МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный
университет им. Н. И. Лобачевского»

**Ю. А. Кузнецов, А. В. Семенов**

# МЕТОДЫ ОПТИМИЗАЦИИ: ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Учебно-методическое пособие

Рекомендовано методической комиссией
Института экономики и предпринимательства
для студентов ННГУ, обучающихся по направлению подготовки
38.03.05 «Бизнес-информатика», квалификация – бакалавр

Нижний Новгород
2022

УДК 519.852
ББК 22.18
К89

К-89 Кузнецов Ю. А., Семенов А. В. МЕТОДЫ ОПТИМИЗАЦИИ: ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ: Учебно-методическое пособие. Нижний Новгород: Нижегородский госуниверситет, 2022. – 43 с.

Рецензент: к.ф.-м.н., доцент **А. А. Тюхтина**

Учебно-методическое пособие посвящено основам линейного программирования, а также вопросам его применения к исследованию экономических задач. Приводятся примеры и задачи для самостоятельного решения.

Пособие предназначено для преподавателей ИЭП ННГУ, ведущих дисциплины «Методы оптимизации», «Методы оптимальных решений», «Исследование операций» и студентов, обучающихся по направлению подготовки 38.03.05 «Бизнес-информатика», бакалаврской образовательной программы «Аналитические методы и информационные технологии поддержки принятия решений в экономике и бизнесе».

*Ответственный за выпуск:*
председатель методической комиссии ИЭП ННГУ,
к.э.н., доцент **Макарова С. Д.**

УДК 519.852
ББК 22.18

**СОДЕРЖАНИЕ**

## Введение

В августе 1939 года в Ленинграде вышла монография Л.В. Канторовича «Математические методы организации и планирования производства». Отправной точкой написания данной работы явилось исследование одной задачи, предложенной Институту математики и механики ЛГУ лабораторией Фанерного треста. Задача заключалась в следующем. Имеется несколько станков разной производительности, на которых нужно изготовить определенное количество заготовок. Как наилучшим образом распределить работу по станкам, чтобы максимально быстро выполнить план?

Л.В. Канторович обнаружил, что целый ряд проблем, относящихся к научной организации производства самого разнообразного характера (вопросы наилучшего распределения работы станков и механизмов, максимального уменьшения отходов, наилучшего использования сырья, транспорта и пр.) приводят к одной и той же группе экстремальных задач. Термин «линейное программирование» появился в 1951 в работах американских учёных (Дж. Б. Данциг, Т. Купманс). Слово программирование выбрано потому, что набор переменных, подлежащих нахождению, обычно определяет программу (план) работы некоторого экономического объекта.

# §1. Основы линейного программирования

*Задачей линейного программирования* называется задача минимизации или максимизации линейной функции при линейных ограничениях.

## 1.1. Примеры задач линейного программирования

***Производственная задача***

Пусть имеется $m$ видов ресурсов в количестве $b_1, b_2, \ldots, b_m$, которые могут быть использованы при производстве $n$ видов изделий. Известны величины $a_{ij}$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$, характеризующие расход $i$-го ресурса на производство единицы $j$-го изделия. Предполагается, что технология производства линейна, т.е. затраты ресурсов растут прямо пропорционально объему производству. Пусть известен доход $c_j$ от реализации единицы $j$-го изделия. Производственная задача состоит в поиске плана выпуска изделий, максимизирующего суммарный доход.

Обозначив через $x_j$ $(j = 1, \ldots, n)$ величину выпуска $j$-го изделия, получим задачу линейного программирования

$$\sum_{j=1}^{n} c_j x_j \to \max, \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1, \ldots, m, \quad x_j \ge 0, \quad j = 1, \ldots, n.$$

***Задача о рационе***

Пусть имеется $n$ продуктов питания (хлеб, мясо, молоко, картофель и т.п.) и $m$ полезных веществ (жиры, белки, углеводы, витамины и т.п.). Известны следующие параметры:

$a_{ij}$ – содержание $i$-го вещества в единице $j$-го продукта;

$c_j$ – цена единицы $j$-го продукта;

$b_i$ – потребность определенного индивидуума в $i$-м веществе (в расчете, скажем, на месяц).

Обозначив через $x_j$ потребление индивидуумом $j$-го продукта, получаем задачу о выборе наиболее дешевого рациона питания, удовлетворяющего всем потребностям:

$$\sum_{j=1}^{n} c_j x_j \to \min, \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m, \quad x_j \geq 0, \quad j = 1, \ldots, n.$$

***Транспортная задача***

Сущность транспортной задачи линейного программирования состоит в наивыгоднейшем прикреплении поставщиков однородного продукта ко многим потребителям этого продукта.

Предполагается, что в пунктах отправления (предприятиях-изготовителях, складах и т.д.) имеется однородный продукт, причем в пункте $i$ в количестве $a_i$ единиц. Этот продукт следует распределить между пунктами назначения (предприятиями-потребителями), при этом потребность пункта $j$ составляет $b_j$ единиц, так, чтобы минимизировать транспортные издержки: $c_{ij}$ – затраты на перевозку единицы груза из $i$ в $j$. Пусть $x_{ij}$ объем перевозки из $i$ в $j$. Таким образом приходим к так называемой замкнутой транспортной задаче:

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

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

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

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

Для разрешимости такой задачи необходимо и достаточно выполнения соотношения

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

## 1.2. Формы записи задачи линейного программирования

Существует ряд специальных форм записи задачи линейного программирования. Задача в форме

$$\begin{gathered}\sum_{j=1}^{n} c_j x_j \to \min, \\ \sum_{j=1}^{n} a_{ij} x_j \ge b_i, \quad i = 1,\ldots,k, \\ \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = k+1,\ldots,m, \\ x_j \ge 0, \quad j = 1,\ldots,s,\end{gathered} \tag{1.1}$$

называется ***общей*** задачей линейного программирования, в форме

$$\sum_{j=1}^{n} c_j x_j \to \min, \qquad \sum_{j=1}^{n} a_{ij} x_j \ge b_i, \quad i = 1,\ldots,m, \tag{1.2}$$

– ***основной***, в форме

$$\begin{gathered}\sum_{j=1}^{n} c_j x_j \to \min, \quad \sum_{j=1}^{n} a_{ij} x_j \ge b_i, \quad i = 1,\ldots,m, \\ x_j \ge 0, \quad j = 1,\ldots,n,\end{gathered} \tag{1.3}$$

– ***стандартной***, в форме

$$\begin{gathered}\sum_{j=1}^{n} c_j x_j \to \min, \quad \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,\ldots,m, \\ x_j \ge 0, \quad j = 1,\ldots,n,\end{gathered} \tag{1.4}$$

– ***канонической***. Здесь $c = (c_1, \ldots, c_n) \in \mathbf{R}^n$, $b = (b_1, \ldots, b_m) \in \mathbf{R}^m$, $A$ – матрица размера $m \times n$ со строками $a_1, \ldots, a_m$ и элементами $a_{ij}$.

Для задач (1.1)-(1.4) используют также векторно-матричные записи. Например, задачу (1.2) можно записать в виде

$$\langle c, x \rangle \to \min, \quad \langle a_i, x \rangle \ge b_i, \quad i = 1,\ldots,m,$$

или в виде

$$\langle c, x \rangle \to \min, \quad Ax \ge b. \tag{1.5}$$

Формально, задачи (1.2)-(1.4) являются частными случаями общей задачи (1.1): при $k = m$, $s = 0$ получается (1.2), при $k = m$, $s = n$ – (1.3), при $k = 0$, $s = n$

– (1.4). Однако и общая задача может быть представлена в форме любой из трёх остальных. Так, задача (1.1) принимает основную форму, если заменить в ней систему ограничений-равенств на эквивалентную систему ограничений-неравенств

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad -\sum_{j=1}^{n} a_{ij} x_j \geq -b_i, \qquad i = k+1, \ldots, m. \tag{1.6}$$

Если при этом сделать замену переменных

$$x_j = x_j^1 - x_j^2, \quad x_j^1 \geq 0, \quad x_j^2 \geq 0, \quad j = s+1, \ldots, n, \tag{1.7}$$

то задача (1.1) примет стандартную форму. Если же ограничения-неравенства в (1.1) записать в виде

$$\sum_{j=1}^{n} a_{ij} x_j - u_i = b_i, \quad u_i \geq 0, \quad i = 1, \ldots, k, \tag{1.8}$$

где $u_i$ – дополнительная переменная (формально входящая в целевую функцию с нулевым коэффициентом), и вновь использовать замену переменных (1.7), то задача (1.1) превратится в каноническую.

Таким образом, любую задачу линейного программирования на минимум или максимум можно представить в любой из указанных форм. Для этого, наряду с приёмами (1.6)-(1.8), необходимо использовать умножение целевой функции или ограничений-неравенств на $-1$, что позволяет переходить от максимизации к минимизации и менять знаки неравенств.

## 1.3. Геометрическая интерпретация задач линейного программирования

Приведем геометрическую интерпретацию задач линейного программирования на примере задачи в основной форме (1.5). В рассматриваемом случае множество $X$ допустимых точек $x$ представляет собой полиэдр, образованный пересечением полупространств, задаваемых ограничениями задачи. Отметим, что при исследовании задач экономики допустимые точки $x$ называют также *планами*. Линии уровня целевой функции $f(x) = \langle c, x \rangle$ образуют семейство параллельных гиперплоскостей $H_{c\alpha} = \{x \in \mathbf{R}^n \mid \langle c, x \rangle = \alpha\}$ $(\alpha \in \mathbf{R})$ (Рис. 1.1-1.3). Градиент целевой функции всюду одинаков ($f'(x) = c$) и является нормалью каждой из данных гиперплоскостей. Таким образом, поиск решения задачи сводится к нахождению максимального числа $\alpha^*$ среди всех $\alpha$ таких, что гиперплоскость $H_{c\alpha}$ имеет непустое пересечение с

$X$.Тогда $X^* = H_{c\alpha^*} \cap X$ – множество решений задачи.

Следует отметить, что в рассматриваемой задаче ЛП на минимум при поиске $\alpha^*$ проходит как бы перемещение гиперплоскости $H_{c\alpha}$ в направлении, противоположном вектору $c$. Если же решается задача ЛП на максимум и, стало быть, ищется максимальное $\alpha^*$, удовлетворяющее указанным требованиям, то $H_{c\alpha}$ перемещается в направлении вектора $c$.

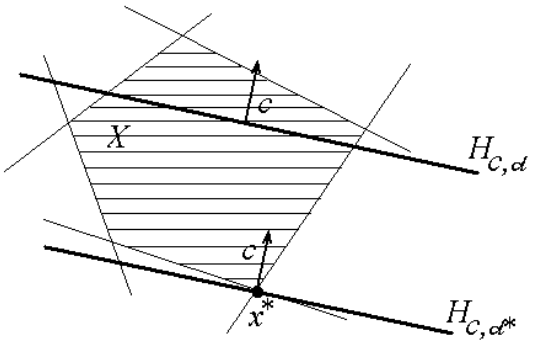


Рис. 1.1

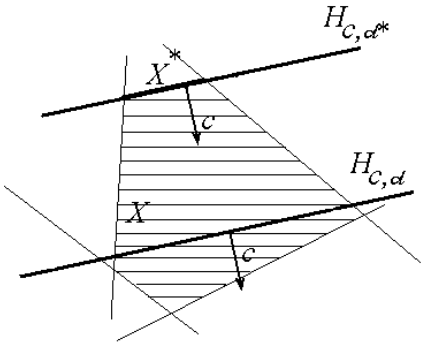


Рис. 1.2

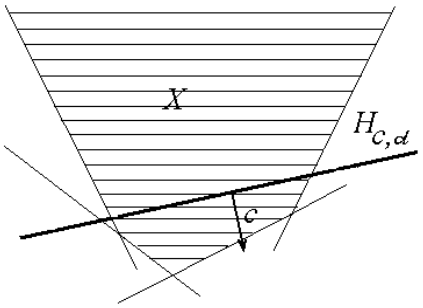


Рис. 1.3

Ясна характерная особенность задачи линейного программирования: если её решение существует, то оно достигается обязательно на границе. При этом, либо решение единственно (Рис. 1.1), либо множество решений бесконечно (Рис. 1.2). При неограниченном $X$ решений может и не быть, то есть $H_{c\alpha} \cap X = \varnothing$ (Рис. 1.3).

Имеет место следующая

**Теорема 1.1.** *Пусть задача линейного программирования имеет решение, причём её допустимое множество обладает, по крайней мере, одной вершиной. Тогда существует такая вершина, которая служит решением задачи. В частности, если решение единственно, то оно обязательно является вершиной.*

Таким образом, если задача линейного программирования на максимум удовлетворяет условиям теоремы 1.1, то для отыскания её решения достаточно

найти все вершины допустимого множества задачи (число которых конечно) и выбрать из них ту, в которой функция принимает максимальное значение.

Описанный метод полного перебора вершин практически применим лишь для малоразмерных задач линейного программирования, поскольку при большом числе переменных и ограничений он требует огромной вычислительной работы. Тем не менее, именно идея перебора вершин (но не полного, а упорядоченного) лежит в основе численного метода решения задач линейного программирования, называемого симплекс-методом.

**Пример 1.1.** Используя геометрические построения найти решение задачи:

$$\begin{aligned} x_1 + 5x_2 &\to \max \\ x_1 + 2x_2 &\le 10, \\ 3x_1 + 2x_2 &\le 18, \\ x_1 - x_2 &\ge -7, \\ 2x_1 - x_2 &\le 19. \end{aligned}$$

Первым шагом на координатной плоскости строится допустимое множество $X$ (Рис.1.4). Далее – вектор $c$ и линия уровня $H_{c\alpha}$ целевой функции, перпендикулярная $c$. Т.к. решается задача максимизации, то построенную линию уровня передвигаем в направлении вектора $c$ до тех пор, пока она не покинет пределов множества $X$. В результате находим точку $x^*$, в которой целевая функция принимает максимальное значение. Данная точка является пересечением прямых (1) и (3), т.е. является решением системы

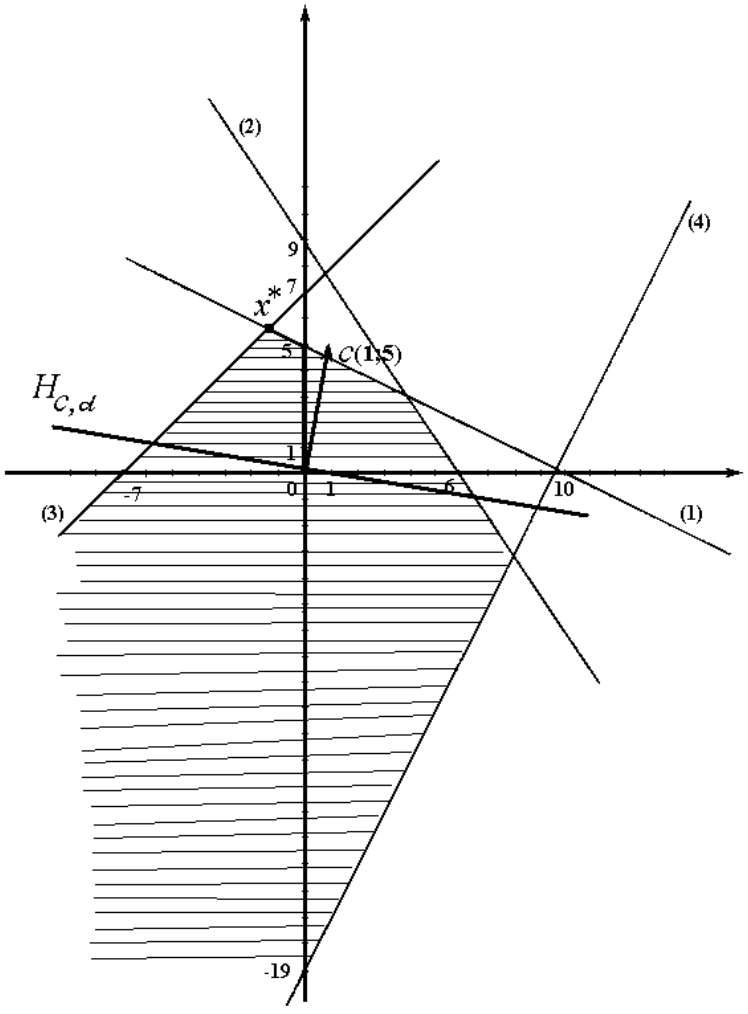


Рис. 1.4

$$\begin{aligned} x_1 + 2x_2 &= 10, \\ x_1 - x_2 &= -7. \end{aligned}$$

Решая систему, находим координаты точки

максимума $x_1^* = -4/3$, $x_2^* = 17/3$, после чего – максимальное значение целевой функции $f^* = 27$.

## 1.4. Двойственность в задачах линейного программирования

Каждой задаче линейного программирования можно поставить в соответствие другую задачу линейного программирования, называемую двойственной. Их совместное изучение составляет предмет теории двойственности, являющейся важным разделом линейного программирования.

Рассмотрим общую задачу линейного программирования (1.1). *Двойственной* к задаче (1.1) является следующая задача:

$$\begin{gathered} \sum_{i=1}^{m} p_i b_i \to \max, \\ \sum_{i=1}^{m} p_i a_{ij} \le c_j, \quad j = 1,\ldots,s, \\ \sum_{i=1}^{m} p_i a_{ij} = c_j, \quad j = s+1,\ldots,n, \\ p_i \ge 0, \quad i = 1,\ldots,k. \end{gathered} \tag{1.9}$$

При этом задача (1.1) называется *прямой*.

Проанализируем механизм построения двойственной задачи (1.9). Прежде всего, – это задача максимизации, тогда как (1.1) – задача минимизации. Коэффициенты $c_j$ целевой функции задачи (1.1) служат коэффициентами правых частей ограничений задачи (1.9), а коэффициенты $b_i$ правых частей ограничений задачи (1.1) – коэффициентами целевой функции задачи (1.9). Если в задаче (1.1) матрица коэффициентов $a_{ij}$ умножается на вектор переменных $x_j$ справа, то в задаче (1.9) она умножается на вектор переменных $p_i$ слева. Имеется взаимно однозначное соответствие между ограничениями задачи (1.9) и переменными задачи (1.1): $j$-е ограничение задачи (1.9) является неравенством (обратного, нежели в (1.1), знака), если на $j$-ю переменную задачи (1.1) наложено условие неотрицательности; в противном случае $j$-е ограничение задачи (1.9) является равенством. Аналогичное соответствие имеется между переменными задачи (1.9) и ограничениями задачи (1.1).

Особо можно отметить, как выглядят двойственные задачи к задачам линейного программирования в упомянутых ранее частных формах.

Двойственной к задаче линейного программирования в основной форме

$$\sum_{j=1}^{n} c_j x_j \to \min, \qquad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m,$$

является задача линейного программирования в канонической форме

$$\sum_{i=1}^{m} b_i p_i \to \max, \quad \sum_{i=1}^{m} p_i a_{ij} = c_j, \quad j = 1, \ldots, n,$$
$$p_i \geq 0, \quad i = 1, \ldots, m.$$

Двойственной к задаче линейного программирования в стандартной форме

$$\sum_{j=1}^{n} c_j x_j \to \min, \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m,$$
$$x_j \geq 0, \quad j = 1, \ldots, n,$$

является задача линейного программирования также в стандартной форме

$$\sum_{i=1}^{m} b_i p_i \to \max, \quad \sum_{i=1}^{m} p_i a_{ij} \leq c_j, \quad j = 1, \ldots, n,$$
$$p_i \geq 0, \quad i = 1, \ldots, m.$$

Двойственной к задаче линейного программирования в канонической форме

$$\sum_{j=1}^{n} c_j x_j \to \min, \quad \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m,$$
$$x_j \geq 0, \quad j = 1, \ldots, n,$$

является задача линейного программирования в основной форме

$$\sum_{i=1}^{m} b_i p_i \to \min, \qquad \sum_{i=1}^{m} p_i a_{ij} \leq c_j, \quad j = 1, \ldots, n.$$

Далее можно рассмотреть основные факты теории двойственности для задач линейного программирования.

**Теорема 1.2.** *Задачи* (1.1) *и* (1.9) *взаимодвойственны, т.е. с точностью до эквивалентной формы записи двойственной к задаче* (1.9) *является задача* (1.1).

Ниже под $X$ и $P$ понимаются допустимые множества задач (1.1) и (1.9) соответственно.

**Теорема 1.3.** *Для любых* $x \in X$ *и* $p \in P$ *выполняется неравенство*

$$\langle c, x\rangle \geq \langle b, p\rangle.$$

**Теорема 1.4.** *Пусть* $x^* \in X$, $p^* \in P$. *Тогда*

1) *если*

$$\langle c, x^*\rangle = \langle b, p^*\rangle, \tag{1.10}$$

*то* $x^*$ *– решение задачи* (1.1), *а* $p^*$ *– решение задачи* (1.9);

2) *равенство* (1.21) *равносильно совокупности условий*

$$x_j^*\left(\sum_{i=1}^{m} p_i^* a_{ij} - c_j\right) = 0, \qquad j = 1,\ldots,s, \tag{1.11}$$

$$p_i^*\left(\sum_{j=1}^{n} a_{ij} x_j^* - b_i\right) = 0, \qquad i = 1,\ldots,k. \tag{1.12}$$

Равенство (1.10) принято называть *соотношением двойственности*, условия (1.11), (1.12) – *условиями дополняющей нежёсткости*.

**Теорема 1.5.** (теорема двойственности). *Прямая задача* (1.1) *имеет решение в том и только том случае, если двойственная задача* (1.9) *имеет решение; при этом значения данных задач совпадают, то есть для любых их решений* $x^*$ *и* $p^*$ *выполнено соотношение двойственности* (1.10).

При практическом применении теории двойственности особенно полезным является следующее утверждение, непосредственно вытекающее из теорем 1.4 и 1.5.

**Теорема 1.6.** (о дополняющей нежёсткости). *Точки* $x^* \in X$ *и* $p^* \in P$ *являются решениями взаимодвойственных задач* (1.1) *и* (1.9) *соответственно в том и только том случае, если выполняются условия* (1.11), (1.12).

Таким образом, взаимодвойственные задачи линейного программирования имеют или не имеют решения одновременно. Следующая теорема показывает, что уже по их допустимым множествам можно отличить первый случай от второго.

**Теорема 1.7.** *Если допустимые множества взаимодвойственных задач* (1.13) *и* (1.14) *непусты, то обе они имеют решение. Если же только у одной из*

*них допустимое множество непусто, то её значение бесконечно, то есть*

$$\text{если } X \neq \varnothing,\ P = \varnothing,\ \text{то } \sup_X \langle c, x\rangle = \infty;$$

$$\text{если } X = \varnothing,\ P \neq \varnothing,\ \text{то } \inf_P \langle b, y\rangle = -\infty.$$

Наиболее эффективным является применение теории двойственности в тех случаях, когда двойственная задача решается проще, чем прямая. Если решение $p^*$ двойственной задачи (1.9) найдено, то, подставляя $p^*$ в (1.11), (1.12), можно получить систему на $X$ для определения решения $x^*$ прямой задачи (1.1). Приведем пример такого способа решения задачи линейного программирования.

**Пример 1.2.** Найти решение задачи

$$\begin{aligned}
&6x_1 + 11x_2 + 5x_3 + x_4 \to \min,\\
&-3x_1 + x_2 + 3x_3 - x_4 \geq 1,\\
&5x_1 + 3x_2 - 5x_3 - 3x_4 \geq 7,\\
&x_i \geq 0,\ \ i = 1,2,3,4.
\end{aligned}$$

Поскольку здесь $k = 2$, то для решения данной задачи целесообразно перейти к двойственной задаче, которую можно решить с помощью геометрических построений. Двойственной к исходной является следующая

$$\begin{aligned}
&p_1 + 7p_2 \to \max,\\
&-3p_1 + 5p_2 \leq 6,\\
&p_1 + 3p_2 \leq 11,\\
&3p_1 - 5p_2 \leq 5,\\
&-p_1 - 3p_2 \leq 1,\\
&p_i \geq 0,\ \ i = 1,2.
\end{aligned}$$

Из геометрических построений легко видеть, что ее решением является точка пересечения прямых $-3p_1 + 5p_2 = 6,\ p_1 + 3p_2 = 11,$ т.е. $p^* = (37/14,\ 39/14)$. Отметим, что $\langle b, p^*\rangle = 310/14.$

В точке $p^*$ третье и четвертое ограничения выполняются как строгие неравенства, поэтому решение $x^*$ прямой задачи должно удовлетворять условию $x_3^* = x_4^* = 0.$ Значения $x_1^*,\ x_2^*$ находим из системы $-3x_1 + x_2 = 1,\ 5x_1 + 3x_2 = 7,\ x_i \geq 0,\ i = 1,2$. Ее решение – $x_1^* = 4/14,$ $x_2^* = 26/14$. Тогда решением исходной задачи является точка $x^* = (4/14,\ 26/14, 0, 0),$ при этом $f^* = 310/14.$

## 1.5. Возможные интерпретации двойственной задачи и двойственных переменных

Как правило, если прямая задача линейного программирования имеет экономический смысл, то и двойственной задаче можно придать реальное содержание.

Рассмотрим производственную задачу:

$$\sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1,\ldots,m, \tag{1.13}$$

$$x_j \ge 0, \quad j = 1,\ldots,n.$$

Задача, двойственная задаче (1.13), имеет вид

$$\sum_{i=1}^{m} p_i b_i \to \min, \tag{1.14}$$

$$\sum_{i=1}^{m} p_i a_{ij} \ge c_j, \quad j = 1,\ldots,n, \tag{1.15}$$

$$p_i \ge 0, \quad i = 1,\ldots,m.$$

Для выяснения смысла двойственной задачи и двойственных переменных прежде всего проведем анализ размерностей.

Поскольку размерность величины $c_j$ – «руб./изделие», размерность элемента $a_{ij}$ – «ресурс/изделие», то $j$-е неравенство в (1.15) будет содержательно, если размерность величины $p_i$ – «руб./ресурс», т.е. $p_i$ – стоимость единицы $i$-го ресурса. Тогда $\sum_{i=1}^{m} p_i b_i$ имеет размерность «руб.» и выражает стоимость всех ресурсов. Левая часть $i$-го ограничения в (1.15) имеет размерность «руб.» и выражает стоимость всех ресурсов, израсходованных на изготовление единицы $j$-го изделия. Таким образом, двойственной задаче (1.14), (1.15) можно придать следующее содержание: найти такие цены ресурсам, чтобы стоимость ресурсов на изготовление единицы каждого изделия не была меньше стоимости изделия, а стоимость всех ресурсов, израсходованных на производство изделий, была минимальной.

При такой интерпретации смысл истинной цены имеют лишь величины $c_j$, величины же $p_i$ интерпретированы как цены, поэтому их называют также объективно-обусловленными оценками. Такое название соответствует следующей интерпретации сопряженных переменных. В силу теоремы двойственности на решениях прямой и двойственной задач выполняется ра-

венство

$$\langle c, x^* \rangle = \langle b, p^* \rangle .$$

Если допустить, что при малых вариациях $\Delta b$ вектора $b$ решение $p^*$ двойственной задачи не меняется, то из (1.16) получаем

$$\langle c,\ \tilde{x} - x^* \rangle = \langle \Delta b,\ p^* \rangle,$$

т.е. вектор $p^*$ имеет смысл градиента критерия качества в пространстве ресурсов и указывает скорость изменения (чувствительность) критерия в зависимости от изменения вектора ресурсов. В частности, если $i$-й ресурс не полностью используется в производстве изделий, то согласно принципу дополняющей нежесткости $p_i^* = 0$, и поэтому максимальная стоимость изготовленных изделий не будет зависеть от малых колебаний запаса $b_i$ $i$-го ресурса. Приведенной интерпретации можно придать точный смысл, если вычислить производную критерия качества по направлениям в пространстве ресурсов.

## 1.6. Симплекс-метод решения задач линейного программирования

Основным численным методом решения задач линейного программирования является так называемый симплекс-метод.

Термин «симплекс-метод» связан с тем историческим обстоятельством, что первоначально метод был разработан применительно к задаче линейного программирования, допустимое множество которой имело вид $X = \{x \in \mathbf{R}^n \mid \sum_{i=1}^{n} x_i = 1\}$ (это множество именуется стандартным симплексом).

Симплекс-метод предназначен для решения задачи линейного программирования в канонической форме

$$\langle c, x \rangle \to \min, \quad Ax = b, \quad x \geq 0, \tag{1.16}$$

где $A$ – матрица размера $m \times n$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^n$. Это не ограничивает общность метода, так как любая задача линейного программирования может быть представлена в такой форме.

Всюду под $X$ будет пониматься допустимое множество задачи (1.16):

$$X = \{x \in \mathbf{R}^n \mid Ax = b,\ x \geq 0\}.$$

Обозначим через $a^1$, ... , $a^n$ столбцы матрицы $A$. Для любой точки $x = (x_1, \ldots, x_n) \in X$ введем множество

$$J(x) = \{j \mid x_j > 0,\ 1 \leq j \leq n\},$$

состоящее из индексов ее положительных координат. Отметим, что из условия $x \in X$ следует

$$b = Ax = \sum_{j \in J(x)}^{n} a^j x_j ,$$

т.е. вектор $b$ является положительной линейной комбинацией столбцов $a^j$ $(j \in J(x))$.

**Определение.** Точка $x$ допустимого множества $X$ называется *опорной,* если столбцы $a^j$ $(j \in J(x))$ линейно независимы.

**Теорема 1.8.** *Понятие опорной точки множества X эквивалентно понятию его крайней точки.*

**Теорема 1.9.** *Если множество X непусто, то оно имеет опорные точки и число их конечно. При этом для любой точки* $x \in X$ *существует такая опорная точка* $x' \in X$, *что* $J(x') \subset J(x)$.

Таким образом, из теоремы 1.1 следует следующее утверждение

**Теорема 1.10.** *Если множество решений задачи* (1.16) *непусто, то оно содержит хотя бы одну опорную точку допустимого множества X.*

Как уже отмечалось, для поиска решения задачи (1.16) достаточно перебрать лишь опорные точки допустимого множества $X$, число которых конечно. Та опорная точка, в которой целевая функция $\langle c, x \rangle$ принимает наименьшее значение, будет одним из решений задачи (1.16) (если они вообще существуют). Чтобы найти все опорные точки, необходимо для каждого множества индексов $J \subset \{1,...,n\}$ такого, что столбцы $a^j$ $(j \in J)$ линейно независимы, составить относительно $x_j$ $(j \in J)$ систему

$$\sum_{j \in J}^{n} a^j x_j = b. \qquad (1.17)$$

Если эта система имеет решение и оно неотрицательно, то, положив $x_j = 0$ при $j \notin J$, получим, что точка $x = (x_1,...,x_n)$ − опорная. В противном случае осуществляется переход к следующему $J$. Ясно, что в результате этой процедуры ни одна опорная точка не будет пропущена.

Как известно, для того чтобы решить линейную систему (1.17) или убедиться, что она не имеет решения, а также чтобы исследовать векторы на линейную независимость, требуется конечное число арифметических операций. Таким образом, принципиально возможно решить задачу линейного программирования за конечное число шагов методом полного перебора опорных точек. Однако, при больших $n$ и $m$ этот простой в идейном отношении метод требует огромной вычислительной работы.

Итак, метод полного перебора опорных точек практической ценности не имеет. Но он естественным образом подводит к основной идее *симплекс-*

*метода:* полный перебор следует заменить упорядоченным. Данный метод представляет собой вычислительную процедуру, в которой специальным образом генерируется последовательность $x^0$, $x^1$, ... , $x^t$ опорных точек множества *X*. На очередной *t*-й итерации в зависимости от знаков некоторых параметров делается один из следующих выводов:

1) $x^t$ – решение задачи (1.16);

2) задача (1.16) не имеет решений;

3) существует (конструктивно указываемая) «лучшая» опорная точка $x^{t+1}$, в которой целевая функция принимает меньшее значение, чем

$$\langle c, x^{t+1} \rangle < \langle c, x^t \rangle . \qquad (1.18)$$

Поскольку, во-первых, число опорных точек конечно и среди них обязательно имеется решение задачи (1.16), если она вообще разрешима, а во-вторых, в силу (1.18) возврат к однажды просмотренным опорным точкам уже невозможен, то за конечное число итераций эта процедура приведет к выводу 1) или 2).

Теоретически не исключается (такие примеры построены), что последовательность $x^0$, $x^1$, ... , $x^t$ пройдет по всем опорным точкам множества *X*. Однако для подавляющего числа задач линейного программирования вида (1.16) количество итераций симплекс-метода находится в пределах от *m* до 2*m*.

В дальнейшем будем предполагать, что ранг матрицы *A* равен *m*, т. е. все ее строки линейно независимы. К этому всегда можно прийти, исключив из системы $Ax = b$ линейно зависимые уравнения. На практике данная операция обычно осуществляется одновременно с поиском начальной опорной точки $x^0$ по методу искусственного базиса.

Отметим, что у любой опорной точки множества *X* может быть не более *m* положительных координат, так как в пространстве $\mathbf{R}^m$ может быть не более *m* линейно независимых векторов.

**Определение.** *Базисом* опорной точки *x* множества *X* называется произвольная линейно независимая система из *m* столбцов матрицы *A*, включающая в себя все столбцы, соответствующие положительным координатам точки *x*.

**Определение.** Опорная точка *x* называется *невырожденной,* если она имеет ровно *m* положительных координат. Задача (1.16) называется *невырожденной,* если любая ее опорная точка невырождена. В противных случаях говорят о *вырожденной* точке или задаче.

Для вырожденной задачи возможен случай, когда на очередной итерации не произойдет перехода к новой опорной точке, а значит, и уменьшения значения целевой функции, т.е. будет выполняться $x^{t+1} = x^t$ и, значит, $\langle c, x^{t+1} \rangle = \langle c, x^t \rangle$ вместо (1.18).

***Итерация симплекс-метода***

Пусть в результате предыдущих итераций уже получена последовательность $x^0$, $x^1$, ... , $x^t$ опорных точек множества $X$. Для краткости положим $x^t = x$.

Пусть $a^j$ $(j \in J)$ − некоторый базис точки $x$. Поскольку это базис в $\mathbf{R}^m$, то все столбцы матрицы $A$ можно представить как линейные комбинации данных столбцов, т.е.

$$a^k = \sum_{j \in J} a^j \lambda_{jk},\ k = 1,..,n,$$

где $\lambda_{jk}$ $(j \in J)$ – некоторые числа. Положим

$$\Delta_k = \sum_{j \in J} c_j \lambda_{jk} - c_k,\ k = 1,..,n.$$

Для любого $k \in J$ очевидно, имеем: $\lambda_{kk} = 1$, $\lambda_{jk} = 0$, при $j \in J \setminus \{k\}$ и $\Delta_k = 0$.

В зависимости от знаков параметров $\lambda_{jk}$, $\Delta_k$ при $j \in J$, $k \notin J$ выполняется хотя бы одно из трех условий:

I. Для любого индекса $k \notin J$ справедливо неравенство $\Delta_k \le 0$.

II. Существует индекс $s \notin J$ такой, что $\Delta_s > 0$ и $\lambda_{js} \le 0$ при всех $j \in J$.

III. Существует индекс $s \notin J$ такой, что $\Delta_s > 0$ и $\lambda_{js} > 0$ при некотором $j \in J$.

Оказывается, каждому из этих условий соответствует один из упомянутых ранее выводов.

**Теорема 1.11 (правило оптимальности).** *Если выполняется условие* I, *то x – решение задачи* (1.16).

**Теорема 1.12 (правило отсутствия решения).** *Если выполняется условие* II, *то задача* (1.16) *не имеет решения*.

Итак, при выполнении условий I или II работа метода заканчивается на данной итерации.

Далее, положим

$$\alpha = \min_{j \in J:\, \lambda_{js} > 0} \frac{x_j}{\lambda_{js}}, \qquad (1.19)$$

где $s$ взято из условия III. Пусть этот минимум достигается на индексе $r$, т.е.

$$\alpha = \frac{x_r}{\lambda_{rs}},\ r \in J,\ \lambda_{rs} > 0. \qquad (1.20)$$

Пусть

$$x'_j = \begin{cases} x_j - \alpha\lambda_{js}, & \text{если } j \in J, \\ \alpha, & \text{если } j = s, \\ 0, & \text{если } j \notin J, j \neq s. \end{cases} \qquad (1.21)$$

**Теорема 1.13 (правило перехода к новой вершине).** *Если выполняется условие* III, *то точка* $x'$, *определенная формулами* (1.21), (1.20), *является опорной, причем столбцы* $a^j$, $j \in J'$, *где* $J' = (J\backslash\{r\})\backslash\{s\}$, *образуют ее базис*.

На следующей ($t$+1)-й итерации в качестве $x^{t+1}$ принимается указанная опорная точка $x'$ с данным базисом $a^j$, $j \in J'$. При этом говорят, что столбец $a^r$ *выводится* из базиса, а столбец $a^s$ *вводится* в базис. Элемент $\lambda_{rs}$ называется *ведущим*.

***Замечание* 1.1.** Вообще говоря, индекс $s$ определяется из условия III неоднозначно. Для невырожденной задачи индекс $r$, удовлетворяющий (1.20), единствен; обратное означало бы, что опорная точка $x'$ содержит менее $m$ положительных координат. Но для вырожденной задачи таких $r$ может быть много. На практике обычно выбирают наименьшие $s$ и $r$, удовлетворяющие указанным требованиям.

***Замечание* 1.2.** В теореме 1.13 возможны два случая: $\alpha = 0$ или $\alpha > 0$. При $\alpha = 0$ имеем $x' = x$, т.е. происходит лишь замена одного базиса точки $x$ другим. При $\alpha > 0$ заведомо $x' \neq x$ и $\langle c, x'\rangle < \langle c, x\rangle$. Если точка $x$ невырождена, и, значит, $J = J(x)$, то $x_r > 0$ и поэтому обязательно $\alpha > 0$. Но для вырожденной точки $x$ случай $\alpha = 0$ не исключается. Таким образом, для вырожденной задачи возможна ситуация, когда итерации описанной процедуры сведутся к перебору базисов одной и той же опорной точки, не являющейся решением; причем с некоторого момента эти базисы начнут повторяться, так как число их конечно. В таком случае говорят, что произошло *зацикливание* симплекс-метода. Известны специально построенные примеры задач линейного программирования, в которых это явление действительно наблюдается. Однако при решении реальных задач линейного программирования, многие из которых вырождены, зацикливание встречается крайне редко: если даже в процессе вычислений некоторая опорная точка повторяется, то, как правило, рано или поздно обнаруживается такой ее базис, который позволяет перейти к новой опорной точке.

Несмотря на это, разработаны различные уточнения основной процедуры симплекс-метода, позволяющие полностью исключить возможность зацикливания. Одним из уточнений является так называемое *лексикографическое правило* выбора индекса $r$ в формуле (1.20). Это правило состоит в следующем.

Пусть $J_0$ – множество всех индексов, на которых достигается минимум в (1.19). Если $J_0$ состоит из единственного $r$, то $r$ – искомый индекс. В

противном случае рассмотрим множество $J_1$ всех индексов из $J_0$, на которых достигается минимум отношения $\lambda_{j1}/\lambda_{js}$ при $j \in J_0$. Если $J_1$ состоит из единственного *r*, то *r* – искомый индекс. В противном случае рассматривается множество $J_2$ всех индексов из $J_1$, на которых достигается минимум отношения $\lambda_{j2}/\lambda_{js}$ при $j \in J_1$, и т.д. Если этот процесс еще не закончился ранее, то на *n*-м шаге рассматривается множество $J_n$ всех индексов из $J_{n-1}$, на которых достигается минимум отношения $\lambda_{jn}/\lambda_{js}$ при $j \in J_{n-1}$. Индекс $r \in J_n$ и берется в (1.20).

При определенных условиях на базис начальной опорной точки $x^0$ применение описанного правила выбора *r* на каждой итерации симплекс-метода позволяет избежать зацикливания.

Итак, на следующей итерации описанная процедура повторяется для построенной опорной точки $x'$ с базисом $a^j$, $j \in J'$. В первую очередь столбцы матрицы *A* выражаются через данные:

$$a^k = \sum_{j \in J'} a^j \lambda'_{jk},\ k = 1,\ldots,n,$$

а затем вычисляются параметры

$$\Delta'_k = \sum_{j \in J} c_j \lambda'_{jk} - c_k,\ k = 1,\ldots,n. \tag{1.22}$$

Вообще говоря, определение коэффициентов $\lambda'_{jk}$ требует решения соответствующих систем линейных уравнений, что является довольно трудоемкой операцией. Однако оказывается, что, зная величины $\lambda_{jk}$, $\Delta_k$, параметры $\lambda'_{jk}$, $\Delta'_k$ можно вычислить значительно проще.

**Теорема 1.14.** *При любом* $k = 1,\ldots,n$ *выполняются соотношения*

$$\lambda'_{jk} = \begin{cases} \lambda_{jk} - \dfrac{\lambda_{js}}{\lambda_{rs}} \lambda_{rk}, & \text{если } j \in J \setminus \{r\}, \\ \dfrac{\lambda_{rk}}{\lambda_{rs}}, & \text{если } j = s, \end{cases} \tag{1.23}$$

$$\Delta'_k = \Delta_k - \frac{\Delta_k}{\lambda_{rs}} \lambda_{rk}. \tag{1.24}$$

***Организация ручного счета по симплекс-методу***

Основной конструкцией является здесь так называемая *симплекс-таблица T,* связанная с текущей опорной точкой *x* и данным ее базисом $a^j$ $(j \in J)$.

Эта таблица представляет собой матрицу размера $(m + 1) \times (n + 1)$. Над

столбцами таблицы выписываются буквенные обозначения $a^1$, ... , $a^m$, $b$ столбцов матрицы $A$ и вектора правых частей ограничений задачи (1.16), а с левой стороны таблицы – обозначения $a^j\,(j\in J)$, столбцов матрицы $A$, образующих базис, и обозначение $\Delta$. На указанные места таблицы заносятся численные значения параметров $\lambda_{jk}$, $x_j$, $\Delta_k$ при $j\in J$, $k=1,..,n$. В правом нижнем углу таблицы проставляется значение целевой функции задачи в данной точке $x$, т.е. $\langle c,x\rangle$.

| | $a^1$ | ... | $a^k$ | ... | $a^s$ | ... | $a^n$ | $b$ |
|---|---|---|---|---|---|---|---|---|
| ⋮ | ⋮ | | ⋮ | | ⋮ | | ⋮ | ⋮ |
| $a^j$ | $\lambda_{j1}$ | ... | $\lambda_{jk}$ | ... | $\lambda_{js}$ | ... | $\lambda_{j1}$ | $x_j$ |
| ⋮ | ⋮ | | ⋮ | | ⋮ | | ⋮ | ⋮ |
| $a^r$ | $\lambda_{r1}$ | ... | $\lambda_{rk}$ | ... | $\boxed{\lambda_{rs}}$ | ... | $\lambda_{rn}$ | $x_r$ |
| ⋮ | ⋮ | | ⋮ | | ⋮ | | ⋮ | ⋮ |
| $\Delta$ | $\Delta_1$ | ... | $\Delta_k$ | ... | $\Delta_s$ | ... | $\Delta_n$ | $\langle c,\ x\rangle$ |

Затем проводится анализ симплекс-таблицы с тем, чтобы выяснить, какое из условий I-III выполняется. Если выполняется условие I или условие II расчеты заканчиваются.

Пусть выполняется условие III, т. е. $\Delta_s>0$, $\lambda_{js}>0$ при некоторых $s\notin J$ и $j\in J$. Если таких $s$ несколько, то, в соответствии с замечанием 1.1, выбирается, например, минимальный из них. Столбец $a^s$ должен быть *введен* в базис. Далее, по формуле (1.19) вычисляется $\alpha$ и выбирается, например, минимальный индекс $r$, удовлетворяющий (1.20) (если возникло зацикливание, то следует использовать более трудоемкое лексикографическое правило выбора $r$). Столбец $a^r$ должен быть *выведен* из базиса. *Ведущий* элемент $\lambda_{rs}>0$ в таблице $T$ выделяется. Затем по формулам (1.21), (1.22), (1.23), (1.24) осуществляется переход к новой симплекс-таблице $T'$, соответствующей опорной точке $x'$ с базисом $a^j$, $j\in J'=(J\setminus\{r\})\cup\{s\}$. При этом в таблице $T'$ слева вместо «$a^r$» ставится «$a^s$», а остальные буквенные обозначения остаются неизменными. Легко видеть, что переход к таблице $T'$ сводится к следующим элементарным операциям со строками таблицы $T$:

1) для получения строки $(\lambda'_{j1},...,\lambda'_{jn},x'_j)$ таблицы $T'$ при $j\in J\setminus\{r\}$ из строки $(\lambda_{j1},...,\lambda_{jn},x_j)$ таблица $T$ вычитается ее строка $(\lambda_{r1},...,\lambda_{rn},x_r)$, умноженная на коэффициент $\lambda_{js}/\lambda_{rs}$;

2) для получения строки $(\lambda'_{s1},...,\lambda'_{sn},x'_s)$ таблицы $T'$ строка $(\lambda_{r1},...,\lambda_{rn},x_r)$ таблицы Т делится на коэффициент $\lambda_{rs}$;

3) для получения строки $(\Delta'_1,\ldots,\Delta'_n,\langle c,x'\rangle)$ таблицы $T'$ (в полной аналогии с 1)) из строки $(\Delta_1,\ldots,\Delta_n,\langle c,x\rangle)$ таблицы $T$ вычитается ее строка $(\lambda_{r1},\ldots,\lambda_{rn},x_r)$, умноженная на коэффициент $\Delta_s / \lambda_{rs}$.

Назначение указанных коэффициентов состоит в том, чтобы в столбце под обозначением «$a^s$» получить все нули, кроме единицы на $s$-м месте, т.е. $\lambda_{js}=0$ при $j\in J\setminus\{r\}$, $\lambda_{ss}=1$ и $\Delta'_s=0$.

Затем проводится анализ таблицы $T'$. Если выполняется условие III, то осуществляется переход к следующей симплекс-таблице, и т.д.

**Пример 1.3.** Решить задачу

$$\begin{aligned} &5x_1 + x_2 + 2x_3 + x_4 \to \max,\\ &x_1 - x_2 + x_3 = 1,\\ &2x_1 + x_2 + x_4 = 5,\\ &x_j \ge 0,\quad j=1,\ldots,4. \end{aligned}$$

Для решения данного примера будем использовать симплекс-метод. С этой целью приведем задачу к следующему виду

$$\begin{aligned} &-5x_1 - x_2 - 2x_3 - x_4 \to \min,\\ &x_1 - x_2 + x_3 = 1,\\ &2x_1 + x_2 + x_4 = 5,\\ &x_j \ge 0,\quad j=1,\ldots,4. \end{aligned}$$

Легко видеть, что точка $x^0$= (0, 0, 1, 5) является опорной, при этом столбцы $a^3, a^4$ образуют естественный базис в $\mathbf{R}^2$, и значит, коэффициенты разложения по нему столбцов $a^1, a^2$ совпадают с их координатами. С учетом всего этого строим первую симплекс-таблицу:

| | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $b$ |
|---|---|---|---|---|---|
| $a^3$ | 1 | -1 | 1 | 0 | 1 |
| $a^4$ | 2 | 1 | 0 | 1 | 5 |
| $\Delta$ | 1 | 2 | 0 | 0 | -7 |

Для построенной симплекс-таблицы выполняется условие III, поэтому строим следующую симплекс-таблицу, при этом $a^3$ выводим из базиса, а $a^1$

вводим в базис:

| | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $b$ |
|---|---|---|---|---|---|
| $a^1$ | 1 | -1 | 1 | 0 | 1 |
| $a^4$ | 0 | 3 | -2 | 1 | 3 |
| $\Delta$ | 0 | 3 | -1 | 0 | -8 |

Т.к. выполняется условие III, то строим следующую симплекс-таблицу, при этом $a^4$ выводим из базиса, а $a^2$ вводим в базис:

| | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $b$ |
|---|---|---|---|---|---|
| $a^1$ | 1 | 0 | 1/3 | 1/3 | 2 |
| $a^2$ | 0 | 1 | -2/3 | 1/3 | 1 |
| $\Delta$ | 0 | 0 | 1 | -1 | -11 |

Для построенной симплекс-таблицы выполняется условие III, поэтому строим следующую симплекс-таблицу, при этом $a^1$ выводим из базиса, а $a^3$ вводим в базис:

| | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $b$ |
|---|---|---|---|---|---|
| $a^3$ | 3 | 0 | 1 | 1 | 6 |
| $a^2$ | 2 | 1 | 0 | 1 | 5 |
| $\Delta$ | -3 | 0 | 0 | -21 | -17 |

На данном шаге получена таблица, удовлетворяющая условию I, таким образом полученная точка $x^* = (0, 5, 6, 0)$ является решением задачей, при этом максимальное значение целевой функции в исходной задаче равно 17.

### *Поиск начальной опорной точки*

Описанная выше процедура симплекс-метода предполагает, что начальная опорная точка $x^0$ уже дана. В некоторых частных случаях отыскание $x^0$ не составляет труда (см. пример 1.3). Но, вообще говоря, поиск $x^0$, т.е. какой-нибудь опорной точки множества *X*, по своей трудоемкости сопоставим с самой процедурой симплекс-метода. Рассмотрим один из общих методов построения начальной опорной точки – *метод искусственного базиса.*

Без ограничения общности можно считать, что в задаче (1.16) выполнено условие $b \geq 0$ (этого всегда можно добиться, умножая ограничения-равенства задачи на –1). Рассмотрим вспомогательную задачу линейного программирования

$$\sum_{i=1}^{m} u_i \to \min, \quad Ax + u = b, \quad x \geq 0, \quad u \geq 0. \tag{1.25}$$

Допустимое множество этой задачи можно записать в виде

$$\hat{X} = \{(x,u) \in \mathbf{R}^n \times \mathbf{R}^m \mid \sum_{j=1}^{n} a^j x_j + \sum_{i=1}^{m} e^i u_i = b,\ x \geq 0,\ u \geq 0\},$$

где $e^1$, ..., $e^m$ – единичные орты в $\mathbf{R}^m$. Отсюда ясно, что (0, $b$) – опорная точка множества $\hat{X}$. Т.к. целевая функция задачи (1.25) ограничена снизу (нулем) на $\hat{X}$, то эта задача разрешима. Применяя к ней процедуру симплекс-метода с указанной начальной опорной точкой, найдем опорную точку $(x^*, u^*)$ множества $\hat{X}$, являющуюся решением. Пусть $f^* = \sum_{i=1}^{m} u_i^*$ значение задачи (1.25).

**Теорема.** *Если* $f^* = 0$, *то* $x^*$ *– опорная точка множества X. Если* $f^* > 0$, *то задача* (1.16) *не имеет допустимых точек*: $X = \varnothing$.

Таким образом, при решении задачи (1.16) основная процедура симплекс-метода применяется дважды: сначала с ее помощью решается задача (1.25), а затем – сама задача (1.16).

## 1.7. Задачи

1. Преобразовать к основной, стандартной и канонической формам

1.1.

$$\begin{array}{rcrcrcl} 4x_1 & + & x_2 & + & 3x_3 & \to & \max, \\ 2x_1 & + & x_2 & + & x_3 & \geq & 1, \\ & & 3x_2 & + & x_3 & = & 2, \\ -x_1 & + & 5x_2 & + & 2x_3 & \leq & 4, \\ & & x_2 \geq 0, & & x_3 \geq 0. & & \end{array}$$

1.2.

$$
\begin{aligned}
&2x_1 - x_2 - 5x_3 \to \max,\\
&4x_1 + 3x_2 + x_3 = 4,\\
&-x_1 + 6x_2 - x_3 \geq 2,\\
&7x_1 - x_2 + 2x_3 \leq 5,\\
&x_1 \geq 0, \quad x_2 \geq 0.
\end{aligned}
$$

1.3.

$$
\begin{aligned}
&x_1 + x_2 - x_3 - x_4 \to \min,\\
&x_1 + 3x_2 - x_4 \geq 2,\\
&-x_1 + x_2 + x_4 \geq 1,\\
&x_2 + x_3 \leq 3,\\
&x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.
\end{aligned}
$$

2. Используя геометрические построения найти решения следующих задач:

2.1.

$$
\begin{aligned}
&x_1 + 2x_2 \to \max\\
&3x_1 - 2x_2 \leq 6,\\
&-x_1 + 2x_2 \leq 4,\\
&3x_1 + 2x_2 \leq 12,\\
&x_1 \geq 0.
\end{aligned}
$$

2.2.

$$
\begin{aligned}
&2x_1 + x_2 \to \max\\
&-x_1 + x_2 \leq 2,\\
&x_1 + 2x_2 \leq 7,\\
&4x_1 - 3x_2 \leq 6,\\
&x_1 \geq 0, \quad x_2 \geq 0.
\end{aligned}
$$

2.3.

$$
\begin{aligned}
&7x_1 + 5x_2 \to \max,\\
&x_1 + x_2 \geq 3,\\
&x_1 + 5x_2 \geq 5,\\
&2x_1 + x_2 \geq 4.
\end{aligned}
$$

2.4.

$$
\begin{aligned}
&-x_1 + 2x_2 \to \max,\\
&2x_1 - 3x_2 \geq 0,\\
&x_1 - x_2 \leq 3,\\
&2x_1 - x_2 = 4.
\end{aligned}
$$

2.5.

$$
\begin{aligned}
&8x_1 - 2x_2 - 3x_3 \to \max,\\
&-x_1 + 3x_2 + x_3 \leq 4,\\
&7x_1 - 2x_3 \leq 16,\\
&2x_1 - x_2 - x_3 = 2,\\
&x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.
\end{aligned}
$$

3. Определить, имеются ли среди указанных точек решения задач линейного

программирования

3.1.

$$
\begin{array}{rcrcrcrcl}
-2x_1 & + & 3x_2 & + & x_3 & & & \to & \max, \\
3x_1 & + & x_2 & + & 2x_3 & - & x_4 & \le & 3, \\
x_1 & + & x_2 & + & x_3 & - & 3x_4 & \le & -1, \\
-5x_1 & + & 2x_2 & - & x_3 & + & x_4 & \le & -3,
\end{array}
$$

$x^1=(1, 3, 0, 3)$,

$x^2=(0, -1, 3, 2)$,

$x^3=(5, 0, -6, 0)$.

3.2.

$$
\begin{array}{rcrcrcrcl}
x_1 & + & 5x_2 & - & 3x_3 & - & 5x_4 & \to & \min, \\
-2x_1 & - & x_2 & + & 4x_3 & + & x_4 & \le & 1, \\
x_1 & - & x_2 & + & x_3 & + & 2x_4 & \le & -3, \\
3x_1 & + & 2x_2 & - & x_3 & - & x_4 & = & 5,
\end{array}
$$

$$x_2 \ge 0, \quad x_3 \ge 0,$$

$x^1=(0, 5/2, 1/2, -1/2)$,

$x^2=(1, 0, 0, -2)$,

$x^3=(0, 0, 0, 1)$.

3.3.

$$
\begin{array}{rcrcrcrcl}
-x_1 & & & + & 3x_3 & + & 5x_4 & \to & \max, \\
5x_1 & + & 3x_2 & - & x_3 & + & x_4 & \le & 9, \\
-x_1 & + & 4x_2 & + & 2x_3 & + & x_4 & \le & 9, \\
-2x_1 & - & x_2 & + & 3x_3 & + & 2x_4 & \le & -1,
\end{array}
$$

$$x_j \ge 0, \quad j = 1, \dots, 4,$$

$x^1=(1, 1, 0, 1)$,

$x^2=(2, 0, 1, 0)$,

$x^3=(5/12, 2, 0, 11/12)$.

4. Найти решения задач методом полного перебора вершин

4.1.

$$\begin{aligned} x_1 + x_2 + x_3 &\to \max, \\ x_1 - x_2 + x_3 &\le 4, \\ 2x_1 + x_2 + x_3 &\le 3, \\ 3x_1 + x_2 + 2x_3 &\le 6, \\ -x_1 + 2x_2 - x_3 &\le -3. \end{aligned}$$

4.2.

$$\begin{aligned} x_1 + x_2 + 2x_3 + 3x_4 &\to \min, \\ x_1 + x_2 + 3x_3 + 4x_4 &= 12, \\ x_1 - x_2 + x_3 - x_4 &= 2, \\ x_j \ge 0, \quad j &= 1, \dots, 4, \end{aligned}$$

4.3.

$$\begin{aligned} x_1 + 2x_2 - 3x_3 &\to \min, \\ -x_1 + 2x_2 + 2x_3 &\le 1 \\ x_1 + x_2 - x_3 &= 0 \\ x_j \ge 0, \quad j &= 1, \dots, 3. \end{aligned}$$

5. Построить задачи двойственные к следующим задачам линейного программирования:

5.1.

$$\begin{aligned} 17x_1 - 5x_2 + x_3 + x_4 - 8x_5 &\to \max, \\ 3x_1 - 3x_2 - x_3 + 4x_4 + 7x_5 &\le 11, \\ x_1 - 5x_2 - 5x_3 + x_4 + 2x_5 &\ge -8, \\ x_1 + x_2 + x_3 + 3x_4 - x_5 &= 4, \\ x_1 \ge 0, \qquad & x_4 \ge 0. \end{aligned}$$

5.2.

$$\begin{aligned} 4x_1 - 6x_2 - 2x_3 + 3x_4 + x_5 &\to \min, \\ x_1 + 2x_2 - 3x_3 + x_4 - 3x_5 &\ge -5, \\ 2x_1 + 3x_2 + x_3 + x_4 + 2x_5 &\ge 1, \\ -2x_1 - x_2 - x_4 - x_5 &\le 3. \end{aligned}$$

5.3.

$$
\begin{array}{rcrcrcl}
 & & 3x_2 & - & 2x_3 & + & x_4 \quad \to \quad \min, \\
-x_1 & & & + & x_3 & - & x_4 \quad = \quad 5, \\
2x_1 & + & x_2 & - & 2x_3 & + & 2x_4 \quad \le \quad 7, \\
x_j \ge 0, & & j = 1, \ldots, 4. & & & &
\end{array}
$$

5.4.

$$
\begin{array}{rcrcrcrcrcl}
4x_1 & + & x_2 & + & x_3 & + & 2x_4 & + & x_5 & \to & \max, \\
4x_1 & + & x_2 & - & x_3 & - & x_4 & + & x_5 & \ge & 9, \\
x_1 & + & x_2 & - & x_3 & + & x_4 & + & 6x_5 & = & 10, \\
-x_1 & - & 3x_2 & + & 5x_3 & & & & & \le & 1, \\
x_j \ge 0, & & j = 1, \ldots, 4. & & & & & & & &
\end{array}
$$

6. Используя теорию двойственности и геометрические построения, найти решения следующих задач:

6.1.

$$
\begin{array}{rcrcrcl}
7x_1 & & & + & x_3 & - & 4x_4 \quad \to \quad \max, \\
x_1 & - & x_2 & + & 2x_3 & - & x_4 \quad \le \quad 6, \\
2x_1 & + & x_2 & - & x_3 & & \quad \le \quad -1, \\
x_j \ge 0, & & j = 1, \ldots, 4. & & & &
\end{array}
$$

6.2.

$$
\begin{array}{rcrcrcrcl}
x_1 & & & + & x_3 & & + & x_5 \quad \to \quad \max, \\
x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & - & x_5 \quad \le \quad 6, \\
x_1 & - & x_2 & - & 2x_3 & + & x_4 & + & x_5 \quad \le \quad 5, \\
x_j \ge 0, & & j = 1, \ldots, 5. & & & & & &
\end{array}
$$

6.3.

$$
\begin{array}{rcrcrcrcrcl}
4x_1 & + & 5x_2 & + & 2x_3 & + & x_4 & + & 2x_5 & \to & \min, \\
-3x_1 & + & 5x_2 & + & 4x_3 & + & 2x_4 & + & 2x_5 & = & 1, \\
-4x_1 & - & 6x_2 & - & x_3 & + & x_4 & + & 3x_5 & = & -1, \\
x_j \ge 0, & & j = 1, \ldots, 5. & & & & & & & &
\end{array}
$$

6.4.

$$
\begin{array}{rcrcrcrcl}
6x_1 & + & 3x_2 & - & x_3 & - & 2x_4 & \to & \max, \\
3x_1 & + & 2x_2 & + & x_3 & + & 4x_4 & \le & 0, \\
2x_1 & + & 2x_2 & - & x_3 & - & x_4 & = & 1,
\end{array}
$$

$x_j \ge 0, \quad j = 2, 3, 4.$

6.5.

$$
\begin{array}{rcrcrcrcl}
2x_1 & + & 4x_2 & - & x_3 & - & x_4 & \to & \max, \\
-2x_1 & + & 2x_2 & - & x_3 & - & x_4 & \le & -2, \\
x_1 & - & x_2 & + & x_3 & & & \le & 1, \\
3x_1 & + & x_2 & & & + & x_4 & = & 5,
\end{array}
$$

$x_j \ge 0, \quad j = 1, 2, 4.$

6.6.

$$
\begin{array}{rcrcrcrcrcl}
-x_1 & + & 3x_2 & + & 31x_3 & + & 25x_4 & - & 4x_5 & \to & \min, \\
x_1 & & & + & 4x_3 & + & 3x_4 & + & x_5 & = & 2, \\
& & x_2 & + & x_3 & + & x_4 & - & x_5 & = & 1, \\
x_1 & + & x_2 & + & 10x_3 & + & 8x_4 & + & x_5 & \le & 1, \\
2x_1 & - & x_2 & + & 9x_3 & + & 6x_4 & + & 5x_5 & \le & 0,
\end{array}
$$

$x_j \ge 0, \quad j = 1, 2, 5.$

6.7.

$$
\begin{array}{rcrcrcrcrcl}
4x_1 & - & 8x_2 & - & 7x_3 & - & 2x_4 & & & \to & \max, \\
& - & 3x_2 & - & 4x_3 & - & x_4 & + & 2x_5 & \le & 1, \\
4x_1 & + & x_2 & + & 2x_3 & + & 5x_4 & - & 3x_5 & = & -4, \\
x_1 & + & x_2 & + & x_3 & + & 2x_4 & - & 5x_5 & = & -1,
\end{array}
$$

$x_j \ge 0, \quad j = 1, 2, 3, 4.$

7. Решить симплекс-методом следующие задачи, начав с указанных вершин:

7.1.

$$
\begin{array}{rcrcrcrcl}
x_1 & + & 3x_2 & + & x_3 & - & x_4 & \to & \max, \\
x_1 & + & 2x_2 & & & + & x_4 & = & 3, \\
-x_1 & + & x_2 & + & x_3 & & & = & 1,
\end{array}
$$

$x_j \ge 0, \quad j = 1, \ldots, 4;$

$x^0 = (0, 0, 1, 3).$

7.2.

$$
\begin{array}{rcrcrcrcrcl}
x_1 & + & 3x_2 & + & 2x_3 & + & 4x_4 & - & 2x_5 & \to & \min, \\
-x_1 & & & + & x_3 & - & 2x_4 & & & = & -2, \\
 & & x_2 & - & x_3 & + & x_4 & - & 2x_5 & = & 0, \\
2x_1 & + & x_2 & & & + & 5x_4 & + & x_5 & = & 7,
\end{array}
$$

$x_j \geq 0, \quad j = 1, \ldots, 5;$

$x^0 = (3, 1, 1, 0, 0).$

7.3.

$$
\begin{array}{rcrcrcrcrcl}
3x_1 & - & 2x_2 & + & x_3 & + & 3x_4 & + & 3x_5 & \to & \max, \\
2x_1 & - & x_2 & + & x_3 & + & x_4 & + & x_5 & = & 2, \\
-4x_1 & + & 3x_2 & - & x_3 & - & x_4 & - & 3x_5 & = & -4, \\
3x_1 & + & 2x_2 & + & 3x_3 & + & 5x_4 & & & = & 3,
\end{array}
$$

$x_j \geq 0, \quad j = 1, \ldots, 5;$

$x^0 = (1, 0, 0, 0, 0).$

8. Решить симплекс-методом следующие задачи

8.1.

$$
\begin{array}{rcrcrcrcrcl}
-x_1 & + & x_2 & - & 2x_3 & + & 3x_4 & + & x_5 & \to & \max, \\
x_1 & + & 2x_2 & - & x_3 & - & 2x_4 & + & x_5 & \leq & 3, \\
-x_1 & - & x_2 & + & x_3 & + & 2x_4 & + & x_5 & \leq & 1, \\
2x_1 & + & x_2 & + & x_3 & - & x_4 & & & \leq & 1,
\end{array}
$$

$x_j \geq 0, \quad j = 1, \ldots, 5.$

8.2.

$$
\begin{array}{rcrcrcrcl}
x_1 & + & 2x_2 & - & 4x_3 & & & \to & \max, \\
x_1 & - & x_2 & - & x_3 & + & x_4 & \leq & 1, \\
2x_1 & - & x_2 & + & x_3 & & & \leq & 3, \\
-x_1 & + & 3x_2 & - & 2x_3 & - & x_4 & \leq & 2,
\end{array}
$$

$x_j \geq 0, \quad j = 1, \ldots, 4.$

8.3.

$$
\begin{array}{rcrcrcrcl}
x_1 & + & x_2 & + & x_3 & - & 2x_4 & \to & \min, \\
2x_1 & - & x_2 & & & + & x_4 & \le & 3, \\
x_1 & + & x_2 & + & x_3 & - & x_4 & \le & 1, \\
x_1 & + & 2x_2 & - & x_3 & & & \le & 1, \\
x_1 & + & 3x_2 & - & 2x_3 & + & x_4 & \le & 1, \\
x_j \ge 0, & & j = 1, \ldots, 4. & & & & & &
\end{array}
$$

9. Решить симплекс-методом следующие задачи, определив начальные вершины методом искусственного базиса:

9.1.

$$
\begin{array}{rcrcrcrcrcl}
-2x_1 & + & 2x_2 & + & x_3 & + & 2x_4 & - & 3x_5 & \to & \max, \\
-2x_1 & + & x_2 & - & x_3 & - & x_4 & & & = & 1, \\
x_1 & - & x_2 & + & 2x_3 & + & x_4 & + & x_5 & = & 4, \\
-x_1 & + & x_2 & & & & & - & x_5 & = & 4, \\
x_j \ge 0, & & j = 1, \ldots, 5. & & & & & & & &
\end{array}
$$

9.2.

$$
\begin{array}{rcrcrcrcrcl}
5x_1 & + & 4x_2 & + & 3x_3 & + & 2x_4 & - & 3x_5 & \to & \max, \\
2x_1 & + & x_2 & + & x_3 & + & x_4 & - & x_5 & = & 3, \\
x_1 & - & x_2 & & & + & x_4 & + & x_5 & = & 1, \\
-2x_1 & - & x_2 & - & x_3 & + & x_4 & & & = & 1, \\
x_j \ge 0, & & j = 1, \ldots, 5. & & & & & & & &
\end{array}
$$

9.3.

$$
\begin{array}{rcrcrcrcrcl}
2x_1 & + & x_2 & - & x_3 & + & 3x_4 & - & 2x_5 & \to & \max, \\
8x_1 & + & 2x_2 & + & 3x_3 & + & 9x_4 & + & 9x_5 & = & 30, \\
5x_1 & + & x_2 & + & 2x_3 & + & 5x_4 & + & 6x_5 & = & 19, \\
x_1 & + & x_2 & & & + & 3x_4 & & & = & 3, \\
x_j \ge 0, & & j = 1, \ldots, 5. & & & & & & & &
\end{array}
$$

10. Чаеразвесочная фабрика выпускает чай сортов *А* и *Б*, смешивая индийский, цейлонский и кенийский чай. В таблице приведены нормы расходов ингредиентов, объем запасов и прибыль от реализации 1 т чая сорта *А* и *Б*.

| Ингредиенты | Нормы расхода (т/т) | | Объем запасов (т) |
|---|---|---|---|
| | *А* | *Б* | |
| Индийский чай | 0,5 | 0,2 | 600 |
| Цейлонский чай | 0,2 | 0,6 | 870 |
| Кенийский чай | 0,3 | 0,2 | 430 |
| Прибыль от реализации 1 т продукции (усл.ед.) | 320 | 290 | |

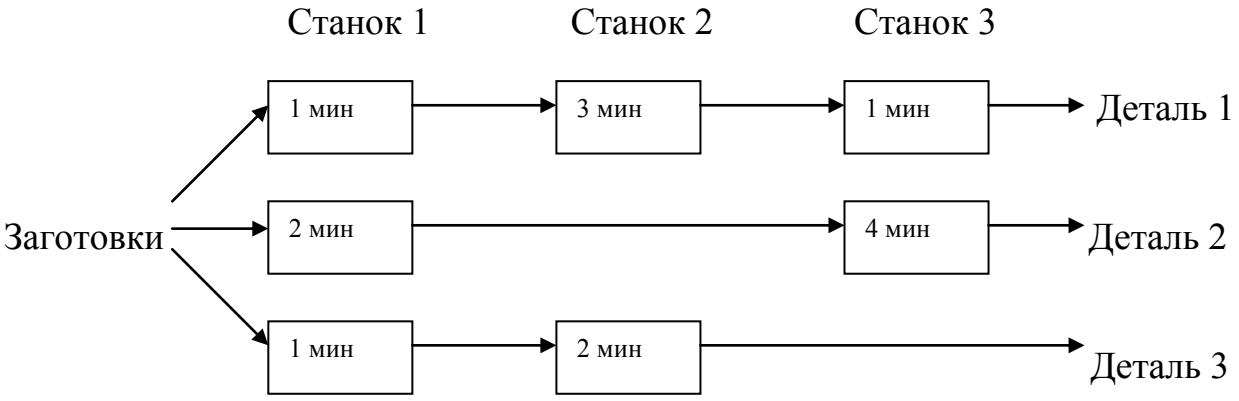
Требуется составить план производства с целью максимизации суммарной прибыли.

11. Рацион кормления коров на молочной ферме может состоять из трех продуктов – сена, силоса и концентратов. Эти продукты содержат белок, кальций и витамины (данные приведены в таблице).

| Продукты | Питательные вещества | | |
|---|---|---|---|
| | Белок (г/кг) | Кальций (г/кг) | Витамины (мг/кг) |
| Сено | 50 | 10 | 2 |
| Силос | 70 | 6 | 3 |
| Концентраты | 180 | 3 | 1 |

В расчете на одну корову суточные нормы потребления белка и кальция составляют не менее 2000 и 210 г соответственно. Потребление витаминов строго дозировано и должно быть равно 87 мг в сутки. Требуется составить самый дешевый рацион, если стоимость 1 кг сена, силоса и концентрата равна соответственно 1,5 , 2 и 6 усл.ед.

12. Цех выпускает три вида деталей, которые изготовляют на трех станках. На рисунке представлена технологическая схема изготовления детали каждого вида с указанием времени ее обработки на станках.




Суточный ресурс времени станков 1, 2 и 3 составляет соответственно 890, 920 и

840 мин. Стоимость одной детали вида 1, 2 и 3 равна соответственно 3, 1 и 2 усл.ед. Требуется составить суточный план производства с целью максимизации стоимости выпущенной продукции.

13. На мебельной фабрике требуется раскроить 5000 прямоугольных листов фанеры размером 4×5 м каждый, с тем чтобы получить два вида прямоугольных деталей: деталь *А* должна иметь размер 2×2 м, деталь *Б* – размер 1×3 м. Необходимо, чтобы деталей *А* оказалось не меньше, чем деталей *Б*. Каким образом следует производить раскрой, чтобы получить минимальное (по площади) количество отходов?

14. Для серийного производства некоторого изделия требуются комплексы заготовок профильного проката. Каждый комплект состоит из двух заготовок длиной 1800 мм и пяти заготовок длиной 700 мм. Как следует раскроить 770 полос проката стандартной длины 6000 мм, чтобы получить наибольшее количество указанных комплектов.

15. Металлургическому заводу требуется уголь с содержанием фосфора не более 0,03% и с долей зольных примесей не более 3,25%. Завод закупает три сорта угля *A, B, C* с известным содержанием примесей. Содержание примесей и цена исходных продуктов приведены в таблице

| Сорт угля | Содержание (%) | | Цена 1 т (усл.ед.) |
|---|---|---|---|
| | фосфора | золы | |
| *A* | 0,06 | 2,0 | 3 |
| *B* | 0,04 | 4,0 | 3 |
| *C* | 0,02 | 3,0 | 4,5 |

В какой пропорции нужно смешать исходные продукты *A, B, C*, чтобы смесь удовлетворяла ограничениям на содержание примесей и имела минимальную цену?

16. Для изготовления определённого сплава из свинца, цинка и олова используется сырьё в виде следующих пяти сплавов из тех же металлов, отличающихся составом и стоимостью 1 кг.

| Сплав / Компоненты | Содержание в % | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Свинец | 10 | 10 | 40 | 60 | 30 |
| Цинк | 10 | 30 | 50 | 30 | 20 |
| Олово | 80 | 60 | 10 | 10 | 50 |
| Стоимость | 4,0 | 4,5 | 5,8 | 6,0 | 7,5 |

Определить, сколько необходимо взять сплава каждого вида, чтобы изготовлять с минимальной себестоимостью сплав, содержащий 20% свинца, 30% цинка и 50% олова.

17. Компания производит полки для ванных комнат двух размеров *A* и *B*. Агенты по продаже считают, что на рынке может быть реализовано до 550 полок. Для каждой полки типа *A* требуется 2 м$^2$ материала, а для полки типа *B* – 3 м$^2$ материала. Компания может получить до 1200 м$^2$ материала в неделю. Для изготовления одной полки типа *A* требуется 12 мин. машинного времени, а для изготовления одной полки типа *B* – 30 мин.; оборудование можно использовать 160 ч. в неделю. Если прибыль от продажи полок типа *A* составляет 3 усл.ед., а от полок типа *B* – 4 усл.ед., то сколько полок каждого типа следует выпускать в неделю, чтобы прибыль, получаемая от реализации полок, была максимальной?

18. Фирма производит три вида продукции *A*, *B*, *C*, для выпуска каждой из которых требуется определённое время обработки на всех четырёх устройствах 1, 2, 3, 4.

| Вид продукции | Время обработки (ч.) | | | | Прибыль (усл.ед.) |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| A | 1 | 3 | 1 | 2 | 3 |
| B | 6 | 1 | 3 | 3 | 6 |
| C | 3 | 3 | 2 | 4 | 4 |

Пусть время работы на устройствах – соответственно 84, 42, 21 и 42 ч. Определить, какую продукцию и в каких количествах следует производить. (Предполагается, что рынок сбыта для каждого продукта не ограничен; временем, требуемым для переключения устройства в зависимости от вида продукции, можно пренебречь; рассмотрите только задачу максимизации прибыли).

19. В области имеются два цементных завода и три потребителя их продукции – домостроительные комбинаты. В таблице указаны суточные объемы производства цемента, суточные потребности в нем комбинатов и стоимость перевозки 1 т цемента от каждого завода к каждому комбинату.

| Заводы | Производство цемента (т/сут) | Стоимость перевозки 1 т цемента (усл.ед.) | | |
|---|---|---|---|---|
| | | Комбинат 1 | Комбинат 2 | Комбинат 3 |
| 1 | 40 | 10 | 15 | 25 |
| 2 | 60 | 20 | 30 | 35 |
| | Потребность в цементе (т/сут) | 50 | 20 | 30 |

Требуется составить план суточных перевозок цемента с целью минимизации транспортных расходов.

20. В некоторой местности в двух пунктах *A* и *B* имеется потребность в дополнительном транспорте. В пункте *A* требуется 5 дополнительных автобусов, а в пункте *B* – 7. Известно, что 3, 4 и 5 автобусов могут быть получены соответственно из гаражей $G_1$, $G_2$ и $G_3$. Как следует распределить эти автобусы между пунктами *A* и *B*, чтобы минимизировать их суммарный пробег? Расстояния от гаражей до пунктов *A* и *B* приведены в таблице.

| Гараж | Расстояние до пунктов | |
|---|---|---|
| | A | B |
| $G_1$ | 3 | 4 |
| $G_2$ | 1 | 3 |
| $G_3$ | 4 | 2 |

21. При откорме животных каждое животное ежедневно должно получить не менее 60 ед. питательного вещества *A*, не менее 50 ед. вещества *B* и не менее 12 ед. вещества *C*. Указанные питательные вещества содержат три вида корма. Содержание единиц питательных веществ в 1 кг каждого из видов корма приведено в следующей таблице:

| Питательные вещества | Количество единиц питательных веществ в 1 кг корма вида | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| *A* | 1 | 3 | 4 |
| *B* | 2 | 4 | 2 |
| *C* | 1 | 4 | 3 |

Составьте дневной рацион, обеспечивающий получение необходимого количества питательных веществ при минимальных денежных затратах, если цена 1 кг корма 1 вида составляет 9 руб., корма 2 вида – 12 руб. и 3 вида – 10руб.

22. Для серийного изготовления детали механический цех может использовать пять различных технологий обработки на токарном, фрезерном, строгальном и шлифовальном станках. В таблице указано время (в минутах) обработки детали на каждом станке в зависимости от технологического способа, а также общий ресурс рабочего времени каждого станка за смену.

| Станки | Технологические способы | | | | | Ресурс времени станков (мин) |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Токарный | 2 | 1 | 3 | 0 | 1 | 4100 |
| Фрезерный | 1 | 0 | 2 | 2 | 1 | 2000 |
| Строгальный | 1 | 2 | 0 | 3 | 2 | 5800 |
| Шлифовальный | 3 | 4 | 2 | 1 | 1 | 10800 |

Требуется указать технологию, максимизирующую выпуск.

## §2. Применение Microsoft Excel для решения задач линейного программирования

Для решения задач конечномерной оптимизации могут применяться различные пакеты прикладных программ, такие как, Matlab, Maple, Mathcad. Между тем, наиболее доступным является Excel, а точнее его инструмент **Поиск решения**(Р ис. 2.1). Он может применяться как к линейным, так и нелинейным оптимизационным задачам.

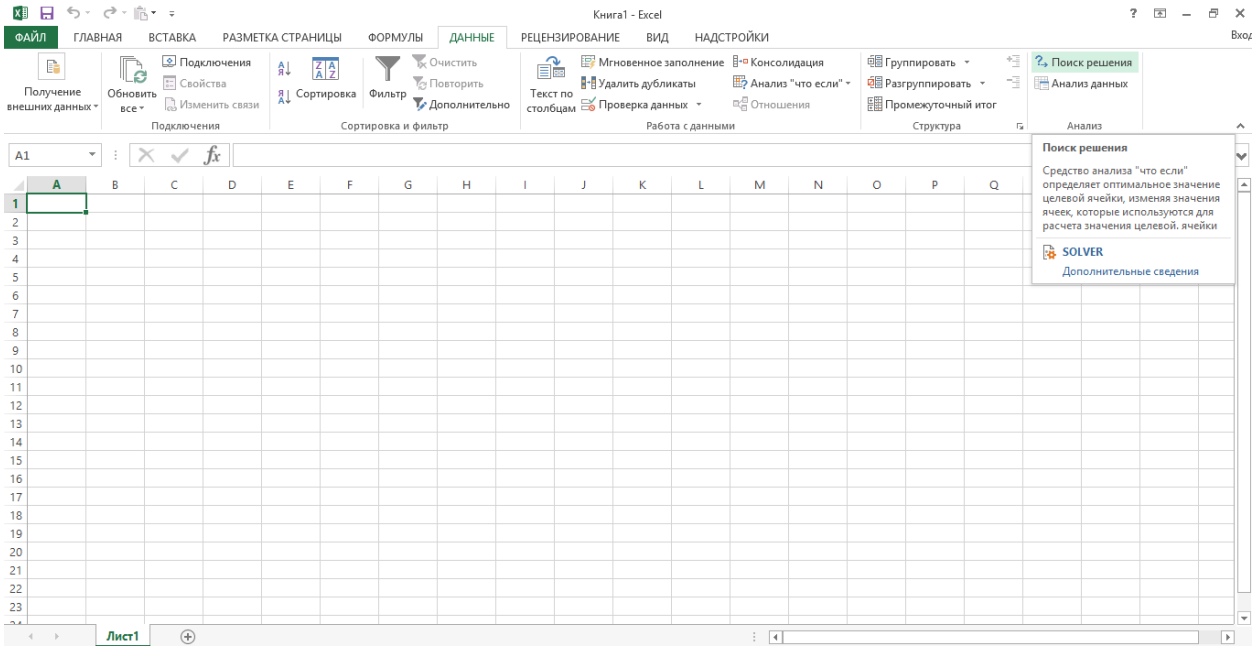



Рис. 2.1

При запуске **Поиск решения**, открывается окно диалога, показанное на рис. 2.2

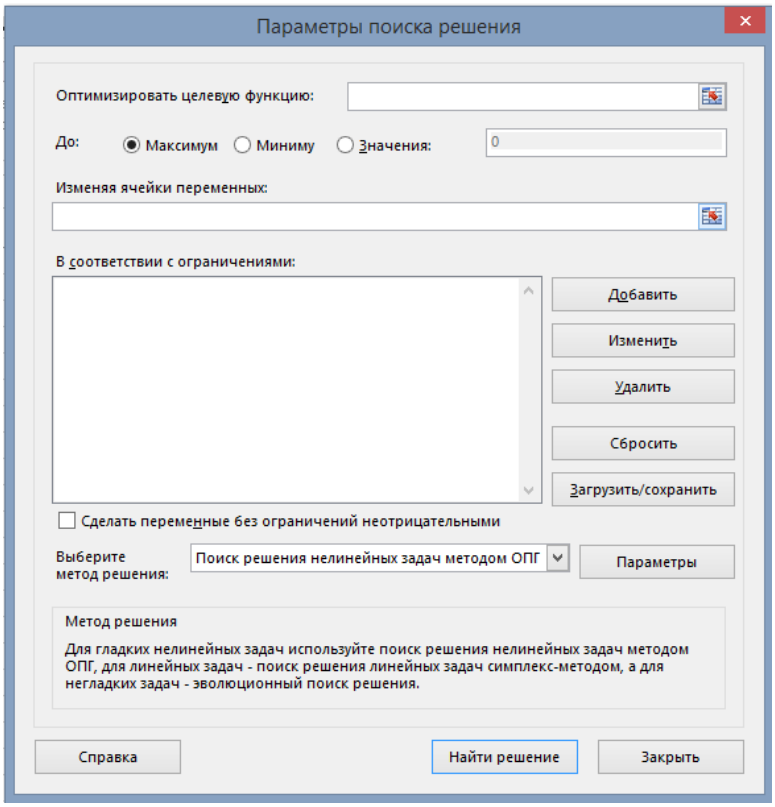



Рис. 2.2

Следует отметить, что для задач линейного программирования целесообразно выбрать Симплекс-метод поиска решения (Рис. 2.3).

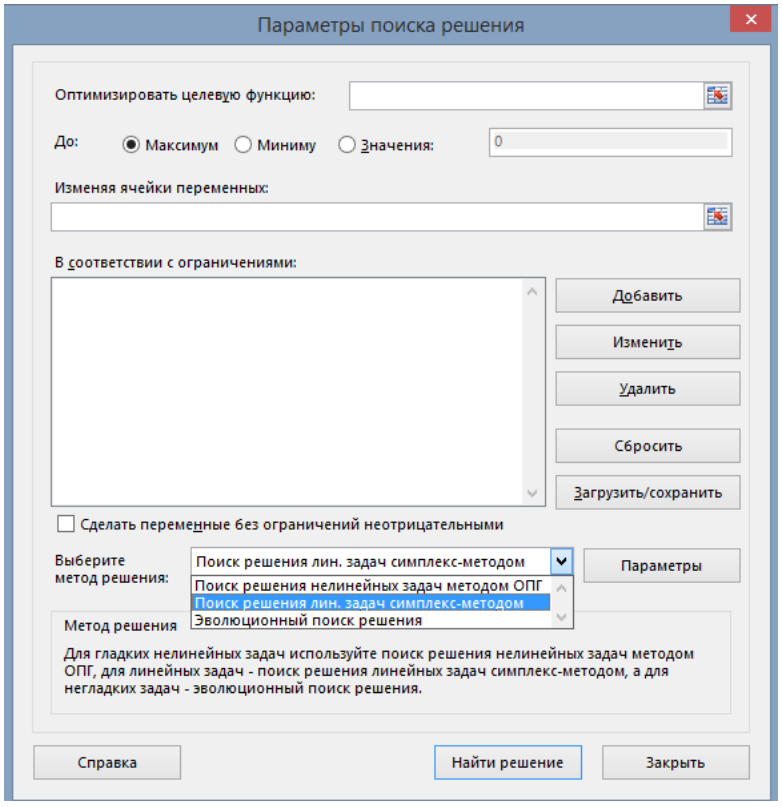



Рис. 2.3

Далее рассмотрим процесс решения задачи оптимизации в Excel на примере следующей задачи линейного программирования:

$$
\begin{aligned}
&2x_1 + x_2 + 3x_3 + 5x_4 \to \max,\\
&2x_1 + 3x_2 + x_3 + 2x_4 \le 30,\\
&4x_1 + 2x_2 + x_3 + 2x_4 \le 40,\\
&x_1 + 2x_2 + 3x_3 + x_4 \le 25,\\
&x_i \ge 0,\ i = 1,2,3,4.
\end{aligned}
$$

Прежде всего, на рабочем листе определим ячейки, которые соответствуют переменным $x_1, x_2, x_3, x_4$: ячейки C2:C5. Далее в ячейке C7 зададим целевую функцию, а в ячейках C9:C11 – правые части ограничений (Рис. 2.4).

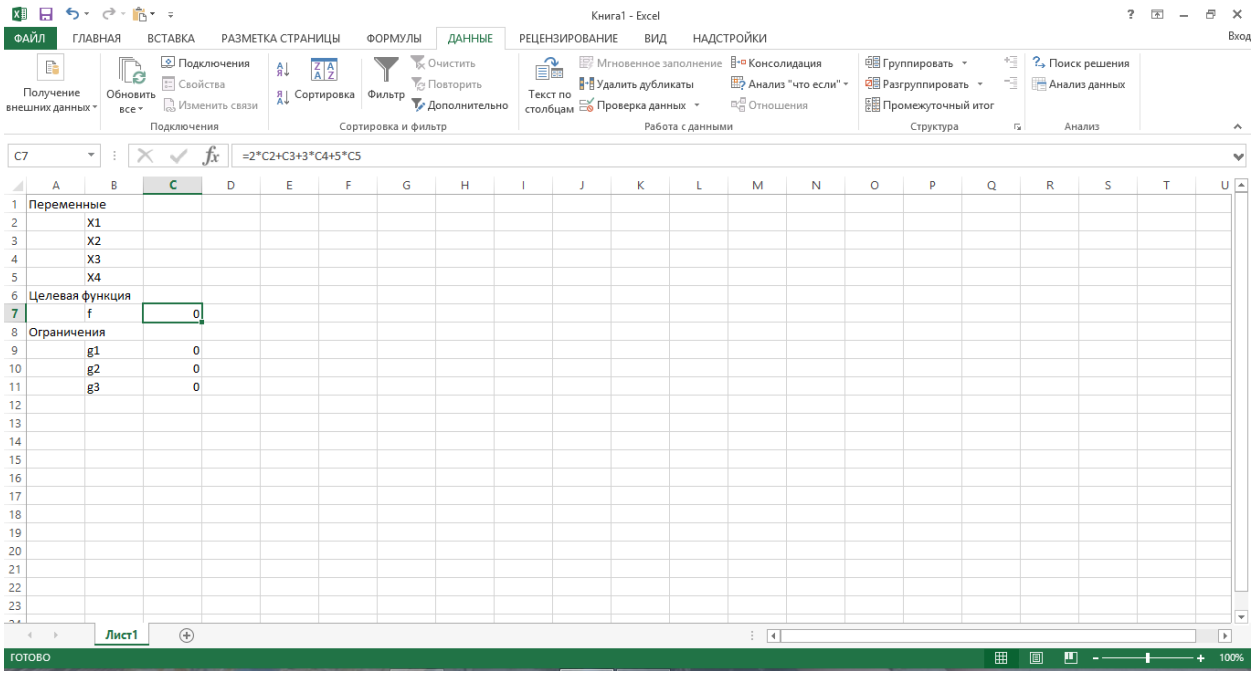



Рис. 2.4

После запуска **Поиск решения** открывается окно диалога. В поле **Оптимизировать целевую функцию** следует задать целевую ячейку, введя ссылку на ячейку C7 или назначенное ей имя либо щелкнув на ячейке в рабочем листе. В данном примере требуется найти наибольшее возможное значение, поэтому в группе **До** переключатель следует установить **Максимум**.

Следующим шагом в поле **Изменяя ячейки переменных** необходимо задать ячейки с переменными, указав на них ссылки или их имени либо выделив ячейки в рабочем листе. В рассматриваемом примере это будут ячейки C2:C5 (Рис. 2.5).

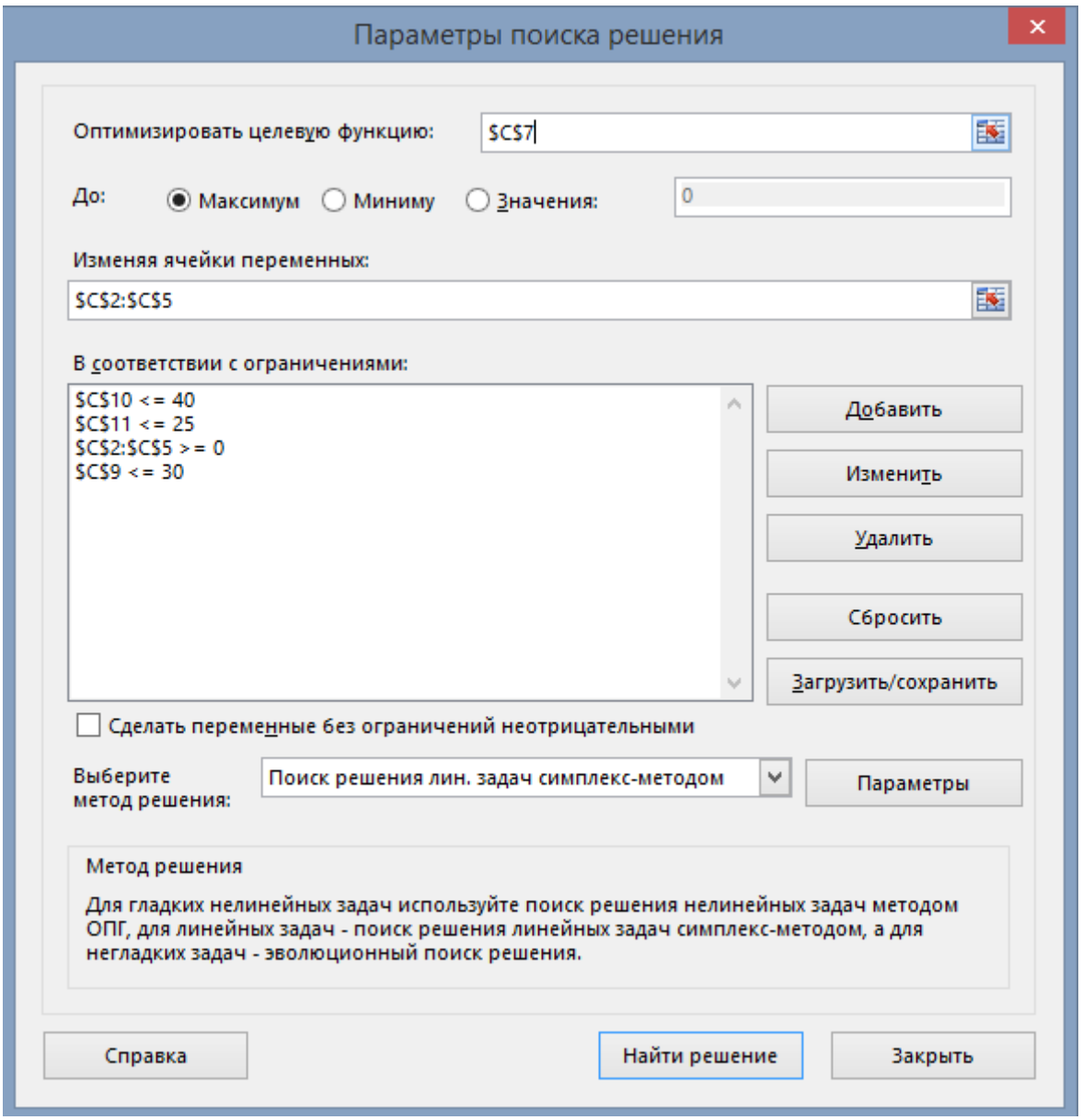



Рис. 2.5

Последний шаг, задание ограничений. Чтобы задать ограничения, в окне диалога Поиск решения следует нажать кнопку **Добавить** и заполнить окно диалога **Добавление ограничения.** Ограничение состоит из трех компонентов: ссылки на ячейку, оператора сравнения и значения ограничения. После задания ограничения нажмите кнопку ОК, чтобы вернуться в окно диалога Поиск решения, или нажмите кнопку Добавить для задания следующего ограничения (Рис. 2.6).

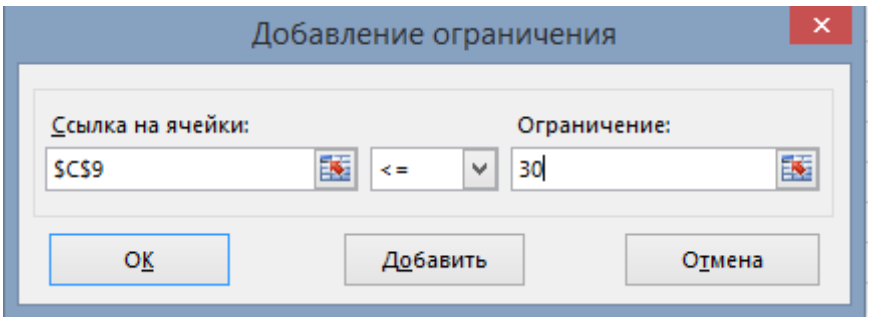



Рис. 2.6

Следует отметить, что в случае решения задач целочисленного программирования на ячейки переменных следует наложить ограничение целочисленности: в окне диалога Добавление ограничения в качестве оператора сравнения необходимо указать пункт **цел.**

После заполнения окна диалога Поиск решения следует нажать кнопку Найти решение. **Поиск решения** находит оптимальное решение для заданной целевой ячейки при выполнении всех ограничений и выводит окно диалога (Рис. 2.7):

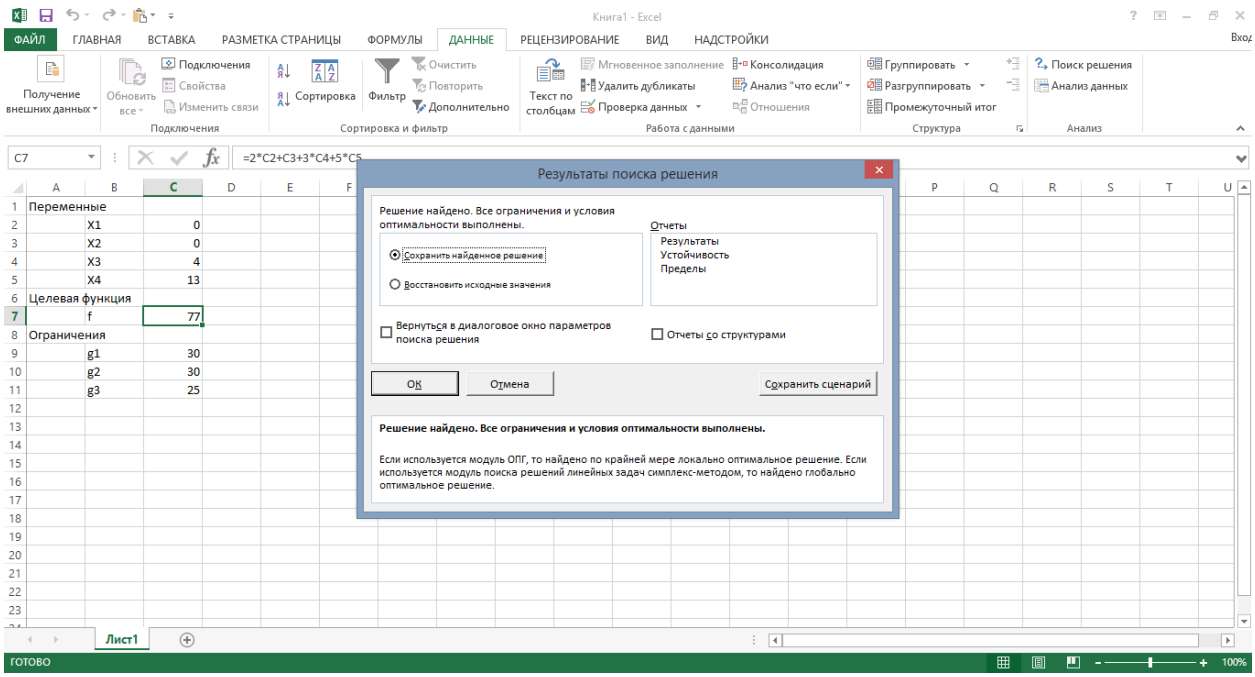



Рис. 2.7

Значения, отображаемые в рабочем листе, представляют собой оптимальное решение задачи. Можно либо оставить эти значения в листе, если установить переключатель **Сохранить найденное решение** и нажать кнопку ОК, либо восстановить значения, которые содержались в листе перед активизацией поиска решения, если нажать кнопку Отмена или установить переключатель **Восстановить исходные значения** и нажать кнопку ОК. Можно также сохранить найденные значения в качестве сценария.

Кроме вставки оптимальных значений в изменяемые ячейки поиск решения позволяет представлять результаты в виде трех отчетов: **Результаты**, **Устойчивость** и **Пределы**. Для генерации одного или нескольких отчетов следует выделить их названия в окне диалога Результаты поиска решения. Каждый отчет сохраняется на отдельном листе текущей книги, а названия отчетов отображаются на ярлычках.

**Отчет по устойчивости** содержит информацию о том, насколько целевая ячейка чувствительна к изменениям ограничений и переменных. Этот отчет имеет два раздела: один для изменяемых ячеек, а второй для ограничений (Рис. 2.8).

**Microsoft Excel 15.0 Отчет об устойчивости**
**Лист: [Книга1]Лист1**

Ячейки переменных

| Ячейка | Имя | Окончательное Значение | Приведенн. Стоимость | Целевая функция Коэффициент | Допустимое Увеличение | Допустимое Уменьшение |
|---|---|---|---|---|---|---|
| $C$2 | X1 | 0 | -3 | 2 | 3 | 1E+30 |
| $C$3 | X2 | 0 | -6,6 | 1 | 6,6 | 1E+30 |
| $C$4 | X3 | 4 | 0 | 3 | 12 | 0,5 |
| $C$5 | X4 | 13 | 0 | 5 | 1 | 3 |

Ограничения

| Ячейка | Имя | Окончательное Значение | Тень Цена | Ограничение Правая сторона | Допустимое Увеличение | Допустимое Уменьшение |
|---|---|---|---|---|---|---|
| $C$10 | g2 | 30 | 0 | 40 | 1E+30 | 10 |
| $C$11 | g3 | 25 | 0,2 | 25 | 65 | 10 |
| $C$9 | g1 | 30 | 2,4 | 30 | 10 | 21,66666667 |

Рис. 2.8

**Отчет по результатам** содержит целевую ячейку, список изменяемых ячеек и ограничений (Рис. 2.9). Этот отчет также содержит информацию о таких параметрах каждого ограничения, как значение ячейки, состояние и допуск. Допуск – это разность между значением, выводимым в ячейке ограничения при получении решения, и числом, заданным в правой части формулы ограничения.

**Результат: Решение найдено. Все ограничения и условия оптимальности выполнены.**

**Модуль поиска решения**

Модуль: Поиск решения лин. задач симплекс-методом

Время решения: 0,016 секунд.

Число итераций: 2 Число подзадач: 0

**Параметры поиска решения**

Максимальное время Без пределов, Число итераций Без пределов, Precision 0,000001

Максимальное число подзадач Без пределов, Максимальное число целочисленных решений Без пределов, Целочисленное отклонение 1%

Ячейка целевой функции (Максимум)

| Ячейка | Имя | Исходное значение | Окончательное значение |
|---|---|---|---|
| $C$7 | f | 0 | 77 |

Ячейки переменных

| Ячейка | Имя | Исходное значение | Окончательное значение | Целочисленное |
|---|---|---|---|---|
| $C$2 | X1 | 0 | 0 | Продолжить |
| $C$3 | X2 | 0 | 0 | Продолжить |
| $C$4 | X3 | 0 | 4 | Продолжить |
| $C$5 | X4 | 0 | 13 | Продолжить |

Ограничения

| Ячейка | Имя | Значение ячейки | Формула | Состояние | Допуск |
|---|---|---|---|---|---|
| $C$10 | g2 | 30 | $C$10<=40 | Без привязки | 10 |
| $C$11 | g3 | 25 | $C$11<=25 | Привязка | 0 |
| $C$9 | g1 | 30 | $C$9<=30 | Привязка | 0 |
| $C$2 | X1 | 0 | $C$2>=0 | Привязка | 0 |
| $C$3 | X2 | 0 | $C$3>=0 | Привязка | 0 |
| $C$4 | X3 | 4 | $C$4>=0 | Без привязки | 4 |
| $C$5 | X4 | 13 | $C$5>=0 | Без привязки | 13 |

Рис. 2.9

**Отчет по пределам** содержит информацию о том, в каких пределах значения изменяемых ячеек могут быть увеличены или уменьшены без нарушения ограничений задачи (Рис. 2.10).

**Microsoft Excel 15.0 Отчет об устойчивости**

**Лист: [Книга1]Лист1**

Ячейки переменных

| Ячейка | Имя | Окончательное Значение | Приведенн. Стоимость | Целевая функция Коэффициент | Допустимое Увеличение | Допустимое Уменьшение |
|---|---|---|---|---|---|---|
| $C$2 | X1 | 0 | -3 | 2 | 3 | 1E+30 |
| $C$3 | X2 | 0 | -6,6 | 1 | 6,6 | 1E+30 |
| $C$4 | X3 | 4 | 0 | 3 | 12 | 0,5 |
| $C$5 | X4 | 13 | 0 | 5 | 1 | 3 |

Ограничения

| Ячейка | Имя | Окончательное Значение | Тень Цена | Ограничение Правая сторона | Допустимое Увеличение | Допустимое Уменьшение |
|---|---|---|---|---|---|---|
| $C$10 | g2 | 30 | 0 | 40 | 1E+30 | 10 |
| $C$11 | g3 | 25 | 0,2 | 25 | 65 | 10 |
| $C$9 | g1 | 30 | 2,4 | 30 | 10 | 21,66666667 |

Рис. 2.10

Для каждой изменяемой ячейки этот отчет содержит оптимальное значение, а также наименьшее и наибольшее значения, которые ячейка может принимать без нарушения ограничений.

**Литература**

1. Алексеев, В. М. Сборник задач по оптимизации. Теория. Примеры. Задачи: Учеб. пособие / В. М. Алексеев, Э. М. Галеев, В. М. Тихомиров. – Москва: ФИЗМАТЛИТ, 2011. – 256 с.
2. Аттетков, А. В. Методы оптимизации: учебное пособие / А. В. Аттетков, В. С. Зарубин, А. Н. Канатников. – Москва: РИОР: ИНФРА-М, 2019. – 270 с.
3. Ашманов, С. А. Теория оптимизации в задачах и упражнениях: учебное пособие / С. А. Ашманов, А. В. Тимохов. – Санкт-Петербург: Лань, 2012. – 448 с.
4. Балдин, К. В. Методы оптимальных решений: учебник / К. В. Балдин, В. Н. Башлыков, А. В. Рукосуев; под общ. ред. К. В. Балдина. – Москва: ФЛИНТА, 2020. – 323 с.
5. Пантелеев, А. В. Методы оптимизации. Практический курс: учебное пособие / А. В. Пантелеев, Т. А. Летова. – Москва: Логос, 2020. – 424 с.
6. Сдвижков, О. А. Практикум по методам оптимизации: учебное пособие / О. А. Сдвижков. – Москва: Вузовский учебник: ИНФРА-М, 2020. – 231 с.
7. Сухарев, А. Г. Методы оптимизации: учебник и практикум для бакалавриата и магистратуры / А. Г. Сухарев, А. В. Тимохов, В. В. Федоров. – М. : Издательство Юрайт, 2014 – 367 с.

Юрий Алексеевич **Кузнецов**
Алексей Валерьевич **Семенов**

# МЕТОДЫ ОПТИМИЗАЦИИ: ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

*Учебно-методическое пособие*

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный
университет им. Н. И. Лобачевского»
603950, Нижний Новгород, пр. Гагарина, 23.