

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** |

**Институт** Информационных Технологий

**Кафедра** Вычислительной Техники

**ПРАКТИЧЕСКАЯ РАБОТА №5**

**по дисциплине**

**«Теория принятия решений»**

**Симплексный метод**

Студент группы:ИКБО-42-23 \_\_\_\_Голев С.С.\_\_\_\_ *(Ф. И.О. студента)*

Преподаватель \_\_Железняк Л.М.\_\_

*(Ф.И.О. преподавателя)*

|  |  |  |
| --- | --- | --- |
|  |  |  |

Москва 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc195720344)

[1 СИМПЛЕКСНЫЙ МЕТОД 4](#_Toc195720345)

[1.1 Постановка задачи 4](#_Toc195720346)

[1.2 Математическая модель задачи 4](#_Toc195720347)

[1.3 Результат работы программы 11](#_Toc195720348)

[ЗАКЛЮЧЕНИЕ 12](#_Toc195720349)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 13](#_Toc195720350)

[ПРИЛОЖЕНИЯ 14](#_Toc195720351)

ВВЕДЕНИЕ

Симплексный метод — это один из наиболее распространённых и эффективных алгоритмов решения задач линейного программирования. Он был разработан Джорджем Данцигом в 1947 году и используется для нахождения оптимального решения линейной целевой функции при наличии системы линейных ограничений.

Метод применяется в задачах, где требуется найти наилучшее распределение ограниченных ресурсов (времени, материалов, труда и т.д.) между различными видами деятельности с целью максимизации прибыли или минимизации затрат. Классические примеры применения симплексного метода — это задачи планирования производства, логистики, распределения ресурсов, управления проектами и другие экономико-математические задачи.

Суть метода заключается в последовательном переходе от одной допустимой вершины области решений к другой, улучшая значение целевой функции на каждом шаге, пока не будет достигнут оптимум. Симплексный метод является итерационным и хорошо масштабируется для задач с большим числом переменных и ограничений.

**1 СИМПЛЕКСНЫЙ МЕТОД**

* 1. **Постановка задачи**

***Задание 8.*** Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

***Задача.*** Для изготовления четырех видов продукции А, В, С и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

*Таблица П.8.* Исходные данные задачи.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ресурсы | Нормы расхода сырья на  единицу продукции, ед. | | | | Запасы  ресурсов, ед. |
| А | В | С | D |
| I | 2 | 1 | 0,5 | 4 | 3400 |
| II | 1 | 5 | 3 | 0 | 1200 |
| III | 3 | 0 | 6 | 1 | 3000 |
| Прибыль от единицы продукции, ден. ед. | 7,5 | 3 | 6 | 12 |  |

Требуется определить план выпуска продукции, при котором прибыль от ее реализации будет максимальной.

**1.2 Математическая модель задачи**

Пусть х1 – тип ресурса A, х2 –тип ресурса B, х3 –тип ресурса C, x4 – тип ресурса D. Прибыль от продажи шкафов составит 7.5х1 + 3х2 + 6х3+12x4, прибыль требуется максимизировать.

Ограничения задачи:

Таким образом, переходим к задаче линейного программирования:

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: х5 ≥ 0, х6 ≥ 0, х7 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

Построим начальную симплекс-таблицу. Запишем систему в векторной форме:

Векторы 𝐴5, 𝐴6, 𝐴7 являются линейно независимыми единичными векторами 3х-мерного пространства и образуют базис этого пространства.

Поэтому за базисные переменные выбираем переменные 𝑥5, 𝑥6, 𝑥7. Небазисными переменными являются 𝑥1, 𝑥2, *х*3, *x*4. Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные 𝑥1, 𝑥2, *х*3, *x*4 приравниваем нулю. В результате получим разложение

Которому соответствует первоначальный опорный план

Для проверки плана 𝑥(0) на оптимальность построим первую симплекс-таблицу. Введем в рассмотрение вектор коэффициентов целевой функции при базисных переменных.

В левый столбец Таблицы 5.2 запишем переменные 𝑥5, 𝑥6, 𝑥7 образующие базис, в верхней строке – небазисные переменные 𝑥1, 𝑥2, *x*3, *x*4. В строке 𝑐j запишем коэффициенты целевой функции, соответствующие небазисным переменным с1 = 9, с2 = 3, c3 = 6, c4 = 12. В столбце запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной 𝑥1, состоит из коэффициентов вектора . Аналогично, столбец, определяемый переменной 𝑥2, состоит из коэффициентов вектора . Аналогично, столбец, определяемый переменной 𝑥3, состоит из коэффициентов вектора . И столбец, определяемый переменной *х*4 состоит из коэффициентов вектора Крайний правый столбец заполняется элементами столбца , в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 1.3). Найдем относительные оценки ∆1, ∆2, ∆3, ∆4 и значение целевой функции 𝑄.

*Таблица 1.2 – Начальная симплекс-таблица задачи о максимальном доходе*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 12 |  |
|  |  | X1 | X2 | X3 | X4 |  |
| 0 | X5 | 2 | 1 | 0.5 | 4 | 3400 |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 |
| 0 | X7 | 3 | 0 | 6 | 1 | 3000 |
|  | f |  |  |  |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

*Таблица 1.3 – Заполнение f-строки*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 12 |  |  |
|  |  | X1 | X2 | X3 | X4 |  |  |
| 0 | X5 | 2 | 1 | 0.5 | 4 | 3400 | 3400 / 4 = 850 min |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 | не имеет смысла |
| 0 | X7 | 3 | 0 | 6 | 1 | 3000 | 3000 / 1 = 3000 |
|  | f | –7.5 | –3 | –6 | –12 | 0 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок ∆i ≥ 0. Так как оценки ∆1= −9, ∆2= −11, ∆3= −15 и ∆4= –12 в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка ∆4= −12. В базис будет включена соответствующая ей небазисная переменная 𝑥4. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной 𝑥5. Эта переменная исключается из базиса. В Таблице 1.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число 𝑎14 = 4.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 1.4 ).

*Таблица 1.4 – Новая симплекс-таблица*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |
|  |  | X1 | X2 | X3 | X5 |  |
| 12 | X4 |  |  |  | 1/4 |  |
| 0 | X6 |  |  |  |  |  |
| 0 | X7 |  |  |  |  |  |
|  | f |  |  |  |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

В Таблице 1.4 переменные 𝑥4 и 𝑥5 меняются местами вместе с коэффициентами 𝑐𝑗. Разрешающий элемент заменяется на обратный. В Таблице 1.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 1.5 – Симплекс преобразования*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |
|  |  | X1 | X2 | X3 | X5 |  |
| 12 | X4 | 1/2 | 1/4 | 1/8 | 1/4 | 850 |
| 0 | X6 |  |  |  | 0 |  |
| 0 | X7 |  |  |  | -1/4 |  |
|  | f |  |  |  | 3 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

*Таблица 1.6 – Итерация 0*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 6 | 0 |  |  |
|  |  | X1 | X2 | X3 | X5 |  |  |
| 12 | X4 | 1/2 | 1/4 | 1/8 | 1/4 | 850 | 850 / 1/8 = 6800 |
| 0 | X6 | 1 | 5 | 3 | 0 | 1200 | 1200 / 3 = 400 |
| 0 | X7 | 2.5 | -0.25 | 5.875 | -1/4 | 2150 | 2150 / 5.875 = 365 *min* |
|  | f | -1.5 | 0 | -4.5 | 3 |  |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

Остальные элементы (Таблица 1.6) рассчитываются по «правилу прямоугольника».

Базисное решение, которое дает последняя таблица

Это решение не является оптимальным, так как в f-строке имеются отрицательные оценки ∆1, ∆3.

*Таблица 5.7 – Итерация 1*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 0 | 0 |  |  |
|  |  | X1 | X2 | X7 | X5 |  |  |
| 12 | X4 | 0,45 | 0,25 | -0,02 | 0,25 | 804 | 804/0.25 = 3654 |
| 0 | X6 | -0,28 | 5,13 | -0,51 | 0,126 | 102 | 102/5 = 51 min |
| 6 | X3 | 0,42 | -0,042 | 0,17 | -0,042 | 365 | -1 |
|  | f | 0,41 | -0,19 | 0,77 | 2,811 | 11847 |  |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |  |

*Таблица 5.7 – Итерация 4*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | сj | 7.5 | 3 | 0 | 0 |  |
|  |  | X1 | X6 | X7 | X5 |  |
| 12 | X4 | 0,45 | -0.05 | 0 | 0,25 | 799 |
| 3 | X2 | -0,05 | 0.2 | -0,1 | 0,02 | 20 |
| 6 | X3 | 0,42 | 0 | 0,17 | -0,04 | 367 |
|  | f | 0,40 | 0.04 | 0,75 | 2,81 | 11851 |
|  |  | Δ1 | Δ2 | Δ3 | Δ4 | Q |

Если в последней таблице f-строке не содержит отрицательных оценок, то это свидетельствует об оптимальности полученного решения:

Подставляем базисное решение, которое дает последняя таблица

Где n – количество итераций

Проверим решение по «правилу прямоугольника».

Таким образом, фабрика должна выпускать в течении недели 𝑥4 = 800 шт. шкафов продукции D, 𝑥2 = 20 шт. шкафов продукции B, и 𝑥3 = 367 шт. продукции типа С. Тогда фабрика получит максимальный доход от продажи 11851 [тыс. ден.ед].

**1.3 Результат работы программы**

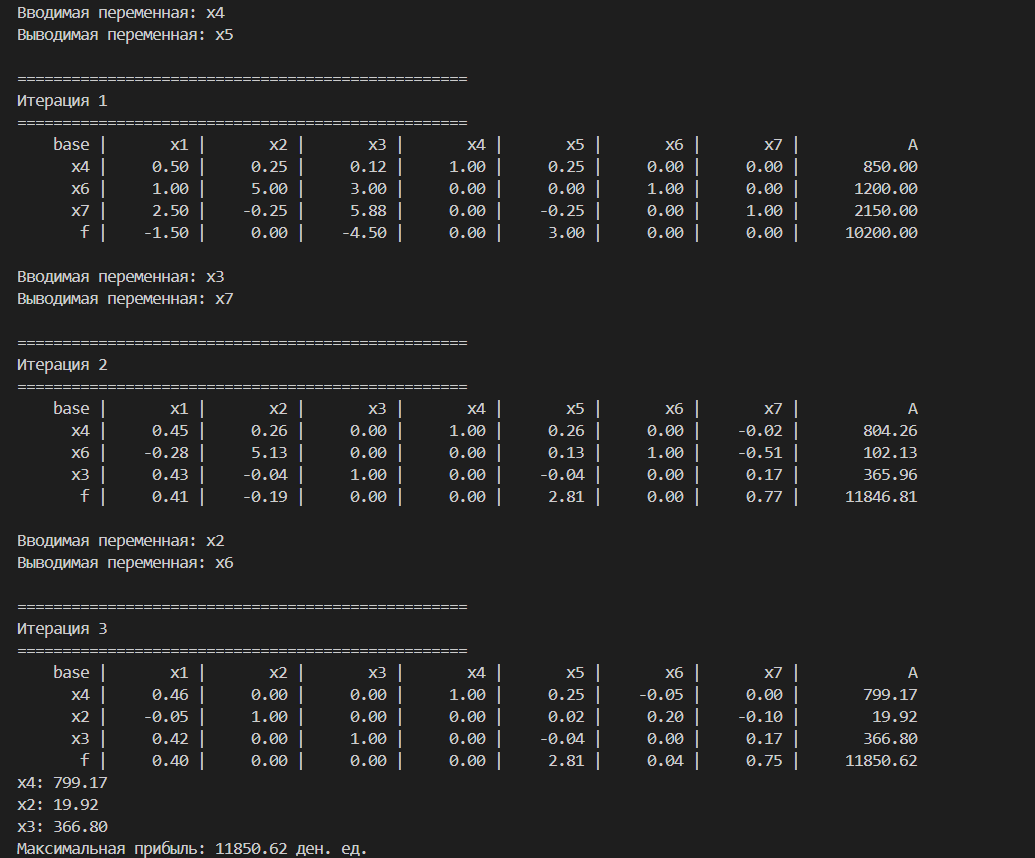


Рисунок 1 – Результат работы программы

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения практической работы была решена задача линейного программирования, связанная с оптимизацией производства подшипников двух типов. На основании исходных данных была составлена математическая модель: целевая функция, отражающая прибыль, и система ограничений, описывающих доступное время работы оборудования. Затем задача была приведена к канонической форме, после чего решена с использованием симплексного метода.

Симплексный метод продемонстрировал свою эффективность для данной задачи, позволив найти оптимальное распределение ресурсов, максимизирующее прибыль предприятия.

К основным преимуществам симплексного метода можно отнести: высокую точность при решении линейных задач; чёткую пошаговую процедуру, удобную для алгоритмизации; возможность применения к задачам с большим числом переменных и ограничений.

Однако у метода есть и недостатки: чувствительность к численным ошибкам при большом количестве итераций; неэффективность при решении задач с огромным числом переменных; невозможность применения к задачам с нелинейными ограничениями или целевой функцией.

В целом, симплексный метод остаётся универсальным и надёжным инструментом для решения широкого класса прикладных задач линейного программирования.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.
2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.
3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

**ПРИЛОЖЕНИЯ**

Приложение А – Код реализации симплексного метода на языке Python.

**Приложение А**

Код реализации симплексного метода на языке Python

*Листинг А.1. Реализация симплексного метда.*

import numpy as np

def print\_table(table, basis, columns, iteration):

print(f"\n{'='\*50}\nИтерация {iteration}\n{'='\*50}")

header = ["base"] + columns

rows = []

for i in range(len(table)-1):

row = [basis[i]] + [f"{val:.2f}" if abs(val) >= 0.01 else "0.00" for val in table[i]]

rows.append(row)

rows.append(["f"] + [f"{val:.2f}" for val in table[-1]])

for row in [header] + rows:

print("{: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >12}".format(\*row))

def simplex\_method():

num\_vars = 4

num\_slack = 3

columns = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'A']

basis = ['x5', 'x6', 'x7']

iteration = 1

table = np.array([

[2, 1, 0.5, 4, 1, 0, 0, 3400],

[1, 5, 3, 0, 0, 1, 0, 1200],

[3, 0, 6, 1, 0, 0, 1, 3000],

[-7.5, -3, -6, -12, 0, 0, 0, 0]

], dtype=float)

print\_table(table, basis, columns, 0)

while True:

z\_row = table[-1, :-1]

if all(z\_row >= 0):

break

entering\_col = np.argmin(z\_row)

print(f"\nВводимая переменная: {columns[entering\_col]}")

ratios = []

for row in table[:-1]:

if row[entering\_col] > 0:

ratios.append(row[-1] / row[entering\_col])

else:

ratios.append(np.inf)

exiting\_row = np.argmin(ratios)

print(f"Выводимая переменная: {basis[exiting\_row]}")

pivot = table[exiting\_row, entering\_col]

table[exiting\_row] = table[exiting\_row] / pivot

for i in range(len(table)):

if i != exiting\_row:

factor = table[i, entering\_col]

table[i] -= factor \* table[exiting\_row]

basis[exiting\_row] = columns[entering\_col]

print\_table(table, basis, columns, iteration)

iteration += 1

for var, val in zip(basis, table[:-1, -1]):

print(f"{var}: {val:.2f}")

print(f"Максимальная прибыль: {table[-1, -1]:.2f} ден. ед.")

simplex\_method()