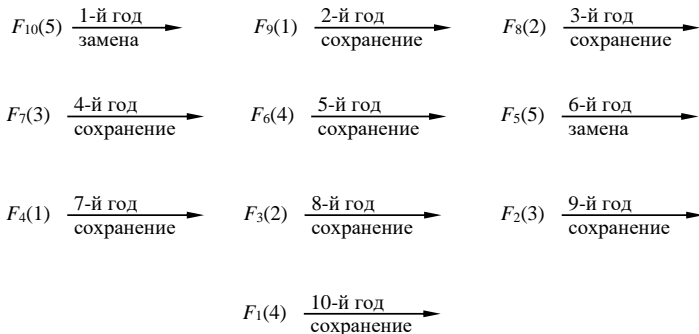


Рисунок 5.4 – Оптимальный план замены оборудования

При решении задачи о замене оборудования, может быть принято решение о замене оборудования не новым, а уже проработавшим некоторое время или учесть возможность капитального ремонта старого оборудования.

Задачу о капитальном ремонте старого оборудования мы упростили. На практике же деталями не пренебрегают. При проведении капитального ремонта необходимо учесть, что в понятие «состояние» системы надо включить время последнего ремонта оборудования. Тогда функция $F_k(t_1, t_2)$ выражает прибыль за последние k лет планового периода при условии, что вначале имелось оборудование возраста t_1 , прошедшее капитальный ремонт после t_2 лет службы. Характеристики r , s и u также будут функциями двух переменных t_1 и t_2 .

Необходимо также учесть, например, случай, когда остаточная стоимость оборудования $s(t)$ зависит от времени, а также, что может быть принято решение не о ремонте, а о замене оборудования новым или уже проработавшим некоторое время.

СПИСОК ЛИТЕРАТУРЫ

1 Identification of the railway reconstruction parameters at imposition of high speed traffic on the existing lines / I. Lebid [et al.] // MATEC Web of Conferences. – V. 294. – 2019. – Article 05003.

2 **Kravchenya, I.** Application of Computer Optimization Methods in the Reconstruction of Railway Lines / I. Kravchenya, T. Dubrovskaya // International Journal of Advanced Multidisciplinary Research and Studies. – V. 3, is. 1. – India, 2023. – P. 585–589.

3 Optimization of public transport schedule on duplicating stretches / I. Lebid [et al.] // Mechatronic Systems 1: Applications in Transport, Logistics, Diagnostics, and Control, Chapter 18. – London, 2021. – P. 209–220.

4 Rationing of the number of signals and interlockings in the operational stock of railway stations / I. Lebid [et al.] // MATEC Web of Conferences. – V. 294. – 2019. – Article 05007.

5 Scheduling technique of route vehicles on duplicating stretches / S. Azemsha [et al.] // Scientific Journal of Silesian University of Technology. Series Transport. – V. 113. – 2021. – P. 5–16.

6 **Аземша, С. А.** Оценка эффективности оптимизации расписания движения городского пассажирского транспорта на дублирующих участках / С. А. Аземша, И. Н. Кравчяня // Вестник СибАДИ, 2021. – Т. 18, № 1. – С. 72–85.

7 **Афанасьев, М. Ю.** Исследование операций в экономике : учеб. пособие / М. Ю. Афанасьев, Б. П. Суворов. – М. : ИНФРА-М, 2006. – 444 с.

8 **Бочаров, Д. И.** Применение методов математического моделирования при решении производственных задач : учеб.-метод. пособие / Д. И. Бочаров, И. Н. Кравчяня. – Гомель : БелГУТ, 2009. – 191 с.

9 **Бурдук, Е. Л.** Исследование операций : учеб.-метод. пособие / Е. Л. Бурдук, И. Н. Кравчяня. – Гомель : БелГУТ, 2008. – 80 с.

10 **Вентцель, Е. С.** Исследование операций. Задачи, принципы, методология : учеб. пособие / Е. С. Вентцель. – М. : Высш. шк., 2001. – 208 с.

11 Исследование операций в экономике : учеб. пособие для вузов / Н. Ш. Кремер [и др.] ; под ред. проф. Н. Ш. Кремера. – М. : ЮНИТИ, 2003. – 407 с.

12 **Косоруков, О. А.** Исследование операций : учеб. / О. А. Косоруков, А. В. Мищенко ; под общ. ред. д-ра экон. наук, проф. Н. П. Тихомирова. – М. : Экзамен, 2003. – 448 с.

12 **Костевич, Л. С.** Исследование операций. Теория игр / Л. С. Костевич, А. А. Лапко. – Минск : Выш. шк., 2008. – 367 с.

13 **Костевич, Л. С.** Математическое программирование: Информационные технологии оптимальных решений : учеб. пособие / Л. С. Костевич. – Минск : Новое знание, 2003. – 424 с.

14 **Кравченя, И. Н.** Математическое моделирование. Линейное и нелинейное программирование, сетевое планирование и управление : учеб.-метод. пособие / И. Н. Кравченя, Е. Л. Бурдук, Т. В. Алымова. – Гомель : БелГУТ, 2014. – 112 с.

15 **Кравченя, И. Н.** Оптимизация расписания городского общественного транспорта разных маршрутов на дублирующих участках / И. Н. Кравченя, А. М. Подколзин // Организация и безопасность дорожного движения. – Тюмень : ТИУ, 2019. – Т. 2. – С.54–61.

16 **Кузнецов, А. В.** Высшая математика. Математическое программирование / А. В. Кузнецов, В. А. Сакович, Н. И. Холод. – Минск : Выш. шк., 2001. – 351 с.

17 **Кузнецов, А. В.** Руководство к решению задач по математическому программированию : учеб. пособие / А. В. Кузнецов, Н. И. Холод, Л. С. Костевич. – Минск : Выш. шк., 2001. – 448 с.

18 **Лебедева, Г. И.** Прикладная математика. Математические модели в транспортных системах : учеб. пособие / Г. И. Лебедева, Н. А. Микулик. – Минск : Асар, 2009. – 512 с.

19 Оптимизация в дорожном движении : учеб.-метод. пособие / В. Н. Галушко [и др.]. – Гомель : БелГУТ, 2021. – 185 с.

20 **Плотников, А. Д.** Математическое программирование : экспресс-курс / А. Д. Плотников. – Минск : Новое знание, 2007. – 171 с.