

|  |
| --- |
| 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-41-23 \_\_Трофимов А.А.\_\_\_ *(Ф. И.О. студента)*

Преподаватель \_\_Железняк Л.М.\_\_

*(Ф.И.О. преподавателя)*

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

Москва 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc133167282)

[1СИМПЛЕКСНЫЙ МЕТОД 3](#_Toc133167283)

[1.1 Постановка задачи 3](#_Toc133167284)

[1.2 Математическая модель задачи 3](#_Toc133167285)

[ЗАКЛЮЧЕНИЕ 9](#_Toc133167286)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 10](#_Toc133167287)

[ПРИЛОЖЕНИЯ 11](#_Toc133167288)

ВВЕДЕНИЕ

Симплекс-метод — это классический алгоритм решения задач линейного программирования, который применяется для нахождения оптимального решения в задачах максимизации или минимизации целевой функции при наличии линейных ограничений. Он используется в экономике, логистике, производственном планировании и других областях, где требуется оптимизация ресурсов. В данной работе решается задача составления плана производства для максимизации дохода предприятия с использованием симплекс-метода, что позволяет продемонстрировать его эффективность.

**1СИМПЛЕКСНЫЙ МЕТОД**

* 1. **Постановка задачи**

Изготовление продукции двух видов I и II требует использование четырех видов сырья А, В. С и D. Запасы сырья ограничены. Указаны норма расхода каждого вида сырья на изготовление единицы продукции, запасы сырья и доход, получаемый от реализации.

**1.2 Математическая модель задачи**

Ограничения задачи:

Таким образом, переходим к задаче линейного программирования:

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: х3 ≥ 0, х4 ≥ 0, х5 ≥ 0, х5 ≥ 0. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

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

Векторы A3, 𝐴4, 𝐴5, 𝐴6 являются линейно независимыми единичными векторами 4х-мерного пространства и образуют базис этого пространства.

Поэтому за базисные переменные выбираем переменные x3, 𝑥4, 𝑥5, 𝑥6. Небазисными переменными являются 𝑥1, 𝑥2. Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные 𝑥1, 𝑥2 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 x3, 𝑥4, 𝑥5, 𝑥6 образующие базис, в верхней строке – небазисные переменные 𝑥1, 𝑥2. В строке 𝑐j запишем коэффициенты целевой функции, соответствующие небазисным переменным с1 = 7, с2 = 5. 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 . 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, и значение целевой функции 𝑄.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 7 | 5 |  |
|  |  | X1 | X2 |  |
| 0 | X3 | 2 | 3 | 1000 |
| 0 | X4 | 2 | 1 | 1300 |
| 0 | X5 | 0 | 3 | 1500 |
| 0 | X6 | 3 | 0 | 1800 |
|  | f |  |  |  |
|  |  | Δ1 | Δ2 | Q |

*Таблица 1.3 – Заполнение f-строки*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | сj | 7 | 5 |  |  |
|  |  | X1 | X2 |  |  |
| 0 | X3 | 2 | 3 | 1000 | 1000/2 = 500 min |
| 0 | X4 | 2 | 1 | 1300 | 1300/2 = 650 |
| 0 | X5 | 0 | 3 | 1500 | *---* |
| 0 | X6 | 3 | 0 | 1800 | 1800/3 = 600 |
|  | f | -7 | -5 | 0 |  |
|  |  | Δ1 | Δ2 | Q |  |

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок ∆i ≥ 0. Так как оценки ∆1= −7, ∆2= −5 в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка ∆1= −7. В базис будет включена соответствующая ей небазисная переменная 𝑥1. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной 𝑥3. Эта переменная исключается из базиса. В Таблице 5.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число 𝑎11 = 2 .

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

*Таблица 1.4 – Новая симплекс-таблица*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 5 |  |
|  |  | X3 | X2 |  |
| 7 | X1 | 1/2 |  |  |
| 0 | X4 |  |  |  |
| 0 | X5 |  |  |  |
| 0 | X6 |  |  |  |
|  | f |  |  |  |
|  |  | Δ1 | Δ2 | Q |

В Таблице 1.4 переменные 𝑥3 и 𝑥1 меняются местами вместе с коэффициентами 𝑐𝑗. Разрешающий элемент заменяется на обратный. В Таблице 5.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

*Таблица 1.5 – Симплекс преобразования*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | сj | 0 | 5 |  |
|  |  | X3 | X2 |  |
| 7 | X1 | 1/2 | 3/2 | 500 |
| 0 | X4 | -1 |  |  |
| 0 | X5 | 0 |  |  |
| 0 | X6 | -3/2 |  |  |
|  | f | 7/2 |  |  |
|  |  | Δ1 | Δ2 | Q |

*Таблица 1.6 – Итерация 0*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | сj | 0 | 5 |  |  |
|  |  | X3 | X2 |  |  |
| 7 | X1 | 1/2 | 3/2 | 500 |  |
| 0 | X4 | -1 | -2 | 300 |  |
| 0 | X5 | 0 | 3 | 1500 |  |
| 0 | X6 | -3/2 | -9/2 | 300 |  |
|  | f | 7/2 | 11/2 | 3500 |  |
|  |  | Δ1 | Δ2 | Q |  |

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

Базисное решение, которое дает последняя таблица

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

Проверим решение по «правилу прямоугольника».

Таким образом, x1 = 500, а x2 = 0 является оптимальным.

**ЗАКЛЮЧЕНИЕ**

В работе решена задача составления плана производства для максимизации дохода предприятия с использованием симплекс-метода. Были построены симплекс-таблицы, проведены итерации, и найдено оптимальное решение: x1 = 500, x2 = 0, f = 3500. Плюсы симплекс-метода: систематический подход, гарантирующий нахождение оптимального решения, и универсальность для задач линейного программирования. Минусы: сложность вычислений при большом числе переменных и ограничений, а также возможная вырожденность, приводящая к циклическим итерациям.

**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

# Функция для вывода симплекс-таблицы

def print\_tableau(tableau, basis, non\_basis):

print("\nСимплекс-таблица:")

header = [""] + non\_basis + ["RHS"]

print("\t".join(header))

for i in range(len(basis)):

row = [f"{basis[i]} ({tableau[i, 0]})"] + [f"{tableau[i, j+1]:.2f}" for j in range(tableau.shape[1]-1)]

print("\t".join(row))

# f-строка

f\_row = ["f"] + [f"{tableau[-1, j+1]:.2f}" for j in range(tableau.shape[1]-1)]

print("\t".join(f\_row))

# Симплекс-метод

def simplex\_method():

# Начальная симплекс-таблица

# Переменные: x1, x2, x3, x4, x5, x6, RHS

tableau = np.array([

[0, 2, 3, 1, 0, 0, 0, 1000], # x3: 2x1 + 3x2 + x3 = 1000

[0, 2, 1, 0, 1, 0, 0, 1300], # x4: 2x1 + x2 + x4 = 1300

[0, 0, 3, 0, 0, 1, 0, 1500], # x5: 3x2 + x5 = 1500

[0, 3, 0, 0, 0, 0, 1, 1800], # x6: 3x1 + x6 = 1800

[0, -7, -5, 0, 0, 0, 0, 0] # f: -7x1 - 5x2

], dtype=float)

# Базисные и небазисные переменные

basis = ["x3", "x4", "x5", "x6"]

non\_basis = ["x1", "x2"]

iteration = 0

while True:

print(f"\nИтерация {iteration}")

print\_tableau(tableau, basis, non\_basis)

# Шаг 1: Проверка на оптимальность (есть ли отрицательные элементы в f-строке)

if np.all(tableau[-1, 1:-1] >= 0):

print("\nОптимальное решение найдено!")

break

# Шаг 2: Выбор ведущего столбца (наибольший отрицательный коэффициент в f-строке)

pivot\_col = np.argmin(tableau[-1, 1:-1]) + 1

print(f"Ведущий столбец: {non\_basis[pivot\_col-1]} (индекс {pivot\_col})")

# Шаг 3: Выбор ведущей строки (минимальное положительное отношение RHS / элемент ведущего столбца)

ratios = []

*Продолжение листинга А.1*

for i in range(len(basis)):

if tableau[i, pivot\_col] > 0:

ratios.append(tableau[i, -1] / tableau[i, pivot\_col])

else:

ratios.append(float('inf'))

pivot\_row = np.argmin(ratios)

if ratios[pivot\_row] == float('inf'):

print("Задача не имеет конечного решения (неограничена).")

return

print(f"Ведущая строка: {basis[pivot\_row]} (индекс {pivot\_row}), отношение = {ratios[pivot\_row]:.2f}")

# Шаг 4: Пересчет таблицы

pivot\_element = tableau[pivot\_row, pivot\_col]

print(f"Разрешающий элемент: {pivot\_element}")

# Обновляем ведущую строку

tableau[pivot\_row, :] /= pivot\_element

# Обновляем остальные строки

for i in range(tableau.shape[0]):

if i != pivot\_row:

factor = tableau[i, pivot\_col]

tableau[i, :] -= factor \* tableau[pivot\_row, :]

# Обновляем базис

basis[pivot\_row] = non\_basis[pivot\_col-1]

iteration += 1

# Извлекаем решение

solution = {"x1": 0, "x2": 0}

for i in range(len(basis)):

var = basis[i]

if var in solution:

solution[var] = tableau[i, -1]

print("\nРешение:")

print(f"x1 = {solution['x1']:.2f}")

print(f"x2 = {solution['x2']:.2f}")

print(f"Максимальное значение целевой функции f = {tableau[-1, -1]:.2f}")

# Запускаем симплекс-метод

simplex\_method()