ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ

Кафедра «Программное обеспечение и управление
в технических системах»

# УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ
## по дисциплине
## «МЕТОДЫ ОПТИМИЗАЦИИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ»
## к практическим занятиям по теме
## «Решение задачи линейного программирования с помощью симплекс метода»

**для студентов очной и заочной формы обучения по специальностям 230105 – программное обеспечение вычислительной техники и автоматизированных систем, 230201 – информационные системы и технологии и направлениям подготовки бакалавриата 230100 – информатика и вычислительная техника, 230400 – информационные системы и технологии**

Составитель:
К.ф.-м.н., доцент Вержаковская М.А.

САМАРА
ИУНЛ ПГУТИ
2011

Вержаковская М.А. Учебно-методические указания по дисциплине «Методы оптимизации в информационных системах» к практическим занятиям по теме «Решение задачи линейного программирования с помощью симплекс метода» – Самара: ПГУТИ, 2011. – 46 с., ил.

Учебно-методические указания предназначены для студентов очной и заочной форм обучения по специальностям 230105 – программное обеспечение вычислительной техники и автоматизированных систем, 230201 – информационные системы и технологии, а также по направлениям подготовки бакалавриата 230100 – информатика и вычислительная техника, 230400 – информационные системы и технологии. Данные указания служат руководством для подготовки к практическим занятиям по дисциплине «Методы оптимизации в информационных системах» на тему «Решение задачи линейного программирования с помощью симплекс метода».

Учебно-методические указания подготовлены на кафедре «Программное обеспечение и управление в технических системах».

Учебно-методические указания рекомендованы к изданию методическим Советом ПГУТИ

Рецензент – заведующий кафедрой программного обеспечения и управления в технических системах ГОУВПО ПГУТИ, д.т.н., профессор Тарасов В.Н.

## СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Данный курс предназначен для студентов специальностей по направлениям «Программное обеспечение вычислительной техники и автоматизированных систем», «Информационные системы и технологии», изучающих дисциплину «Методы оптимизации в информационных системах».

Целью преподавания дисциплины «Методы оптимизации в информационных системах» является изучение теоретических основ моделирования и решения задач математического программирования. Цель преподавания дисциплины также состоит в усвоении роли методов оптимизации в повышении эффективности устройств автоматики и систем управления, формирования знаний и умений по постановке и решению оптимизационных задач.

Задачи дисциплины:

- обучить студентов основным методам решения оптимизационных задач;
- привить студентам устойчивые навыки математического моделирования с использованием ЭВМ;
- дать опыт проведения вычислительных экспериментов.

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Примеры задач линейного программирования

*Задача планирования выпуска продукции (планирование производства)*

Машиностроительное предприятие для изготовления четырех видов продукции использует токарное, фрезерное, сверлильное, расточное и шлифовальное оборудование, а также комплектующие изделия. Кроме того, сборка изделий требует выполнения сборочно-наладочных работ. Нормы затрат всех видов ресурсов на изготовление каждого из изделий приведены в таблице 1.1. В этой же таблице указан имеющийся фонд каждого из ресурсов и прибыль от реализации одного изделия каждого вида.

Таблица 1.1 − Нормы затрат на изготовление одного изделия

| Ресурсы | Нормы затрат на изготовление одного изделия | | | | Общий объем ресурсов |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| Производительность оборудования (чел.-час) | | | | | |
| Токарного | 550 | - | 620 | - | 64270 |
| Фрезерного | 40 | 30 | 20 | 20 | 4800 |
| Сверлильного | 86 | 110 | 150 | 52 | 22360 |
| Расточного | 160 | 92 | 158 | 128 | 26240 |
| Шлифовального | - | 158 | 30 | 50 | 7900 |
| Комплект изделия (шт.) | 3 | 4 | 3 | 3 | 520 |
| Сборочно-наладочные работы (чел.-час.) | 4,5 | 4,5 | 4,5 | 4,5 | 720 |
| Прибыль от реализации одного изделия | 315 | 278 | 573 | 370 | |

Надо составить такой план выпуска продукции, чтобы получить максимальную прибыль. Построим математическую модель данной задачи. Пусть $x_1$ – планируемое количество изделий 1-го типа, $x_2$, $x_3$, $x_4$ – планируемое количество изделий 2-го, 3-го, 4-го типов соответственно.

Тогда

$$f = 315x_1 + 278x_2 + 573x_3 + 370x_4 \quad (1.1)$$

прибыль предприятия. Так как нам надо получить как можно большую прибыль, то будем искать наибольшее (максимальное) значение функции $f$. Общие объемы ресурсов накладывают на нашу задачу ограничения

$$550x_1 + 620x_2 \le 64270$$

$40x_1 + 30x_2 + 20x_3 + 20x_4 \le 4800$
$86x_1 + 110x_2 + 150x_3 + 52x_4 \le 22360$
$160x_1 + 92x_2 + 158x_3 + 128x_4 \le 26240$ (1.2)
$158x_2 + 30x_3 + 50x_4 \le 7900$
$3x_1 + 4x_2 + 3x_3 + 3x_4 \le 520$
$4{,}5x_1 + 4{,}5x_2 + 4{,}5x_3 + 4{,}5x_4 \le 720.$

Так как целевая функция и ограничения являются линейными, то это задача линейного программирования.

*Задача планирования капитальных вложений*

Для электроснабжения трех стройплощадок используются две трансформаторные подстанции. Первая стройплощадка потребляет 130 кВт, а вторая и третья по 140 кВт. Мощность первой ТП – 250 кВт, а второй 160 кВт. Расстояние от первой ТП – 600, 400, 200 м, до первой, второй и третьей площадки соответственно. От второй ТП: − 500, 300 и 200 м соответственно. Требуется составить схему электроснабжения с минимальными капитальными вложениями.

Капитальные вложения для устройства линий рассчитываются по следующей формуле:

$$\sum_{j=1}^{3}\sum_{i=1}^{2}\frac{S_{ij}K_0}{\sqrt{3U_j}}l_{ij} = \alpha\sum_{j=1}^{3}\sum_{i=1}^{2}S_{ij}l_{ij}, \qquad (1.3)$$

где: $S_{ij}$ – мощность, потребляемая $j$-ой площадкой от $i$-ой ТП, кВт;
$l_{ij}$ – расстояние от $j$-ой площадки до $i$-ой ТП, м.;
$\alpha$ – постоянный множитель.

Итак, нам надо минимизировать функцию

$$f = \alpha\,(6S_{11} + 4S_{12} + 2S_{13} + 5S_{21} + 3S_{22} + 2S_{23}), \qquad (1.4)$$

при ограничениях:

$S_{11} + S_{21} = 130;$

$S_{12} + S_{22} = 140;$

$S_{13} + S_{23} = 140;$ (1.5)

$S_{11} + S_{12} + S_{13} = 250;$

$S_{21} + S_{22} + S_{23} = 160.$

Как и в предыдущем примере имеем задачу линейного программирования.

## 1.2 Основные определения

**ОПРЕДЕЛЕНИЕ 1.1.** Задача, состоящая в нахождении наибольшего (наименьшего) значения функции

$$f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \tag{1.6}$$

на множестве точек $x^T = (x_1, \ldots, x_\text{n})$, удовлетворяющих системе ограничений вида:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n < R_1 > b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n < R_2 > b_2 \\ ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n < R_m > b_m \end{cases} \tag{1.7}$$

называется задачей *линейного программирования общего вида*.

Здесь:

$f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n$ – целевая функция;

$R_i,\ i = \overline{1,m}$ – операции отношения =, ≥, ≤;

$c_i,\ i = \overline{1,n};\ a_{ij}, i = \overline{1,m},\ j = \overline{1,n};\ b_i,\ i = \overline{1,m}$ – заданные константы.

**ОПРЕДЕЛЕНИЕ 1.2.** Всякую точку $x^T = (x_1, \ldots, x_\text{n})$, компоненты которой удовлетворяют всем ограничениям системы (1.2), будем называть допустимой точкой или допустимым решением задачи, или допустимым планом задачи.

Задача линейного программирования состоит, таким образом, в нахождении такой допустимой точки $\boldsymbol{x}^{(0)}$ (такого допустимого плана) среди множества допустимых точек, при которой целевая функция принимает max (min) значение. Допустимое решение $\boldsymbol{x}^{(0)} = (x_1^{(0)}, \ldots, x_\text{n}^{(0)})^\text{T}$, доставляющее целевой функции оптимальное значение (оптимум), будет называться оптимальным решением или оптимальным планом задачи. В дальнейшем будем говорить, к примеру, только о нахождении max целевой функции.

Задачу линейного программирования, представленную в виде:

$$f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \to \max(\min)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \end{cases} \tag{1.8}$$

$$x_i \geq 0,\ i = \overline{1,n}$$

будем считать *канонической задачей линейного программирования*.

## 1.3 Геометрическая интерпретация двумерной задачи линейного программирования и ее решение

Рассмотрим двумерную задачу:

$$f(x) = c_1 x_1 + c_2 x_2 \to \max(\min) \qquad (1.9)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 < R_1 > b_1 \\ a_{21}x_1 + a_{22}x_2 < R_2 > b_2 \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 < R_m > b_m \end{cases} \qquad (1.10)$$

Каждое из ограничений $a_{i1}x_1 + a_{i2}x_2 < R_i > b_i$ определяет в плоскости, с системой координат $x_1$ о $x_2$ множество точек, лежащих по одну сторону от прямой $a_{i1}x_1 + a_{i2}x_2 = b_i$ (т.е. полуплоскость). Множество всех точек плоскости, координаты которых удовлетворяют всем ограничениям, т.е. принадлежат сразу всем полуплоскостям, определяемым отдельными ограничениями, будет представлять собой допустимое множество. Очевидно, что, если множество не пусто, то это будет некоторый многоугольник (возможно и неограниченный, возможно вырождающийся в отрезок, точку). Многоугольник будет выпуклым, что легко показать, т.е. любые две его точки можно соединить отрезком, точки которого принадлежат допустимому множеству.

**ОПРЕДЕЛЕНИЕ 1.3.** Множество $D$-точек $n$-мерного евклидова пространства будем называть выпуклым, если для любых $\boldsymbol{x}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})^T$ и $\boldsymbol{x}^{(2)} = (x_1^{(2)}, \ldots, x_n^{(2)})^T$ из множества $D$ и любых $\alpha \geq 0$, $\beta \geq 0$ таких, что $\alpha + \beta = 1$, точка $\boldsymbol{x} = \alpha\boldsymbol{x}^{(1)} + \beta\boldsymbol{x}^{(2)}$ также принадлежит $D$.

**ОПРЕДЕЛЕНИЕ 1.4.** Вершиной выпуклого множества в $R_n$ назовем такую точку, которую нельзя представить в виде $\boldsymbol{x} = \alpha\boldsymbol{x}^{(1)} + \beta\boldsymbol{x}^{(2)}$, $\alpha > 0$, $\beta > 0$, $\alpha + \beta = 1$, ни при каких $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$.

Рассмотрим теперь геометрическую интерпретацию решения задачи линейного программирования. Пусть допустимая область задачи (1.9)−(1.10) оказалась непустой (в соответствии с рисунком 1.1).

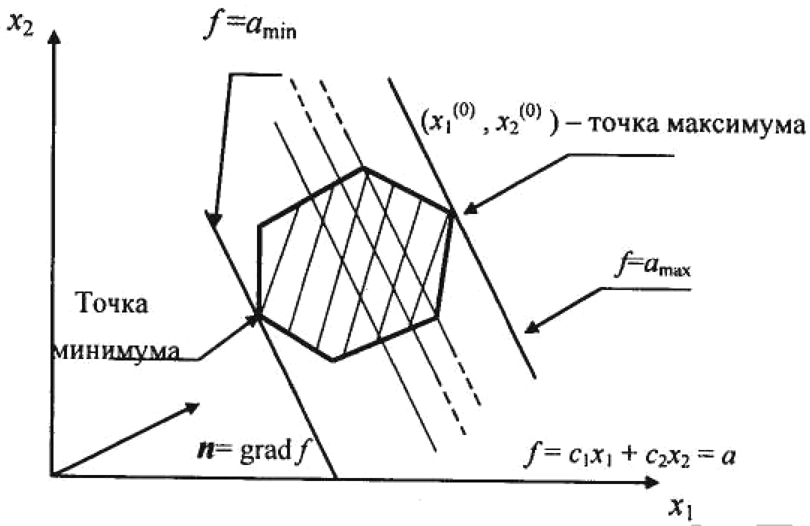


Рисунок 1.1 – Геометрическая интерпретация задачи линейного программирования

Мы хотим найти те точки допустимой области, координаты которых доставляют целевой функции наибольшее значение. Построим линию уровня целевой функции $f = c_1x_1 + c_2x_2 = a$. Получим множество точек, в точках которого $f$ принимает одно и тоже значение $a$. Перемещая линию уровня в направлении вектора $grad\ f = (c_1, c_2) = \boldsymbol{n}$, нормального к линии уровня, будем получать в пересечении этой линии с допустимой областью точки, в которых целевая функция принимает новое значение, большее чем значение на предшествующих линиях уровня.

Пересечение допустимой области с линией уровня в том ее положении, когда дальнейшее перемещение дает пустое множество, и будет множеством *точек максимума* задачи линейного программирования. Перемещая линию уровня в направлении противоположном вектору $\boldsymbol{n}$, аналогичным образом найдем *точки минимума*.

**Пример 1.1.**

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

$$\begin{cases} -x_1 + x_2 \le 1 \\ x_1 - 2x_2 \le 1 \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0$$

Точка максимума:

$\boldsymbol{x} = (0,\ 1)^{T}$;

$f_{max} = f(0,\ 1) = 1$.

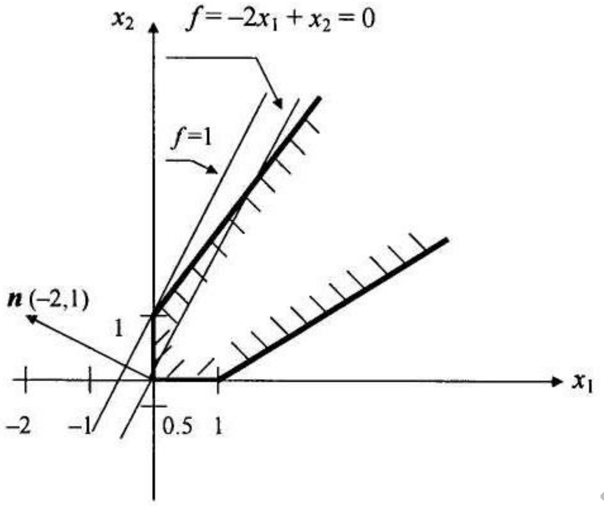


Рисунок 1.2 – Геометрическая интерпретация задачи линейного программирования

**Пример 1.2.**

Легко представить, что задача имеет бесчисленное множество оптимальных решений, а может их не иметь вовсе, как например:

$$f(x) = x_1 + x_2 \to \min(\max)$$

$$\begin{cases} x_1 + x_2 \ge 1 \\ -x_1 + x_2 \le 1 \\ x_1 - 2x_2 \le 2 \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0$$

Целевая функция достигает минимального значения в точках отрезка $x_1 + x_2 = 1$, между точками (0, 1) и (1, 0), а максимального значения функция не достигает (не ограничена в допустимой области).

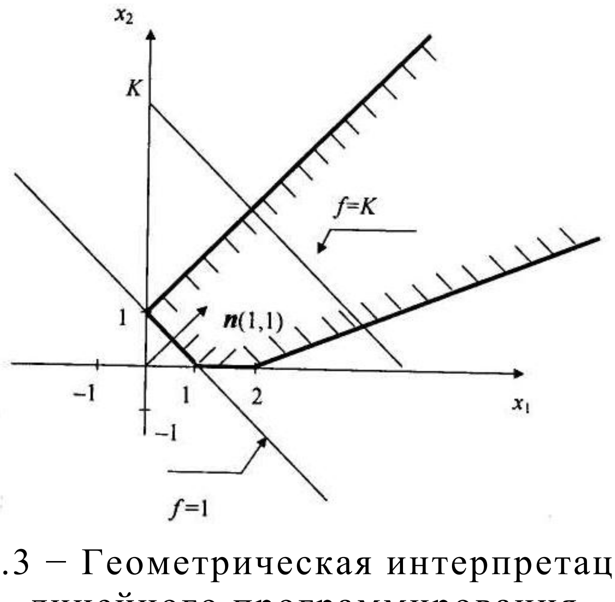


Рисунок 1.3 − Геометрическая интерпретация задачи линейного программирования

## 1.4 Свойства задачи линейного программирования

Рассмотренные примеры позволяют сформулировать основные свойства задачи линейного программирования.

***Свойство 1.*** Допустимая область задачи линейного программирования выпукла, если она не пуста.

***Свойство 2.*** Если допустимая область имеет вершины и задача линейного программирования имеет решение, то оно достигается по крайней мере в одной из вершин.

***Свойство 3.*** Множество решений задачи линейного программирования выпукло.

***Свойство 4.*** Если допустимая область ограничена, то любая задача линейного программирования имеет оптимальное решение.

***Свойство 5.*** Необходимым и достаточным условием существования решения задачи линейного программирования на максимум (минимум) является ограниченность целевой функции сверху (соответственно снизу) в допустимой области.

Все перечисленные свойства справедливы и в общем случае ($n \geq 2$).

Свойства задачи линейного программирования наталкивают на следующую схему решения задачи линейного программирования, известную как симплекс-метод. Именно: пусть рассматриваемая задача имеет непустое допустимое множество с вершинами.

Тогда:

1) тем или иным способом находим какую-нибудь вершину допустимого множества и по определенным критериям определяем, не является ли она оптимальной. Если вершины нет, то допустимая область пуста. Если вершина оптимальна, то задача решена.

Если нет, то

2) используя определенные правила проверяем, нельзя ли утверждать, что задача не имеет оптимального решения (целевая функция не ограничена сверху или, соответственно, снизу на допустимом множестве). Если утверждать это можно, то задача неразрешима.

Если нельзя, то

3) по определенному правилу ищем новую, лучшую вершину и переходим к пункту 1).

Итак, для реализации предложенной схемы необходимо указать способ нахождения вершины допустимого множества, критерий оптимальности или неразрешимости, способ перехода от одной вершины к другой, лучшей.

## 1.5 Обоснование симплекс метода

Пусть имеется задача линейного программирования в канонической форме:

$$f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \to \max \tag{1.11}$$

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n = b_1 \\ ... \\ a_{m1}x_1 + .. + a_{mn}x_n = b_m \end{cases} \tag{1.12}$$

$$x_1 \geq 0, ..., x_m \geq 0$$

Вводя в рассмотрение векторы

$$\boldsymbol{A_j} = (\mathrm{a}_{1\mathrm{j}}, \mathrm{a}_{2\mathrm{j}}, ..., \mathrm{a}_{\mathrm{mj}})^{\mathrm{T}},\ \mathrm{j} = \overline{1, \mathrm{n}} \tag{1.13}$$

мы можем переписать задачу в форме

$$f = (c, x) \to \max$$

$$A_1x_1 + ... + A_nx_n = b \tag{1.14}$$

$$x \geq 0.$$

и трактовать ее следующим образом: из всех разложений вектора $b$ по векторам $A_1$, …, $A_n$, с неотрицательными коэффициентами требуется выбрать хотя бы одно такое, коэффициенты $x_i$, $i = 1,n$, которого доставляют целевой функции $f$ оптимальное значение. Не ограничивая общности считаем ранг матрицы $A$ равным $m$ и $n > m$ (случай $n = m$ тривиален). Дадим ряд определений.

**ОПРЕДЕЛЕНИЕ 1.5.** Ненулевое допустимое решение $x = (x_1, …, x_n)^{\mathrm{T}}$ называется опорным, если векторы $\boldsymbol{A}_{\mathrm{j}}$, соответствующие отличным от нуля координатам вектора $x$, линейно-независимы.

**ОПРЕДЕЛЕНИЕ 1.6.** Ненулевое опорное решение назовем невырожденным, если оно имеет точно «$m$» положительных координат.

**ОПРЕДЕЛЕНИЕ 1.7.** Если число положительных координат опорного решения меньше m, то оно называется вырожденным.

**ОПРЕДЕЛЕНИЕ 1.8.** Упорядоченный набор из «$m$» линейно − независимых векторов $A_{\mathrm{i}}$, соответствующих положительным координатам опорного решения назовем базисом.

**Пример 1.3.**

Дана система ограничений задачи:

$$\begin{cases} 2x_1 + x_2 + 3x_3 = 2 \\ x_1 - 2x_2 + x_4 = b1 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.$$

Здесь

$$m = 2,\ \mathrm{A}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix},\ \mathrm{A}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ \mathrm{A}_3 = \begin{pmatrix} 3 \\ -2 \end{pmatrix},\ \mathrm{A}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ \mathrm{b} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Очевидно, что

$x^{(1)} = (0,\ 2,\ 0,\ 1)^{\mathrm{T}}$, $x^{(2)} = (1,\ 0,\ 0,\ 0)^{\mathrm{T}}$, $x^{(3)} = (\frac{1}{2},\ 1,\ 0,\ \frac{1}{2})^{\mathrm{T}}$ являются допустимыми решениями задачи. Очевидно, что $x^{(1)}$, $x^{(2)}$– являются опорными решениями, поскольку системы векторов $\{A_2, A_4\}$ и $\{A_3\}$, линейно независимы, а $x^{(3)}$ не является опорным решением, так как $\{A_1,\ A_2,\ A_4\}$ – линейно зависимы. Опорное решение $x^{(1)}$– невырожденное, а $x^{(2)}$– вырожденное. Базис невырожденного опорного решения определяется естественным образом. Для $x^{(1)}$ – это $\{A_2, A_4\}$.

Приведем теорему, увязывающую понятие опорного решения и вершины допустимого множества.

**ТЕОРЕМА 1.1.** Вектор $\boldsymbol{x} = (x_1,\ldots,x_n)$ тогда и только тогда является опорным решением задачи, когда точка $x$ является вершиной допустимого множества.

*Доказательство. Необходимость.* Пусть $x$ – опорное решение задачи (1.19). Если $x = 0$, то невозможность ее представления в виде $x = \alpha x^{(1)} + \beta x^{(2)}$, где $\alpha > 0$, $\beta > 0$ и $x^{(2)} \neq x^{(2)}$ – допустимые решения, очевидна.

Пусть $x \neq 0$ и

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_k, \underbrace{0, \ldots, 0}_{n-k}),$$

где $x_1 > 0$, $x_2 > 0$, …, $x_k > 0$.

Предположим, в противоречие с теоремой, что точка $x$ не является вершиной допустимого многогранника:

$$\boldsymbol{x}^{(1)} = (x_1^{(1)},\ x_2^{(1)}, \ldots,\ x_n^{(1)}) \text{ и } \boldsymbol{x}^{(2)} = (x_1^{(2)},\ x_2^{(2)}, \ldots,\ x_n^{(2)}),$$

$$\boldsymbol{x} = \alpha\, x^{(1)} + \beta\, x^{(2)}, \alpha > 0, \beta > 0, \alpha + \beta = 1.$$

Так как последние $n - k$ координат точки $x$ равны нулю, то и последние $n - k$ координат, как точки $x^{(1)}$, так и точки $x^{(2)}$ должны быть равны 0.Таким образом, можно записать:

$$\boldsymbol{x}^{(1)} = (x_1^{(1)}, x_2^{(1)}, \ldots,\ x_k^{(1)}, 0, \ldots, 0),$$

$$\boldsymbol{x}^{(2)} = (x_1^{(2)}, x_2^{(2)}, \ldots,\ x_k^{(2)}, 0, \ldots, 0).$$

Ввиду допустимости точек $x^{(1)}$ и $x^{(2)}$ имеем:

$$\boldsymbol{b} = \boldsymbol{x}_1^{(1)}\boldsymbol{A}_1 + \boldsymbol{x}_2^{(1)}\boldsymbol{A}_2 + \ldots + \boldsymbol{x}_k^{(1)}\boldsymbol{A}_k \tag{1.15}$$

$$\boldsymbol{b} = \boldsymbol{x}_1^{(2)}\boldsymbol{A}_1 + \boldsymbol{x}_2^{(2)}\boldsymbol{A}_2 + \ldots + \boldsymbol{x}_k^{(2)}\boldsymbol{A}_k. \tag{1.16}$$

Вычитая почленно (1.15) из (1.16), получим:

$$(\boldsymbol{x}_1^{(1)} - \boldsymbol{x}_1^{(2)})\boldsymbol{A}_1 + (\boldsymbol{x}_2^{(1)} - \boldsymbol{x}_2^{(2)})\boldsymbol{A}_2 + \ldots + (\boldsymbol{x}_k^{(1)} - \boldsymbol{x}_k^{(2)})\boldsymbol{A}_k = 0 \tag{1.17}$$

Точки $x^{(1)}$ и $x^{(2)}$ различны, следовательно, среди коэффициентов при $A_i$ ($i = 1, 2, \ldots, k$) в (1.17) есть отличные от нуля. А это означает, что векторы $A_1, A_2, \ldots, A_k$ линейно зависимы, что противоречит тому, что $x = (x_1,\ x_2,\ \ldots,\ x_k,\ 0,\ \ldots,\ 0)$ – опорное решение задачи. Следовательно, наше предположение неверно и $x$ вершина допустимого многогранника.

*Достаточность.* Пусть наоборот, точка $x$– вершина допустимого многогранника. Если $x = 0$, то $x$ по условию опорное решение. Пусть, $x \neq 0$ и для определенности $x = (x_1, x_2, \ldots, x_k, 0, \ldots, 0)$, где $x_1 > 0, x_2 > 0, \ldots, x_k > 0$.

Предположим снова от противного, что при этом вектор $x = (x_1, x_2, \ldots, x_k, 0, \ldots, 0)$ являясь, конечно, допустимым для нашей задачи, не является опорным решением. Тогда векторы $A_1, A_2, \ldots, A_k$ – линейно зависимы, т.е. существуют такие действительные числа $\alpha_1, \alpha_2, \ldots, \alpha_k$ не все равные нулю, что справедливо равенство:

$$\alpha_1 \boldsymbol{A}_1 + \alpha_2 \boldsymbol{A}_2 + \ldots + \alpha_k \boldsymbol{A}_k = 0. \qquad (1.18)$$

Являясь допустимым, вектор x удовлетворяет условию:

$$\boldsymbol{x}_1 \boldsymbol{A}_1 + \boldsymbol{x}_2 \boldsymbol{A}_2 + \ldots + \boldsymbol{x}_k \boldsymbol{A}_k = \boldsymbol{b}. \qquad (1.19)$$

Умножим обе части равенства (1.18) на некоторый параметр $\theta$ и сначала вычтем почленно (1.18) из (1.19), а затем сложим (1.18) и (1.19).Получим:

$$(x_1 - \theta\alpha_1)\boldsymbol{A}_1 + (x_2 - \theta\alpha_2)\boldsymbol{A}_2 + \ldots + (x_k - \theta\alpha_k)\boldsymbol{A}_k = \boldsymbol{b}$$

$$(x_1 + \theta\alpha_1)\boldsymbol{A}_1 + (x_2 + \theta\alpha_2)\boldsymbol{A}_2 + \ldots + (x_k + \theta\alpha_k)\boldsymbol{A}_k = \boldsymbol{b}$$

Так как $x_i > 0$ ($i = 1, 2, \ldots, k$), то, очевидно, можно найти такое достаточно малое значение $\theta_0 > 0$ множителя $\theta$, что и в первом и во втором из полученных равенств, все коэффициенты будут неотрицательными. Тогда эти равенства (при $\theta = \theta_0$) означают, что $n$ – мерные векторы

$$x^{(1)} = (x_1 - \theta_0\alpha_1; x_2 - \theta_0\alpha_2; \ldots; x_k - \theta_0\alpha_k; 0; \ldots; 0)$$

и

$$x^{(1)} = (x_1 + \theta_0\alpha_1; x_2 + \theta_0\alpha_2; \ldots; x_k + \theta_0\alpha_k; 0; \ldots; 0)$$

являются допустимыми (причем различными) решениями задачи.

При этом, очевидно, имеем:

$$\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} = (x_1, x_2, \ldots, x_k, 0, \ldots, 0) = x,$$

а это противоречит тому, что $x$ – вершина допустимого многогранника. Следовательно, $x$ не может не быть опорным решением. Теорема доказана.

Таким образом, задача о нахождении вершины допустимого множества свелась к задаче нахождения опорного решения, а, следовательно, к нахождению базиса. Будем считать, что исходный базис $\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, \ldots, \boldsymbol{A}_{i_m}$ дан. Отправляясь от него покажем, как найти опорное решение. Сформулируем условие оптимальности решения, условие отсутствия решения. Покажем, как перейти к базису, дающему лучшее решение.

Дан базис

$\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, ..., \boldsymbol{A}_{i_m}$.

Тогда:

$$b = \boldsymbol{x}_{i_{10}} \boldsymbol{A}_{i_1} + \boldsymbol{x}_{i_{20}} \boldsymbol{A}_{i_2} + ... + \boldsymbol{x}_{i_{m0}} \boldsymbol{A}_{i_m} = [\boldsymbol{A}_{i_1}, ..., \boldsymbol{A}_{i_m}] \cdot (\boldsymbol{x}_{i_{10}}, ..., \boldsymbol{x}_{i_{m0}})^T,$$

$$(\boldsymbol{x}_{i_{10}}, ..., \boldsymbol{x}_{i_{m0}})^T = [\boldsymbol{A}_{i_1}, ..., \boldsymbol{A}_{i_m}]^{-1} \cdot (b_1, ..., b_m)^T$$

дополнив найденный вектор нулевыми значениями переменных $(x_{i_{m+1}}, ..., x_{i_n})$, получаем опорное решение. Исследуем его на оптимальность, для этого исследуем целевую функцию.

Предварительно введем обозначения:

$x_B = (x_{i_1}, ..., x_{i_m})^T$ – вектор базисных переменных;

$B = [\boldsymbol{A}_{i_1}, ..., \boldsymbol{A}_{i_m}]$ – матрица базисных векторов;

$x_D = (x_{i_{m+1}}, ..., x_{i_n})$ – вектор свободных переменных;

$c_B = (c_{i_1}, ..., c_{i_m})^T$ – вектор из коэффициентов целевой функции при базисных неизвестных;

$c_D = (c_{i_{m+1}}, ..., c_{i_n})^T$ – вектор из коэффициентов целевой функции при свободных переменных;

$D = [\boldsymbol{A}_{i_{m+1}}, ..., \boldsymbol{A}_{i_n}]$ – матрица свободных векторов (векторов, не вошедших в базис).

С учетом введенных обозначений ограничения (1.14) можно переписать в виде:

$$\boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{D}\boldsymbol{x}_D = \boldsymbol{b}. \qquad (1.20)$$

Выразим базисные переменные через свободные

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}(\boldsymbol{b} - \boldsymbol{D}\boldsymbol{x}_D) = \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{D}\boldsymbol{x}_D \qquad (1.21)$$

(при $x_D = 0$ получаем базисные компоненты опорного решения).

Подставим $x_B$ в целевую функцию:

$$f = \boldsymbol{c}_B^T \boldsymbol{x}_B + \boldsymbol{c}_D^T \boldsymbol{x}_D = \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{b} - (\boldsymbol{c}_B^T \boldsymbol{B}^{-1} D - \boldsymbol{c}_D^T) \boldsymbol{x}_D \qquad (1.22)$$

Обозначим:

$$\alpha = (\alpha_1, ..., \alpha_{n-m}) = (\boldsymbol{c}_B^T \boldsymbol{B}^{-1} D - \boldsymbol{c}_D^T) \qquad (1.23)$$

и назовем его вектором оценок.

Очевидно, что

$$\alpha \cdot x_c = \sum_{j=1}^{n-m} \alpha_j \cdot x_{i_{m+j}}. \qquad (1.24)$$

Отметим, что произведение $\boldsymbol{B}^{-1} A_j$ представляет разложение вектора $A_j$ по базису, следовательно, в столбцах матрицы $\boldsymbol{B}^{-1}$D стоят разложения векторов, не попавших в базис (свободных) по базису. Проанализируем выражение для целевой функции (1.22):

а) если все компоненты вектора оценок неотрицательные, то максимальное значение целевой функции достигается при $x_B = \boldsymbol{B}^{-1}\boldsymbol{b} x_B$ и $x_D = 0$, то есть построенное опорное решение – оптимальный план (решение) исходной задачи;

б) если среди компонент вектора оценок α есть отрицательные, то опорное решение не оптимально, поскольку, как видно из (1.22)−(1.24), возможно увеличить значение $f$ за счет некоторых компонент вектора $x_D$; при этом возможны две ситуации:

1) отрицательны компоненты вектора оценок, к примеру, $\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{ks}$ и при этом, по крайней мере у одного из свободных векторов $\boldsymbol{A}_{i_{m+k1}}, \boldsymbol{A}_{i_{m+k2}}, \ldots, \boldsymbol{A}_{i_{m+ks}}$ все компоненты разложения по базису не положительны, тогда из (1.21) видно, что можно неограниченно увеличить компоненты вектора $x_B$ за счет увеличения компоненты вектора $x_D$, а, следовательно, целевая функция не ограничена;

2) в случае положительных компонент разложения по базису свободных векторов существует лучшее опорное решение, которое можно получить, введя в базис свободный вектор $\boldsymbol{A}_s$, соответствующий наименьшей из отрицательных компонент вектора оценок (с целью максимально прирастить $f$) и, выведя из базиса тот вектор $\boldsymbol{A}_{i_r}$, исходного базиса, для которого:

$$\min_{x_{i_k s>0}} \frac{x_{i_k 0}}{x_{i_k s}} = \frac{x_{i_r 0}}{x_{i_r 0}} \tag{1.25}$$

где: $x_{\mathrm{i_k 0}}$ – компоненты $x_B$; $x_{\mathrm{i_k s}}$ – компоненты вектора $\boldsymbol{A}_\mathrm{S}$, вводимого в базис, $\mathrm{k} = \overline{1,m}$.

Заменив базис, мы можем вновь перейти к построению опорного решения и его исследования.

Отметим, что координаты всех векторов в новом базисе могут быть найдены через координаты векторов в старом базисе по формулам:

$$x'_{\mathrm{kj}} = \begin{cases} \dfrac{x_{\mathrm{kj}}}{x_{\mathrm{ks}}}, & \text{при } \mathrm{k} = \mathrm{r} \\ x_{\mathrm{kj}} - \dfrac{x_{\mathrm{rj}}}{x_{\mathrm{rs}}} x_{\mathrm{ks}}, & \mathrm{j} = \overline{0,\mathrm{n}},\ \mathrm{k} = \overline{1,\mathrm{m}},\ \mathrm{k} \neq \mathrm{r}. \end{cases} \tag{1.26}$$

В итоге алгоритм решения задачи линейного программирования можно изобразить схемой, представленной на рисунке 1.4.

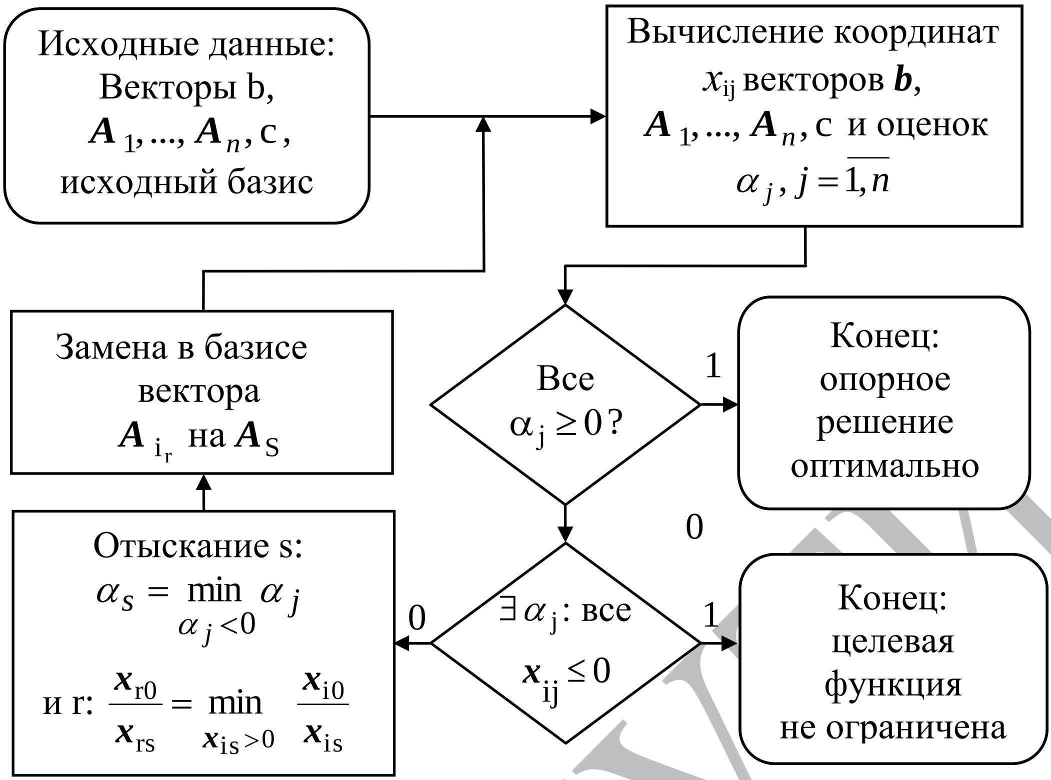


Рисунок 1.4 − Схема алгоритма решения задачи линейного программирования

## 1.6 Нахождение начального базиса

В заключении обратимся к вопросу построения начального базиса. Здесь отметим, что, если в исходной задаче все ограничения имели вид «≤», то в качестве начального базиса (как легко показать) можно брать базис из векторов при дополнительных переменных. В общем случае для построения начального базиса используется специальный прием, известный как метод искусственного базиса, изложенный ниже. Итак, пусть дана задача линейного программирования в канонической форме:

$$f = c \cdot x \to \max;$$

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b} \qquad (1.27)$$

$$\boldsymbol{x} \geq 0.$$

Рассмотрим вспомогательную задачу:

$$G = -y_1 - y_2 - ... - y_m \to \max$$

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n + y_1 = b_1 \\ a_{21}x_1 + ... + a_{2n}x_n + 0 \cdot y_1 + y_2 = b_2 \\ ... \\ a_{m1}x_1 + ... + a_{mn}x_n + 0 \cdot y_1 + ... + y_m = b_m \end{cases} \qquad (1.28)$$

$$x_i \geq 0,\ i = \overline{1,n},$$

$$y_j \geq 0,\ j = \overline{1,m}.$$

Так как целевая функция этой задачи ограничена сверху нулем, то задача имеет оптимальное решение.

Переменные $y_1, y_2, \ldots, y_m$ называются искусственными переменными.

Очевидно, что векторы

$\boldsymbol{A}_{n+1}=(1,0,\ldots,0)^T$, $\boldsymbol{A}_{n+2}=(0,1,0,\ldots,0)^T,\ldots$, $\boldsymbol{A}_{n+m}=(0,0,\ldots,1)^T$

образуют базис для опорного решения

$$x = (\underbrace{0, 0, ..., 0}_{n},\ b_1, b_2, ..., b_m),$$

который называют искусственным базисом. Решая вспомогательную задачу симплекс-методом, мы найдем оптимальное решение $x^{(0)}=(x_1^{(0)},\ldots, x_n^{(0)}, y_1^{(0)},\ldots, y_m^{(0)})$. Если в этом решении, среди искусственных переменных есть положительные, то исходная задача линейного программирования неразрешима, если же $y_i^{(0)} = 0,\ i = \overline{1,m}$, то базис, соответствующий оптимальному решению вспомогательной задачи, можно взять в качестве исходного базиса основной задачи.

## 1.7 Решение в форме симплекс − таблиц

В случае небольшого числа ограничений и переменных, приведенный в предыдущем разделе алгоритм можно легко реализовать без помощи ЭВМ, оформляя решение в виде симплекс-таблиц. Итак, дана задача:

$$f = (\boldsymbol{c}, \boldsymbol{x}) \to \max$$
$$\boldsymbol{A}_1 x_1 + \boldsymbol{A}_2 x_2 ... + \boldsymbol{A}_n x_n = \boldsymbol{b}, \qquad (1.29)$$
$$\boldsymbol{x} \geq 0.$$

и найден исходный базис $\boldsymbol{A}_{i_1}, \boldsymbol{A}_{i_2}, ..., \boldsymbol{A}_{i_m}$. Находим разложения всех векторов $\boldsymbol{A}_1, ..., \boldsymbol{A}_n$ и $\boldsymbol{b}$ по базису. Полученные результаты заносим в таблицу 1.2.

Таблица 1.2 - Симплекс таблица

| № п/п | Базис | $\boldsymbol{c}_{B.}$ | $\boldsymbol{b}$ (опорное решение) | $c_1$ | $c_2$ | … | $c_n$ |
|---|---|---|---|---|---|---|---|
| | | | | $\boldsymbol{A}_1$ | $\boldsymbol{A}_2$ | … | $\boldsymbol{A}_n$ |
| 1 | $\boldsymbol{A}_{i_1}$ | $c_{i_1}$ | $x_{10}$ | $x_{11}$ | $x_{12}$ | … | $x_{1n}$ |
| 2 | $\boldsymbol{A}_{i_2}$ | $c_{i_2}$ | $x_{20}$ | $x_{21}$ | $x_{22}$ | … | $x_{2n}$ |
| 3 | $\boldsymbol{A}_{i_3}$ | $c_{i_3}$ | $x_{30}$ | $x_{31}$ | $x_{32}$ | … | $x_{3n}$ |
| … | … | … | … | … | … | … | … |
| m | $\boldsymbol{A}_{i_m}$ | $c_{i_m}$ | $x_{m0}$ | $x_{m1}$ | $x_{m2}$ | … | $x_{mn}$ |
| Оценки | | $f = \sum_{k=1}^{m} c_{i_k} x_{k0}$ | | $\alpha_1$ | $\alpha_2$ | … | $\alpha_n$ |

Здесь в столбце «Базис» указываются вектора $\boldsymbol{A}_{i_k}$ попавшие в базис; в столбце $\boldsymbol{c}_{B.}$ записываются коэффициенты $c_{i_k}$ при соответствующих неизвестных из целевой функции; в столбце $\boldsymbol{b}$ (опорное решение) записываются координаты разложения вектора $\boldsymbol{b}$ по базису, то есть опорный план; в столбцах $\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_n$ записывают координаты разложения этих векторов по базису, то есть столбцы матрицы $\boldsymbol{B}^{-1}\boldsymbol{D}$. После этого заполняют строку «Оценки»:

$f = \sum_{k=1}^{m} c_{i_k} x_{k0}$ − значение целевой функции для найденного опорного решения и оценки;

$\alpha_j = z_j - c_j = \sum_{k=1}^{m} c_{i_k} \boldsymbol{x}_{kj} - c_j$ – компоненты вектора;

$\boldsymbol{\alpha} = \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_j - c_j$.

Далее, согласно алгоритму, выясняем, будет ли найденное опорное решение оптимальным, если да, то это решение (план) в столбце «$\boldsymbol{b}$», а если нет, то выбираем вектор подлежащий вводу в базис, вектор подлежащий выводу из базиса и переходим к нахождению следующего опорного решения, заполняя новую таблицу.

**Пример 1.4.**

$$f(x) = 5x_1 + 10x_2 \to \max$$

$$\begin{cases} 4x_1 + 3x_2 \le 96 \\ 5x_1 + 8x_2 \le 144 \\ x_1 + 4x_2 \le 48 \end{cases}$$

$x_1 \ge 0, x_2 \ge 0.$

Приведем задачу к каноническому виду:

$$f(x) = 5x_1 + 10x_2 + 0x_3 + 0x_4 + 0x_5 \to \max$$

$$\begin{cases} 4x_1 + 3x_2 + x_3 = 96 \\ 5x_1 + 8x_2 + x_4 = 144 \\ x_1 + 4x_2 + x_5 = 48 \end{cases}$$

$x_i \ge 0, \mathrm{i} = \overline{1,5}.$

Введя векторы:

$$\boldsymbol{b} = \begin{pmatrix} 96 \\ 144 \\ 48 \end{pmatrix}, \ \boldsymbol{A}_1 = \begin{pmatrix} 4 \\ 5 \\ 1 \end{pmatrix}, \ \boldsymbol{A}_2 = \begin{pmatrix} 3 \\ 8 \\ 4 \end{pmatrix}, \ \boldsymbol{A}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \boldsymbol{A}_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \boldsymbol{A}_5 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

можем записать задачу в виде:

$$f(x) = 5x_1 + 10x_2 + 0x_3 + 0x_4 + 0x_5 \to \max$$

$$x_1\boldsymbol{A}_1 + x_2\boldsymbol{A}_2 + x_3\boldsymbol{A}_3 + x_4\boldsymbol{A}_4 + x_5\boldsymbol{A}_5 = \boldsymbol{b}$$

$x_i \ge 0, \mathrm{i} = \overline{1,5}.$

Очевидно, что векторы $\boldsymbol{A}_3, \boldsymbol{A}_4, \boldsymbol{A}_5$ образуют базис и столь же очевидно, что относительно этого базиса:

$$\boldsymbol{b} = 96\boldsymbol{A}_3 + 144\boldsymbol{A}_4 + 48\boldsymbol{A}_5$$

$$\boldsymbol{A}_1 = 4\boldsymbol{A}_3 + 5\boldsymbol{A}_4 + 1\boldsymbol{A}_5$$

$$\boldsymbol{A}_2 = 3\boldsymbol{A}_3 + 8\boldsymbol{A}_4 + 4\boldsymbol{A}_5$$

$$A_3 = 1A_3 + 0A_4 + 0A_5$$
$$A_4 = 0A_3 + 1A_4 + 0A_5$$
$$A_5 = 0A_3 + 0A_4 + 1A_5$$

Заполняем первую симплекс таблицу 1.3:

Таблица 1.3 – Симплекс таблица

| № п/п | Базис | $\boldsymbol{c}_B$ | $\boldsymbol{b}$ (опорное решение) | 5 | 10 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | $\boldsymbol{A}_1$ | $\boldsymbol{A}_2$ | $\boldsymbol{A}_3$ | $\boldsymbol{A}_4$ | $\boldsymbol{A}_5$ |
| 1 | $\boldsymbol{A}_3$ | 0 | 96 | 4 | 3 | 1 | 0 | 0 |
| 2 | $\boldsymbol{A}_4$ | 0 | 144 | 5 | 8 | 0 | 1 | 0 |
| 3 | $\boldsymbol{A}_5$ | 0 | 48 | 1 | $\boxed{4}$ | 0 | 0 | 1 |
| Оценки | | | $f = 0$ | $\alpha_1 = -5$ | $\alpha_2 = -10$ | $\alpha_3 = 0$ | $\alpha_4 = 0$ | $\alpha_5 = 0$ |

Поскольку среди $\alpha_j$ есть отрицательные, то план не оптимален. Находим $\min\limits_{j=1,5} \alpha_j = \alpha_2 = -10$, следовательно, в новый базис будем вводить вектор $\boldsymbol{A}_2$ и столбец $\boldsymbol{A}_2$ назовем *ведущим*.

Найдем отношение координат вектора $\boldsymbol{b}$ к соответствующим, но только положительным координатам вектора $\boldsymbol{A}_2$ и выберем минимальное из них:

$$\min\left(\frac{96}{3}, \frac{144}{8}, \frac{48}{4}\right) = \frac{48}{4} = 12.$$

Из этого следует, что из базиса надо вывести $\boldsymbol{A}_5$. Строку «$\boldsymbol{A}_5$» будем называть *ведущей*. Число **4**, стоящее на пересечении ведущей строки и ведущего столбца, будем называть *ведущим элементом*.

Переходим к нахождению нового опорного решения, для чего составляем новую симплекс таблицу с новым базисом $\boldsymbol{A}_3, \boldsymbol{A}_4, \boldsymbol{A}_2$. Заполнение новой симплекс таблицы начинаем с заполнения строки, соответствующей вновь вводимому вектору $\boldsymbol{A}_2$. Она получается делением элементов ведущей строки на ведущий элемент. Остальные элементы таблицы можно найти исходя из формулы

$$x'_{ij} = x_{ij} - \frac{x_{rj}}{x_{rs}} x_{is}, \tag{1.30}$$

где: $x'_{ij}$ – искомый элемент в новой таблице;

$x_{rs}$ – ведущий элемент;

$x_{rj}$ – элемент, стоящий в проекции элемента $x_{ij}$ на ведущий столбец;

$x_{is}$ – элемент, стоящий в проекции $x_{ij}$ на ведущую строку.

Таблица 1.4 – Вторая симплекс таблица

| № п/п | Базис | $\boldsymbol{c}_B$ | $\boldsymbol{b}$ (опорное решение) | 5 | 10 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | $\boldsymbol{A}_1$ | $\boldsymbol{A}_2$ | $\boldsymbol{A}_3$ | $\boldsymbol{A}_4$ | $\boldsymbol{A}_5$ |
| 1 | $\boldsymbol{A}_3$ | 0 | 60 | 13/14 | 0 | 1 | 0 | -3/4 |
| 2 | $\boldsymbol{A}_4$ | 0 | 48 | $\boxed{3}$ | 0 | 0 | 1 | -2 |
| 3 | $\boldsymbol{A}_2$ | 10 | 12 | 1/4 | 1 | 0 | 0 | 1/4 |
| Оценки | | | $f = 120$ | $\alpha_1 = -\frac{5}{2}$ | $\alpha_2 = 0$ | $\alpha_3 = 0$ | $\alpha_4 = 0$ | $\alpha_5 = \frac{5}{2}$ |

Так как min $\alpha_j = \alpha_1 = -\frac{5}{2} < 0$, то план не оптимален и в базис следует ввести $\boldsymbol{A}_1$.

Найдем:

$$\min\left(\frac{60}{13/4}, \frac{48}{3}, \frac{12}{1/4}\right) = \frac{48}{3} = 16,$$

то есть из базиса выводим $\boldsymbol{A}_4$. Составляем симплекс таблицу (таблица 1.5).

Таблица 1.5 – Третья симплекс таблица

| № п/п | Базис | $\boldsymbol{c}_B$ | $\boldsymbol{b}$ (опорное решение) | 5 | 10 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | $\boldsymbol{A}_1$ | $\boldsymbol{A}_2$ | $\boldsymbol{A}_3$ | $\boldsymbol{A}_4$ | $\boldsymbol{A}_5$ |
| 1 | $\boldsymbol{A}_3$ | 0 | 8 | 0 | 0 | 1 | -13/12 | 17/12 |
| 2 | $\boldsymbol{A}_1$ | 5 | 16 | 1 | 0 | 0 | 1/3 | -2/3 |
| 3 | $\boldsymbol{A}_2$ | 10 | 8 | 0 | 1 | 0 | -1/12 | 5/12 |
| Оценки | | | $f = 160$ | $\alpha_1 = 0$ | $\alpha_2 = 0$ | $\alpha_3 = 0$ | $\alpha_4 = \frac{5}{2}$ | $\alpha_5 = \frac{5}{2}$ |

Так как все $\alpha_j \geq 0$, то найденное опорное решение

$x_1 = 16$, $x_2 = 8$, $x_3 = 8$, $x_4 = 0$, $x_5 = 0$ доставляет целевой функции $f$ максимальное значение

$f_{max} = 5 \cdot 16 + 10 \cdot 8 = 160$.

# 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Задание на практическое занятие

### Тема практического занятия «Решение задачи линейного программирования»

Текст задания:

Найти максимальное и минимальное значение функции f(x) при ограничениях.

Способы решения:

1. Решение задачи линейного программирования геометрическим способом.

*Комментарии по решению:* построить графическое представление решения задачи в редакторе MS Excel или Mathcad.

2. Решение задачи линейного программирования с помощью симплекс-таблицы.

*Комментарии по решению:* построить симплекс-таблицы в редакторе MS Excel.

3. Решение задачи линейного программирования с помощью программы simplecs.

*Комментарии по решению:* программу реализовывать в среде Free Pascal или как проект консольного приложения в среде Lazarus.

## 2.2 Варианты заданий

Вариант заданий выдается преподавателем.

Таблица 1.6 – Варианты заданий

| Номер варианта | Задание | Номер варианта | Задание |
|---|---|---|---|
| 1 | $f(x) = x_1 + x_2 \to \max(\min)$ <br> $\begin{cases} x_1 + 2x_2 \le 14 \\ 2x_1 + x_2 \le 10 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ | 2 | $f(x) = 2x_1 + 3x_2 \to \max(\min)$ <br> $\begin{cases} -4x_1 + 2x_2 \le 4 \\ x_1 + x_2 \le 6 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ |

| | | | |
|---|---|---|---|
| 3 | $f(x)=2x_1+x_2\to\max(\min)$ $\begin{cases}4x_1+x_2\le 16\\ x_1+x_2\le 11\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 4 | $f(x)=2x_1+x_2\to\max(\min)$ $\begin{cases}x_1+x_2\le 3\\ x_1-2x_2\le 1\end{cases}$ $x_1\ge 0, x_2\ge 0$ |
| 5 | $f(x)=x_1+3x_2\to\max(\min)$ $\begin{cases}2x_1+x_2\le 6\\ -x_1-3x_2\le 6\\ -x_1+2x_2\le 8\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 6 | $f(x)=2x_1+3x_2\to\max(\min)$ $\begin{cases}2x_1+x_2\le 10\\ -2x_1+3x_2\le 6\\ x_1+x_2\le 8\end{cases}$ $x_1\ge 0, x_2\ge 0$ |
| 7 | $f(x)=x_1+4x_2\to\max(\min)$ $\begin{cases}-x_1+5x_2\le 20\\ 3x_1-x_2\le 15\\ x_1+x_2\le 6\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 8 | $f(x)=6x_1+2x_2\to\max(\min)$ $\begin{cases}x_1-x_2\le 4\\ -x_1+3x_2\le 6\\ x_1-x_2\le 3\end{cases}$ $x_1\ge 0, x_2\ge 0$ |
| 9 | $f(x)=5x_1+7x_2\to\max(\min)$ $\begin{cases}5x_1-6x_2\le 30\\ -3x_1+14x_2\le 42\\ x_1+4x_2\le 28\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 10 | $f(x)=x_1+2x_2\to\max(\min)$ $\begin{cases}x_1+3x_2\le 6\\ -x_1+x_2\le 1\\ 3x_1-x_2\le 6\end{cases}$ $x_1\ge 0, x_2\ge 0$ |
| 11 | $f(x)=3x_1-2x_2\to\max(\min)$ $\begin{cases}x_1+2x_2\le 8\\ -2x_1+x_2\le 2\\ x_1-x_2\le 12\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 12 | $f(x)=2x_1+3x_2\to\max(\min)$ $\begin{cases}x_1+x_2\le 10\\ -2x_1+3x_2\le 6\\ x_1-x_2\le 4\end{cases}$ $x_1\ge 0, x_2\ge 0$ |
| 13 | $f(x)=x_1+6x_2\to\max(\min)$ $\begin{cases}x_1+x_2\le 10\\ x_1\le 8\\ -x_1+x_2\le 3\end{cases}$ $x_1\ge 0, x_2\ge 0$ | 14 | $f(x)=x_1+2x_2\to\max(\min)$ $\begin{cases}2x_1-x_2\le 6\\ -x_1+3x_2\le 6\\ x_1+x_2\le 4\end{cases}$ $x_1\ge 0, x_2\ge 0$ |

| | | | |
|---|---|---|---|
| 15 | $f(x) = 6x_1 + x_2 \to \max(\min)$ $\begin{cases} x_1 + x_2 \le 20 \\ -x_1 + x_2 \le 15 \\ x_1 - 3x_2 \le 9 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 15 | $f(x) = x_1 + 4x_2 \to \max(\min)$ $\begin{cases} x_1 - 3x_2 \le 6 \\ x_1 + x_2 \le 9 \\ -x_1 + x_2 \le 4 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |
| 17 | $f(x) = x_1 + x_2 \to \max(\min)$ $\begin{cases} 3x_1 + 4x_2 \le 12 \\ 4x_1 - x_2 \le 8 \\ 2x_2 \le 12 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 18 | $f(x) = 2x_1 + x_2 \to \max(\min)$ $\begin{cases} x_1 + x_2 \le 10 \\ -2x_1 + x_2 \le 2 \\ x_1 - 8x_2 \le 8 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |
| 19 | $f(x) = x_1 + x_2 \to \max(\min)$ $\begin{cases} x_1 + 2x_2 \le 14 \\ -5x_1 - 3x_2 \le 15 \\ 2x_1 - 3x_2 \le 12 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 20 | $f(x) = 5x_1 + 4x_2 \to \max(\min)$ $\begin{cases} x_1 - 2x_2 \le 6 \\ -x_1 + x_2 \le 8 \\ x_1 - x_2 \le 10 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |
| 21 | $f(x) = 3x_1 + 2x_2 \to \max(\min)$ $\begin{cases} x_1 + x_2 \le 6 \\ x_1 - 2x_2 \le 4 \\ x_1 - 3x_2 \le 3 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 22 | $f(x) = 5x_1 + 4x_2 \to \max(\min)$ $\begin{cases} x_1 + x_2 \le 18 \\ 5x_1 - x_2 \le 20 \\ x_1 - x_2 \le 8 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |
| 23 | $f(x) = 3x_1 + 2x_2 \to \max(\min)$ $\begin{cases} 2x_1 + x_2 \le 8 \\ x_1 + 3x_2 \le 6 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 24 | $f(x) = x_1 + 4x_2 \to \max(\min)$ $\begin{cases} 2x_1 + x_2 \le 6 \\ x_1 + 3x_2 \le 9 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |
| 25 | $f(x) = x_1 + x_2 \to \max(\min)$ $\begin{cases} 3x_1 + 4x_2 \le 12 \\ 2x_1 - x_2 \le 6 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ | 26 | $f(x) = 2x_1 + 3x_2 \to \max(\min)$ $\begin{cases} x_1 + 4x_2 \le 12 \\ x_1 + x_2 \le 4 \end{cases}$ $x_1 \ge 0, x_2 \ge 0$ |

| | | | |
|---|---|---|---|
| 27 | $f(x)=x_1+2x_2 \to \max(\min)$ <br> $\begin{cases} 5x_1+4x_2 \le 20 \\ 3x_1-x_2 \le 6 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ | 28 | $f(x)=x_1+x_2 \to \max(\min)$ <br> $\begin{cases} x_1+2x_2 \le 8 \\ 6x_1-x_2 \le 3 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ |
| 29 | $f(x)=x_1+5x_2 \to \max(\min)$ <br> $\begin{cases} 2x_1+x_2 \le 24 \\ -x_1+x_2 \le 12 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ | 30 | $f(x)=x_1+x_2 \to \max(\min)$ <br> $\begin{cases} x_1+2x_2 \le 14 \\ -5x_1+3x_2 \le 15 \end{cases}$ <br> $x_1 \ge 0, x_2 \ge 0$ |

### 2.3 Исходный код программы

*Входные данные:*

- N – число переменных;
- M – число ограничений;
- eps – точность;
- ip – признак вида задачи (если на максимум ,то ip = 1,если на минимум, то ip = 0);
- T[i] – коэффициенты целевой функции;
- S – массив из M–2 чисел, содержащий правые части системы;
- R – массив из (M–2)·N чисел, содержащий коэффициенты при неизвестных в системе ограничений.

*Выходные данные:*

- ip – признак окончания решения(ip = 1 – найдено оптимальное решение; ip = 2 – задача не имеет решения; ip = 3 – целевая функция не ограничена);
- nb – массив, содержащий номера переменных в массиве x;
- x – массив из M чисел, содержащий оптимальное решение;
- f – оптимальное значение целевой функции.

```
program  simplecs;
type mas=array[1..100] of real;
    mas1=array[1..100] of integer;
    var
       r,s,t,a,u,x,xk:mas;
       nb:mas1;
      i,l,k,z1,mi,m1,ni,ne,ip,m,n:integer;
       eps,tmin,teta:real;
procedure sol00(r,s,t:mas;var a,x:mas; n,m:integer);
  var k1,k2,k3,j,mj,l,i:integer;
  BEGIN
     l:=m-2;
  for j:=1 to n do
  begin
```

```
  mj:=m*j;
  a[mj]:=0;
  for i:=1 to l do
  begin
  k1:=m*(j-1)+i;
  k2:=l*(j-1)+i;
  a[k1]:=r[k2];
  a[mj]:=a[mj]-r[k2];
  end;
  end;
  for i:=1 to n do
  begin
  k3:=m*i-1;
  a[k3]:=t[i];
  end;
  x[m-1]:=0;
  x[m]:=0;
  for i:=1 to l do
  begin
  x[i]:=s[i];
  x[m]:=x[m]-x[i];
end;END;
 procedure sol01(var u:mas;m:integer);
 var i,j,l:integer;
   BEGIN
   for j:=1 to m do
   for i:=1 to m do
   begin
     l:=m*(j-1)+i;
     u[l]:=0;
     if (i-j)=0 then
     u[l]:=1;
     end;
     END;
  procedure sol02(u,a:mas;m,n,j:integer;var del:real);
  var i,im,ij:integer;
  begin
  del:=0;
  for i:=1 to m do
  begin
  im:=i*m;
  ij:=m*(j-1)+i;
  del:=del+u[im]*a[ij];
  end;
  EnD;
```

```
procedure sol03(var tmin:real;a,u:mas;nb:mas1;m,n:integer;var k:integer);
            var  bul,i,j,m1:integer;
            del:real;
   begin
   tmin:=0;
   m1:=m-2;
   for j:=1 to n do
   begin
  bul:=1;
  i:=1;
  while (bul=1) and (i<=m1) do
  if (j-nb[i])=0  then bul:=2
  else
  i:=i+1;
  if bul<>2 then
  begin
  sol02(u,a,m,n,j,del);
  if (del-tmin)<=0 then
  begin
  tmin:=del;
  k:=j;
  end;
  end;
  end;
  end;
  procedure sol04(u,a:mas;m,n,k:integer;var xk:mas);
  var
     ij,jk,i,j:integer;
    begin
   for i:=1 to m do
   begin
   xk[i]:=0;
   for j:=1 to m do
   begin
   ij:=m*(j-1)+i;
   jk:=m*(k-1)+j;
   xk[i]:=xk[i]+u[ij]*a[jk];
   end;
   end;
   end;
procedure sol05(x,xk:mas;m:integer;var l:integer;var teta:real;eps:real);
var
  i,m1:integer;
  r:real;
 begin
```

```
  teta:=10000;
  m1:=m-2;
   for i:=1 to m1 do
   if(xk[i]-eps)>=0 then
   begin
   r:=x[i]/xk[i];
  if (r-teta)<=0 then
  begin
  teta:=r;
  l:=i;
  end;
  end;
  end;
procedure sol06(var x,xk:mas;m,l:integer;var teta:real);
var i:integer;
begin
  for i:=1 to m do begin
  If (i-l)<>0 then
  x[i]:=x[i]-teta*xk[i]
  else
  x[i]:=teta;
  end;
  end;
procedure sol07(var u:mas;m,l:integer;xk:mas);
var
      m1,j,lj,i,ij:integer;
     begin
  m1:=m-2;
  for j:=1 to m1 do
  begin
  lj:=m*(j-1)+l;
  u[lj]:=u[lj]/xk[l];
  end;
  for i:=1 to m do
  for j:=1 to m1 do
  if (i-l)<>0 then
  begin
  ij:=m*(j-1)+i;
  lj:=m*(j-1)+l;
  u[ij]:=u[ij]-u[lj]*xk[i];
  end;
  end;
procedure sol08(u,a:mas;m,j:integer;var del:real);
var mi,ij,i:integer;
begin
```

```
 del:=0;
   for i:=1 to m do
   begin
   mi:=m*i-1;
   ij:=m*(j-1)+i;
   del:=del+u[mi]*a[ij];
   end;
   end ;
procedure sol09(var tmin:real;var k:integer;a,u:mas;nb:mas1;m,n:integer);
var bul,m1,i,j:integer;
    del:real;
  begin
  tmin:=0;
  m1:=m-2;
  for j:= 1 to n do
  begin
  bul:=1;
  i:=1;
  while (bul=1)  and (i<=m1) do
  if (j-nb[i])=0 then
  bul:=2
  else
  i:=i+1;
  if bul<>2 then
  begin
  sol08(u,a,m,j,del);
  if (del-tmin)<=0 then
  begin
  tmin:=del;
  k:=j;
  end;
  end;
  end;
  end;
procedure sol10(var tmin:real;u,a:mas;nb:mas1;m,n:integer;var k:integer;
eps:real);
 var bul,m1,i,j:integer;
   del,del1:real;
   begin
      tmin:=0;
      m1:=m-2;
   for j:=1 to n do
     begin
       bul:=1;
       i:=1;
```

```
    while (bul=1) and (i<=m1) do
        if (j-nb[i])=0 then
          bul:=2
          else
    i:=i+1;
    if bul<>2 then
    begin
    sol02(u,a,m,n,j,del);
    sol08(u,a,m,j,del1);
    if (abs(del1)-eps)<=0 then
    if (del-tmin)<=0 then
    begin
    tmin:=del;
    k:=j;
    end;
    end;
    end;
    end;
BEGIN
  write(‘ n’);
  read(n);
  writeln(‘m’);readln(m);
  writeln(' (eps)=>');read(eps);
  writeln('ip(ip=1 if MAKS;ip=0 if MIN )=>');
  read(ip);
  for i:=1 to n do
  begin
    writeln('t[',i, ']= ');
    read(t[i]);
    end;
  for i:=1 to m-2 do
  begin
  writeln('s[',i, ']= ');
  read(s[i]);
  end;
  for i:=1 to (m-2)*n do
  begin
   writeln('r[',i, ']= ' );
  read(r[i]);end;
  sol00(r,s,t,a,x,n,m);
  if (ip-1)=0 then
  for i:=1 to n do
  begin
   mi:=m*i-1;
   a[mi]:=-a[mi];
```

```
 end;
   sol01(u,m);
   m1:=m-2;
   for i:=1 to m1 do
   nb[i]:=100011+i;
   ni:=0;
   ne:=1;
3: sol03(tmin,a,u,nb,m,n,k);
2:   if (tmin+eps)>=0 then
   if (ne)=1 then
   if (x[m]+eps)>=0 then
   begin
   ne:=2;
   for i:=1 to m1 do
   if (nb[i]-10000)>0 then
   ne:=3 ;
   if ne=3 then
   begin
   sol10(tmin,u,a,nb,m,n,k,eps);
    goto 2;
    end  else
    begin
    sol09(tmin,k,a,u,nb,m,n);
    goto 2;
    end;end else
    begin
    ip:=2;
    goto 10;
    end
    else
    if (ip-1)<>0 then
    begin
    x[m-1]:=-x[m-1];
    ip:=1;
    goto 10;
    end
    else
    begin
    ip:=1;
    goto 10;
    end
    else
    begin
    sol04(u,a,m,n,k,xk);
    sol05(x,xk,m,l,teta,eps);
```

```
    if (teta+5-10000)<0 then
    begin
    sol06(x,xk,m,l,teta);
    sol07(u,m,l,xk);
    nb[l]:=k;
    ni:=ni+1;
    if ne<>1 then
    if ne=2 then
    begin
    sol09(tmin,k,a,u,nb,m,n);
    goto 2;
    end
    else
    begin
    sol10(tmin,u,a,nb,m,n,k,eps);
    goto 2;
    end
    else
    goto 3 ;end
     else
     begin
     ip:=3;
     goto 10;
     end; end;
10:  writeln('ip=',ip);
 if ip=1 then begin
    for i:=1 to m-2 do
        writeln('x[', nb[i], ']=',x[i]:13);
        for i:=1  to m-2  do
        writeln('nb[',i,']=',nb[i]);end;
        writeln('f=', x[m-1]:13);
        readln; readln;readln;
END.
```

## 2.4 Пример решения задачи с помощью программы simplecs

**Пример**. Найти максимальное значение целевой функции $f(\mathbf{x}) = 10x_1 - x_2 - 9x_3 - 8x_4$ при следующих ограничениях:

$$\begin{cases} -2x_1 + x_2 + 3x_3 + x_4 = 2 \\ -5x_1 + 2x_2 + 3x_4 = 5 \\ 7x_1 - 4x_2 + x_3 + 4x_4 \le 1 \\ 3x_1 + 2x_2 + 5x_3 + 6x_4 \le 10 \end{cases}$$

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

*Входные данные:*
N = 6
M = 6
EPS = 0.1e–6
ip = 1

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | |
|---|---|---|---|---|---|---|---|
| коэффициенты целевой функции | 10 | –1 | –9 | –8 | 0 | 0 | Вводить по строкам |
| свободные члены ограничений | 2 | 5 | 1 | 10 | | | |
| коэффициенты 1-го ограничения | –2 | 1 | 3 | 1 | 0 | 0 | Вводить по столбцам |
| коэффициенты 2-го ограничения | –5 | 2 | 0 | 3 | 0 | 0 | |
| коэффициенты 3-го ограничения | 7 | –4 | 1 | 4 | 1 | 0 | |
| коэффициенты 4-го ограничения | 3 | 2 | 5 | 6 | 0 | 1 | |

*Выходные данные:*
ip = 1 (найдено оптимальное решение)
x(1) = 1.428571E-01
x(2) = 1.142857E+00
x(4) = 1.142857E+00
x(6) = 4.285741E-01
$f$ = –8.857143E+00
Ответ: $x_1$ = 0.142, $x_2$ = 1.142, $x_3$ = 0, $x_4$ = 1.142, $f_{макс}$ = –8.857.


…

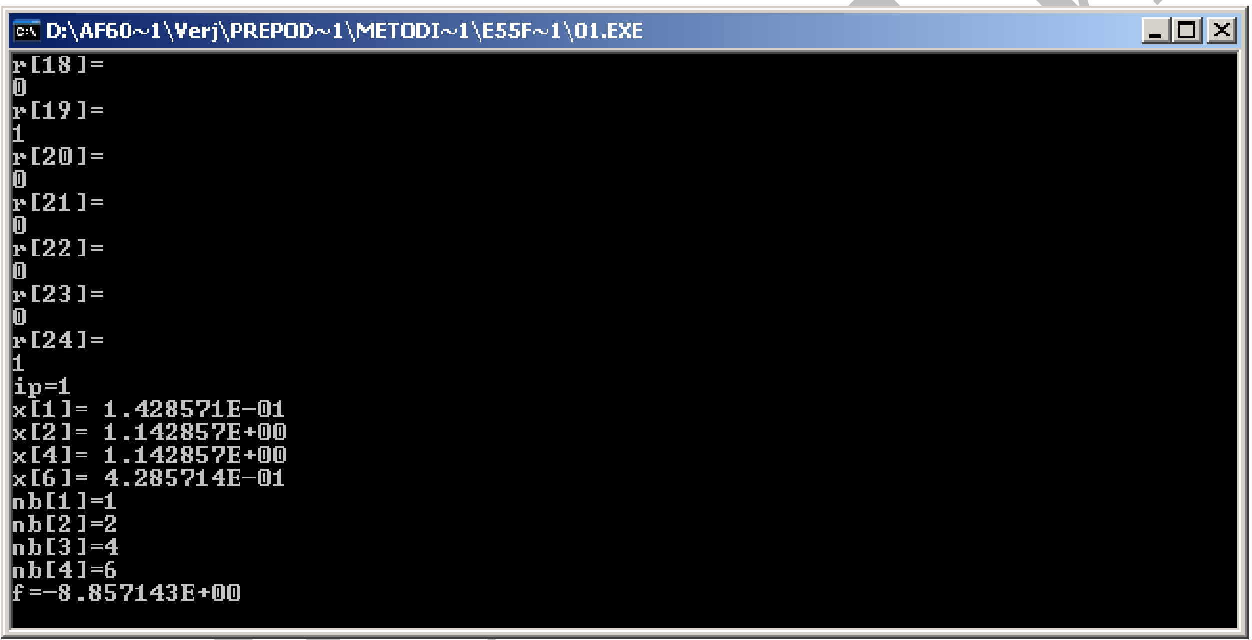


Рисунок 1.5 – Решение задачи линейного программирования с помощью программы simplecs

### 2.5 Вопросы для самоконтроля

1. Дайте определение задачи линейного программирования общего вида.
2. Что такое допустимый план задачи?
3. Сформулируйте определение оптимального решения задачи.
4. Напишите каноническую задачу линейного программирования.
5. Опишите геометрическую интерпретацию задачи линейного программирования.
6. Сформулируйте основные свойства задачи линейного программирования.
7. Дайте определение опорному решению и базису.

8. Что такое невырожденное и вырожденное опорное решение? Чем они отличается друг от друга?

## 2.6 Требования к оформлению

Лабораторная работа оформляется в текстовом процессоре MS Word в печатном виде.

Лабораторная работа выполняется на листах формата А4, которые должны быть сброшюрованы в виде папки.

**Параметры страницы.** Страницы должны иметь следующие поля: слева – 25 мм, справа – 20 мм, сверху и снизу – по 20 мм. Все страницы, за исключением *титульного листа* (номер 1), нумеруются сверху в центре в следующем порядке: страница 2 – *замечания руководителя* по пояснительной записке КР (оставляется пустой), страница 3 – *оглавление* с обязательным указанием номеров страниц разделов, страница 4 и т.д. – *содержательная часть КР*. В контрольной работы приводится *список источников информации*.

**Форматирование текста.** Текст лабороторной работы располагается на одной стороне листа. Стиль абзаца: шрифт Times New Roman размером в 14 пунктов, отступ первой строки 1,27 см, межстрочный интервал – одинарный, выравнивание по ширине, запрет висячих строк. Предусмотреть автоматический перенос слов и проверку орфографии.

Заголовки выполняются стилями: Заголовок 1, Заголовок 2, Заголовок 3 и т.д., учитывая, что размер шрифта каждого стиля должен отличаться от предыдущего на 1 или 2 пункта. Заголовок самого нижнего уровня иерархии должен иметь размер шрифта не менее размера шрифта абзаца. В каждом стиле заголовка должна быть выключена опция « Расстановка переносов».

**Оформление рисунков.** Рисунки выполняются в удобных для пользователя редакторах и импортируются в текст лабороторной работы. Все рисунки должны быть пронумерованы и содержать наименование и, при необходимости, пояснительный текст.

Форматирование рисунков необходимо производить так, чтобы все детали рисунка были различимы при печати, а надписи на рисунках имели размер шрифта примерно на 1 пункт меньше шрифта стиля абзаца.

Рисунки отбивают от текста сверху в пределах 1,5 кегельных, снизу – 3 кегельных (кегль – размер шрифта). Если подпись к рисунку располагается под ним, то ее отбивка от рисунка должна быть меньше, чем от следующего текста.

Общая высота рисунка с подписью и отбивками должна быть кратна кеглю основного шрифта, а для рисунков в оборку – кеглю шрифта, которым делается оборка.

**Оформление таблиц.** Основное поле таблицы заполняется шрифтом на 1 пункт меньше шрифта стиля абзаца; головки таблицы – заполняются шрифтом на 2 пункта меньше размера шрифта основного поля таблицы. Все таблицы должны быть пронумерованы и содержать наименование и при необходимости пояснительный текст.

**Оформление формул.** Набор формул производится с помощью формульного редактора Microsoft Equation, с установками форматирования «по умолчанию». В этом случае размер шрифта устанавливается автоматически. Для удобства организации ссылок на формулы, формулы нумеруются в естественном порядке, и лишь только те, на которые имеются ссылки по тексту.

**Оформление оглавления (содержания).** Кегль шрифта оглавления (содержания) понижают на один пункт по сравнению со шрифтом основного текста, сохраняя гарнитуру.

Слово «Оглавление» («Содержание») набирают прописным вариантом шрифта оглавления (содержания) или с выделением. Допустимо применение повышенного кегля шрифта, однако не выше кегля шрифта, старшей рубрики издания.

Оглавление создать автоматически «Вставка» – «Ссылка» – «Оглавление и указатели» – «Оглавление» (*применительно только при разметке Заголовок стилями, например Заголовок 1, Заголовок 2, Заголовок 3 и т.д.*). При составлении оглавления установить следующие свойства – показать номера страниц и номера страниц по правому краю. Заполнитель и формат оглавления выбирается по усмотрению студента.

**Оформление титульного листа.** Форма титульного листа приведена на рисунке 1.6.

Поволжский государственный университет
телекоммуникаций и информатики
Кафедра «Программное обеспечение и управление
в технических системах»

| | |
|---|---|
| Сдана на проверку | Допустить к защите |
| «___»_____________2011 г. | «___»______________2011 г. |

Лабораторная работа **№ 1**
по дисциплине
«Методы оптимизации в информационных системах»
на тему
«Решение задач линейного программирования с помощью
симплекс метода»

Пояснительная записка
на ____ листах

ВАРИАНТ ____

Студент группы ______
ФИО студента ________________

Самара 2011

Рисунок 1.6 – Форма титульного листа лабораторной работы

## СПИСОК ЛИТЕРАТУРЫ

1. Тарасов В.Н., Бахарева Н.Ф. Математическое программирование. Теория. Алгоритмы. Программы. – Оренбург: ИПК ОГУ, 2007 – 222 с.
2. Акулич И.Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов. – М.: Высш. шк., 1986. – 319 с.
3. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров: Учебное пособие. – М.: Высш. шк., 1994. – 554 с.
4. Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: Учебник для вузов. – М.: Изд-во МГТУ им. Баумана, 2001. – 440 с.
5. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. – М.: Мир, 1982. – 583с.
6. Банди Б. Методы оптимизации. Вводный курс: Пер. с англ. – М.: Радио и связь, 1988. – 128с.
7. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1988. – 552с.
8. Грешилов А.А. Прикладные задачи математического программирования М.: Изд-во МГТУ им. Н.Э. Баумана, 1990. – 189 с.
9. Данциг Дж. Линейное программирование: Пер. с англ. - М.: Прогресс, 1966. – 600 с.
10. Еремин И.И., Астафьев И.Н. Введение в теорию линейного и выпуклого программирования. – М.: Наука, 1976. – 192 с.
11. Карманов В.Г. Математическое программирование. – М.: Наука, 1975. – 272с.
12. Линейное и нелинейное программирование. /Под ред. проф. И.Н. Ляшенко. – Киев.:ВИЩА ШКОЛА, 1975. – 371 с.
13. Мину М. Математическое программирование. Теория и алгоритмы: Пер с франц. – М.: Наука, 1990. – 487 с.
14. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. – М.: Наука, 1986. – 328с.
15. Хедли Дж. Нелинейное и динамическое программирование: Пер. с англ. – М.: Мир, 1967. – 506с.
16. Юдин Д.Б., Гольштейн Е.Г. Линейное программирование. Теория, методы, приложения. – М.: Наука, 1969. – 424 с.