

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

|  |  |
| --- | --- |
| Заведующий кафедрой | «ПОВТ и АС» |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | В.В. Долгов |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

**ОТЧЕТ**

по лабораторно-практической работе по дисциплине «Исследование операций» по кафедре «Программное обеспечение вычислительной техники и автоматизированных системы»

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_А. А. Кизогян\_\_\_

подпись, дата

Обозначение отчета УП.81.0000.000 Группа \_\_\_\_ВМО31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направление 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль Математическое обеспечение и администрирование информационных систем

Преподаватель: проф. Никитина Алла Валерьевна

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата подпись преподавателя

Ростов-на-Дону

2023

# Содержание

1. Лабораторная работа №2 3
   1. Краткие теоретические сведения……………………………………...3
   2. Аналитическое решение……………………………………………….5
   3. Решение задачи линейного программирования стандартными средствами Mathcad………………………………………………….....7
   4. Решение полученное с помощью программного средства……….......8
2. Вопросы к защите лабораторной работы……………………………….....9
3. Вывод…………………………………………………………………...….10
4. Приложение А Листинг программы реализующий симплекс-метод…..11

**Лабораторная работа №2**

**Тема работы:** **Симплекс-метод.**

**Цель работы:** нахождение оптимальных решений задач линейного программирования с использованием симплекс-метода.

**Задание:**

Вариант 13

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

**Краткие теоретические сведения**

Симплекс-метод

Симплекс-метод позволяет эффективно найти оптимальное решение, избегая простой перебор всех возможных угловых точек. Основной принцип метода: вычисления начинаются с какого-то «стартового» базисного решения, а затем ведется поиск решений, «улучшающих» значение целевой функции. Это возможно только в том случае, если возрастание какой-то переменной приведет к увеличению значения функционала. Необходимые условия для применения симплекс-метода:

1. Задача должна иметь каноническую форму.
2. У задачи должен быть явно выделенный базис.

Замечание: Базисный вектор имеет размерность (m\*1), где m – количество уравнений в системе ограничений. Для удобства вычислений и наглядности обычно пользуются симплекс-таблицами:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Базис |  |  | … |  |  |
|  |  |  | … |  |  |
|  |  |  | … |  |  |
| … | … | … | … | … | **…** |
|  |  |  | … |  |  |
| Z |  |  | … |  | 0 |

* В первой строке указывают «наименование» всех переменных.
* В первом столбце указывают номера базисных переменных, а в последней ячейке – букву Z (это строка функционала).
* В «середине таблицы» указывают коэффициенты матрицы ограничений — .
* Последний столбец – вектор правых частей соответствующих уравнений системы ограничений.
* Крайняя правая ячейка – значение целевой функции. На первой итерации ее полагают равной 0.  
  Замечание: Базис – переменные, коэффициенты в матрице ограничений, при которых образуют базисные вектора.  
  Замечание: Если ограничения в исходной задаче представлены неравенствами вида ≤, то при приведении задачи к канонической форме, введенные дополнительные переменные образуют начальное базисное решение.  
  Замечание: Коэффициенты в строке функционала берутся со знаком “-”.  
  **Алгоритм симплекс-метода:**

1. Выбираем переменную, которую будем вводить в базис. Это делается в соответствии с указанным ранее принципом: мы должны выбрать переменную, возрастание которой приведет к росту функционала. Выбор происходит по следующему правилу:

* Если задача на минимум – выбираем максимальный положительный элемент в последней строке.
* Если задача на максимум – выбираем минимальный отрицательный.  
  Такой выбор, действительно, соответствует упомянутому выше принципу: если задача на минимум, то чем большее число вычитаем – тем быстрее убывает функционал; для максимума наоборот – чем большее число добавляем, тем быстрее функционал растет.  
  Замечание: хотя мы и берем минимальное отрицательное число в задаче на максимум, этот коэффициент показывает направление роста функционала, т.к. строка функционала в симплекс-таблице взята со знаком “-”. Аналогичная ситуация с минимизацией.  
  Определение: Столбец симплекс-таблицы, отвечающий выбранному коэффициенту, называется **ведущим столбцом**.

2. Выбираем переменную, которую будем вводить в базис. Для этого нужно определить, какая из базисных переменных быстрее всего обратится в нуль при росте новой базисной переменной. Алгебраически это делается так:

* Вектор правых частей почленно делится на ведущий столбец
* Среди полученных значений выбирают минимальное положительное (отрицательные и нулевые ответы не рассматривают)

Определение: Такая строка называется **ведущей строкой** и отвечает переменной, которую нужно вывести из базиса.  
Замечание: фактически, мы выражаем старые базисные переменные из каждого уравнения системы ограничений через остальные переменные и смотрим, в каком уравнении возрастание новой базисной переменной быстрее всего даст 0. Попадание в такую ситуацию означает, что мы «наткнулись» на новую вершину. Именно поэтому нулевые и отрицательные элементы не рассматриваются, т.к. получение такого результата означает, что выбор такой новой базисной переменной будет уводить нас из области, вне которой решений не существует.

3. Ищем элемент, стоящий на пересечении ведущих строки и столбца.  
Определение: Такой элемент называется **ведущим элементом**.  
4. Вместо исключаемой переменной в первом столбце (с названиями базисных переменных) записываем название переменной, которую мы вводим в базис.  
5. Далее начинается процесс вычисления нового базисного решения. Он происходит с помощью [метода Жордана-Гаусса](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0_%E2%80%94_%D0%96%D0%BE%D1%80%D0%B4%D0%B0%D0%BD%D0%B0).

* Новая Ведущая строка = Старая ведущая строка / Ведущий элемент
* Новая строка = Новая строка – Коэффициент строки в ведущем столбце \* Новая Ведущая строка

Замечание: Преобразование такого вида направлено на введение выбранной переменной в базис, т.е. представление ведущего столбца в виде базисного вектора.  
6. После этого проверяем условие оптимальности. Если полученное решение неоптимально – повторяем весь процесс снова.

**Аналитическое решение**

Вариант 13

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

– целевая функция

- вводим базисные переменные

- матрица коэффициентов

Полагая, что свободные переменные равны 0, получим первый опорный план: = .

Выбираем ведущий столбец, это будет столбец так как это наибольший коэффициент по модулю.

Находим наименьшее значение , как частное от деления из этого получаем что минимальным значением является . Вводим в базис за место переменную .

Из этого следует, что 6 является разрешающим, выделен жирным в таблице 1.

Применяем метод Жордана-Гаусса.

Таблица 1 – 1-й этап симплекс-метода

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Базис | b |  |  |  |  |  |
|  | 480 | 4 | 3 | 1 | 0 | 0 |
|  | 444 | 3 | 4 | 0 | 1 | 0 |
|  | 546 | 2 | **6** | 0 | 0 | 1 |
| Прибыль | 0 | -2 | -4 | 0 | 0 | 0 |

Текущий опорный план не оптимален, так как в индексной строке находятся отрицательные коэффициенты.

Следуя вышеописанным действиям, находим ведущий столбец и минимальное значение . Ведущим столбцом является столбец а минимальным значением является . Вводим в базис за место переменную .

Из этого следует, что является разрешающим, выделен жирным в таблице 2.

Применяем метод Жордана-Гаусса.

Таблица 2 - 2-й этап симплекс-метода

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Базис | b |  |  |  |  |  |
|  | 207 | 3 | 0 | 1 | 0 |  |
|  | 80 |  | 0 | 0 | 1 |  |
|  | 91 |  | 1 | 0 | 0 |  |
| Прибыль | 364 |  | 0 | 0 | 0 |  |

В итоге получаем новую симплекс-таблицу, показанную в таблице 3.

Таблица 3 - 3-й этап симплекс-метода

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Базис | b |  |  |  |  |  |
|  | 63 | 0 | 0 | 1 |  |  |
|  | 48 | 1 | 0 | 0 |  |  |
|  | 75 |  | 1 | 0 |  |  |
| Прибыль | 396 |  | 0 | 0 |  |  |

Текущий опорный план оптимален, так как в индексной строке не находятся отрицательные коэффициенты.

Оптимальный план также можно записать в виде:

,

Также стоит учитывать, что в оптимальный план вошла дополнительная переменная. Следовательно, при реализации такого плана имеются недоиспользованные ресурсы 1-го вида в количестве 63.

**Решение задачи линейного программирования стандартными средствами Mathcad**

На рисунке 1 приведено решение данной задачи в среде Mathcad, это обусловенно тем, что это необходимо для проверки решений найденных аналитическим методом и программным средством.

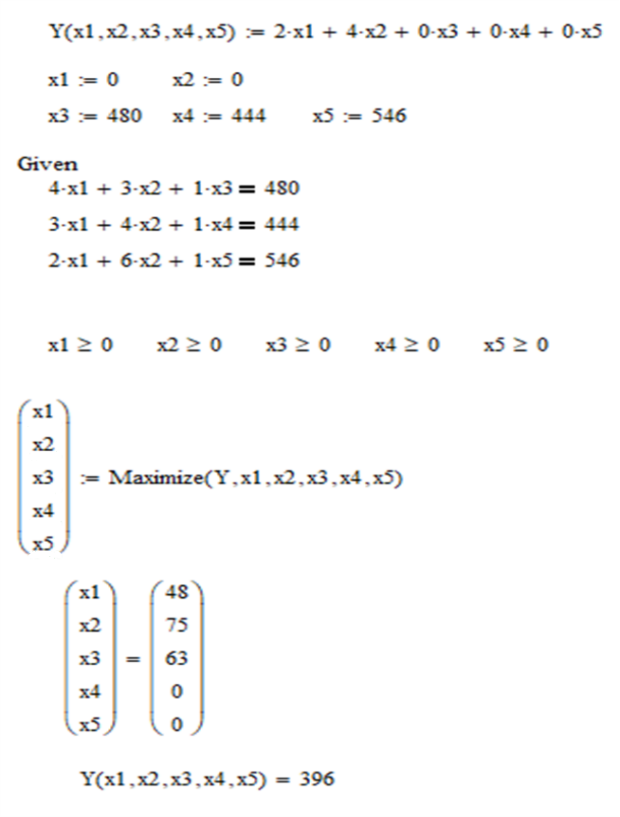


Рисунок 1 – Решение в Mathcad

**Решение полученное с помощью программного средства**

Благодаря программному средству, можно найти оптимальное решение задачи линейного программирования с использованием симплекс-метода. Описание программы приведено в Приложении А. Решение и вывод представлены на рисунках 2 и 3.

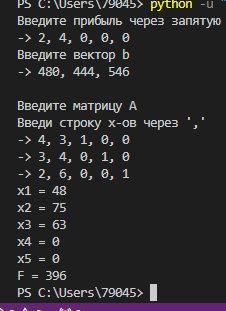


Рисунок 2 - Программная реализация и результат работы программы варианта (13)

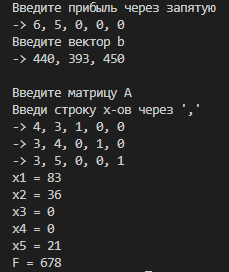


Рисунок 3 - Программная реализация и результат работы программы варианта (3)

**Вопросы к защите лабораторной работы**

**1. В какой форме должна быть записана ЗЛП для решения ее симплекс-методом?**

Задача должна иметь каноническую форму. У задачи должен быть явно выделенный базис.

**2. Какие переменные являются зависимыми (базисными), а какие –**

**независимыми (свободными)?**

Базисные зависят от значений других переменных и они принимают определенные значения, которые получаются в результате решения системы уравнений. Независимые переменные, или свободные переменные, могут принимать любые значения и не зависят от результатов решения системы уравнений.

**3. Какие переменные приравниваются к «0» для нахождения первого опорного плана?**

Свободные переменные.

**4. Что является первым опорным решением ЗЛП?**

Опорным решением задачи линейного программирования называется такое допустимое решение, для которого векторы условий, соответствующие положительным координатам, линейно независимы.

**5. Что выступает критерием оптимальности при минимизации (максимизации) целевой функции?**

Критерий оптимальности — характерный показатель решения задачи, по значению которого оценивается оптимальность найденного решения, то есть максимальное удовлетворение поставленным требованиям.

**6. Каков алгоритм симплекс-метода?**

1. Выбираем переменную, которую будем вводить в базис. Мы должны выбрать переменную, возрастание которой приведет к росту функционала. Выбор происходит по следующему правилу:

Если задача на минимум – выбираем максимальный положительный элемент в последней строке.

Если задача на максимум – выбираем минимальный отрицательный.

2. Выбираем переменную, которую будем вводить в базис. Для этого нужно определить, какая из базисных переменных быстрее всего обратится в нуль при росте новой базисной переменной. Алгебраически это делается так: вектор правых частей почленно делится на ведущий столбец.

Среди полученных значений выбирают минимальное положительное (отрицательные и нулевые ответы не рассматривают).

3. Ищем элемент, стоящий на пересечении ведущих строки и столбца.

4. Вместо исключаемой переменной в первом столбце (с названиями базисных переменных) записываем название переменной, которую мы вводим в базис.

5. Далее начинается процесс вычисления нового базисного решения. Он происходит с помощью метода Жордана-Гаусса.

Новая Ведущая строка = Старая ведущая строка / Ведущий элемент

Новая строка = Новая строка – Коэффициент строки в ведущем столбце \* Новая Ведущая строка

6. После этого проверяем условие оптимальности. Если полученное решение неоптимально – повторяем весь процесс снова.

**7. Как выбирается направляющий столбец?**

Направляющий столбец выбирается при наличии в нем хотя бы одного положительного элемента.

**8. Как выбирается направляющая строка?**

Направляющая строка выбирается по минимальному отношению при условии.

**9. Какой элемент называется разрешающим?**

Элемент, который находится на пересечении разрешающих столбца и строки, называется разрешающим элементом.

**10. Когда ЗЛП не имеет решения?**

ЗЛП не имеет решения, если какой-нибудь столбец коэффициентов свободной переменной не содержит ни одного положительного элемента.

**11. Как определяются элементы новой симплекс-таблицы (разрешающий элемент, элементы направляющего столбца и строки, остальные элементы)**

Разрешающий элемент становится единицей. Элементы разрешающей строки делятся на разрешающий элемент, элементы разрешающего столбца обнуляются, остальные элементы пересчитываются по формуле \* = — ( A × B / РЭ ).

**Вывод**

Экономико-математические модели и методы - ЭМММ - представляют системный подход к управлению проблемами. Они используются для создания абстрактных моделей проблемных ситуаций в виде математических соотношений или графиков. Техника линейного программирования (ЛП) используется для решения задач, связанных с ограничениями и целевой функцией. Однако, экономические и производственные системы сложны для моделирования, из-за непрерывности изменений и других факторов. Поэтому экономико-математические модели имеют ограниченное применение в реальных системах.

**Приложение А Листинг программы**

import numpy as np

def get\_pivot\_position(tableau):

    z = tableau[-1]

    column = next(i for i, x in enumerate(z[:-1]) if x > 0)

    restrictions = []

    for eq in tableau[:-1]:

        el = eq[column]

        restrictions.append(math.inf if el <= 0 else eq[-1] / el)

    row = restrictions.index(min(restrictions))

    return row, column

def pivot\_step(tableau, pivot\_position):

    new\_tableau = [[] for eq in tableau]

    i, j = pivot\_position

    pivot\_value = tableau[i][j]

    new\_tableau[i] = np.array(tableau[i]) / pivot\_value

    for eq\_i, eq in enumerate(tableau):

        if eq\_i != i:

            multiplier = np.array(new\_tableau[i]) \* tableau[eq\_i][j]

            new\_tableau[eq\_i] = np.array(tableau[eq\_i]) - multiplier

    return new\_tableau

def is\_basic(column):

    return sum(column) == 1 and len([c for c in column if c == 0]) == len(column) - 1

def get\_solution(tableau):

    columns = np.array(tableau).T

    solutions = []

    for column in columns[:-1]:

        solution = 0

        if is\_basic(column):

            one\_index = column.tolist().index(1)

            solution = columns[-1][one\_index]

        solutions.append(solution)

    return solutions

def simplex(c, A, b):

    tableau = to\_tableau(c, A, b)

    while can\_be\_improved(tableau):

        pivot\_position = get\_pivot\_position(tableau)

        tableau = pivot\_step(tableau, pivot\_position)

    return get\_solution(tableau)

def to\_tableau(c, A, b):

    xb = [eq + [x] for eq, x in zip(A, b)]

    z = c + [0]

    return xb + [z]

def can\_be\_improved(tableau):

    z = tableau[-1]

    return any(x > 0 for x in z[:-1])

tmp\_c = str(input('Введите прибыль через запятую\n-> ')).replace(" ", "").split(',')

c = list(map(int, tmp\_c))

tmp\_c = str(input('Введите вектор b\n-> ')).replace(" ", "").split(',')

b = list(map(int, tmp\_c))

A = []

print("\nВведите матрицу A\nВведи строку x-ов через ','")

for i in range(0, len(b)):

    tmp\_c = str(input('-> ')).replace(" ", "").split(',')

    tmpVec = list(map(int, tmp\_c))

    A.append(tmpVec)

import math

solution = simplex(c, A, b)

for i in range(0, len(solution)):

    print(f"x{i+1} = {int(solution[i] + (0.5 if solution[i] > 0 else -0.5))}")

sum\_F = 0

for i in range(0, len(solution)):

    for j in range(0, len(c)):

        if i == j:

            sum\_F += int(solution[i] + (0.5 if solution[i] > 0 else -0.5)) \* c[j]

print(f'F = {sum\_F}')