МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

по дисциплине: Исследование операций
тема: “Исследование множества опорных планов системы ограничений задачи линейного программирования (задачи ЛП) в канонической форме”

Выполнил: ст. группы ВТ-231
Масленников Даниил

Проверил:
Вирченко Юрий Петрович

Белгород, 2025 г.

Лабораторная работа №1 «Исследование множества опорных планов системы ограничений задачи линейного программирования (задачи ЛП) в канонической форме.»

Цель работы: изучить метод Гаусса-Жордана и операцию замещения, а также освоить их применение к отысканию множества допустимых базисных видов системы линейных уравнений, и решению задачи линейного программирования простым перебором опорных решений.

Вариант 14

$$\begin{cases} -x_1 + 5x_2 - 4x_3 - 6x_4 + x_6 = -9 \\ 8x_1 + x_2 - x_3 + 2x_5 + 3x_6 = 8 \\ 4x_1 + 3x_2 - 2x_3 + 9x_4 + x_5 + 7x_6 = 1 \end{cases}$$

Оглавление

Блок-схема программы

images/block.png

Код программы

```
import itertools
import numpy as np

EPS = 1e-9

def rearrange_columns(matrix, basis):
    cols = len(matrix[0]) - 1
    new_matrix = []
    for row in matrix:
        new_row = [row[b] for b in basis]
        new_row += [row[c] for c in range(cols) if c not in basis]
        new_row.append(row[-1])
        new_matrix.append(new_row)
    return new_matrix

def gauss_jordan(matrix):
    rows = len(matrix)
    cols = len(matrix[0]) - 1
    leading_columns = []
    lead = 0

    for r in range(rows):
        if lead >= cols:
            break
        i = r
        while abs(matrix[i][lead]) < EPS:
            i += 1
            if i == rows:
                i = r
                lead += 1
                if cols == lead:
                    break
        if i == rows:
            continue
        matrix[i], matrix[r] = matrix[r], matrix[i]
        div = matrix[r][lead]
        matrix[r] = [x / div for x in matrix[r]]
        for i in range(rows):
            if i != r:
                factor = matrix[i][lead]
                matrix[i] = [matrix[i][j] - factor * matrix[r][j] for j in range(cols + 1)]
        leading_columns.append(lead)
        lead += 1
```

```
        return matrix, leading_columns

    def is_valid_basis(basis, leading_columns):
        return len(leading_columns) == len(basis) and all(lc < len(basis) for lc in leading_colu

    def is_feasible(solution):
        return all(x >= -EPS for x in solution)

    def objective_function(solution):
        return sum(solution)

    def print_solution(matrix, basis, solution):
        print("Базисные переменные:", " ".join(f"x{b+1}" for b in basis))
        print("Решение:")
        for r in range(len(matrix)):
            lead_col = -1
            for c in range(len(basis)):
                if abs(matrix[r][c] - 1.0) < EPS:
                    lead_col = c
                    break
            if lead_col == -1:
                if abs(matrix[r][-1]) > EPS:
                    print("Система несовместна")
                continue
            actual_var = basis[lead_col]
            print(f"x{actual_var + 1} = {matrix[r][-1]:.2f}", end="")
            for c in range(len(basis), len(matrix[r]) - 1):
                coeff = matrix[r][c]
                if abs(coeff) > EPS:
                    var_index = c - len(basis)
                    if coeff < 0:
                        print(f" + {-coeff:.2f}*x{var_index + 1}", end="")
                    else:
                        print(f" - {coeff:.2f}*x{var_index + 1}", end="")
            print()
        print("--------------------------")

    def find_basic_solutions(original, num_vars):
        num_eqs = len(original)
        vars_list = list(range(num_vars))
        selector = [True] * num_eqs + [False] * (num_vars - num_eqs)
        feasible_solutions = []
        count = 0

        while True:
            basis = [i for i, selected in enumerate(selector) if selected]
            if len(basis) != num_eqs:
```

```
            continue
        matrix = rearrange_columns(original, basis)
        matrix, leading_columns = gauss_jordan(matrix)

        print(f"Комбинация базиса {count + 1}:")
        print("Преобразованная матрица:")
        for row in matrix:
            print(" ".join(f"{val:8.2f}" for val in row))
        count += 1

        if is_valid_basis(basis, leading_columns):
            solution = [0.0] * num_vars
            for r in range(len(matrix)):
                for c in range(len(basis)):
                    if abs(matrix[r][c] - 1.0) < EPS:
                        solution[basis[c]] = matrix[r][-1]
                        break

            if is_feasible(solution):
                feasible_solutions.append(solution)
                print_solution(matrix, basis, solution)
                print(f"Целевая функция: {objective_function(solution)}")
                print("-------------------------")
            else:
                print("Решение недопустимо (есть отрицательные переменные)")
                print("-------------------------")
        else:
            print("Не является допустимым базисом")
            print("-------------------------")

        if not prev_permutation(selector):
            break

    if feasible_solutions:
        optimal_solution = min(feasible_solutions, key=objective_function)
        print("\nОптимальный опорный план:")
        print(" ".join(f"{x:.2f}" for x in optimal_solution))
        print(f"Целевая функция: {objective_function(optimal_solution)}")
    else:
        print("\nДопустимые решения отсутствуют!")

def prev_permutation(arr):
    i = len(arr) - 1
    while i > 0 and arr[i - 1] <= arr[i]:
        i -= 1
    if i <= 0:
        return False
```

```
    j = len(arr) - 1
    while arr[j] >= arr[i - 1]:
        j -= 1
    arr[i - 1], arr[j] = arr[j], arr[i - 1]
    arr[i:] = arr[:i - 1:-1]
    return True

if __name__ == "__main__":
    matrix = [
        [-1, 5, -4, -6, 0, 1, -9],
        [8, 1, -1, 0, 2, 3, 8],
        [4, 3, -2, 9, 1, 7, 1]
    ]
    find_basic_solutions(matrix, 6)
```

Результаты работы программы:

images/1.png

images/2.png

images/3.png

Аналитическое решение

$$\begin{cases} -x_1 + 5x_2 - 4x_3 - 6x_4 + x_6 = -9 \\ 8x_1 + x_2 - x_3 + 2x_5 + 3x_6 = 8 \\ 4x_1 + 3x_2 - 2x_3 + 9x_4 + x_5 + 7x_6 = 1 \end{cases}$$

- Так как n $>$ m (переменных больше чем уравнений, то система имеет бесконечно много решений)
- Расширенная матрица:

$$\begin{pmatrix} -1 & 5 & -4 & -6 & 0 & 1 & | & -9 \\ 8 & 1 & -1 & 0 & 2 & 3 & | & 8 \\ 4 & 3 & -2 & 9 & 1 & 7 & | & 1 \end{pmatrix}$$

- Приведение матрицы:

1. a_{11} = -1
   - Делим первую строку на -1:

$$\begin{pmatrix} 1 & -5 & 4 & 6 & 0 & -1 & | & 9 \\ 8 & 1 & -1 & 0 & 2 & 3 & | & 8 \\ 4 & 3 & -2 & 9 & 1 & 7 & | & 1 \end{pmatrix}$$

   - Умножаем первую строку на 8 и вычитаем из второй:

$$\begin{pmatrix} 1 & -5 & 4 & 6 & 0 & -1 & | & 9 \\ 0 & 41 & -33 & -48 & 2 & 11 & | & -64 \\ 4 & 3 & -2 & 9 & 1 & 7 & | & 1 \end{pmatrix}$$

   - Умножаем первую строку на 4 и вычитаем из третьей:

$$\begin{pmatrix} 1 & -5 & 4 & 6 & 0 & -1 & | & 9 \\ 0 & 41 & -33 & -48 & 2 & 11 & | & -64 \\ 0 & 23 & -18 & -15 & 1 & 11 & | & -35 \end{pmatrix}$$

2. $a_{22} = 41$

   – Делим вторую строку на 41:

$$\left(\begin{array}{cccccc|c} 1 & -5 & 4 & 6 & 0 & -1 & 9 \\ 0 & 1 & -\frac{33}{41} & -\frac{48}{41} & \frac{2}{41} & \frac{11}{41} & -\frac{64}{41} \\ 0 & 23 & -18 & -15 & 1 & 11 & -35 \end{array}\right)$$

   – Умножаем вторую строку на -5 и вычитаем из первой:

$$\left(\begin{array}{cccccc|c} 1 & 0 & -\frac{1}{41} & \frac{6}{41} & \frac{10}{41} & \frac{14}{41} & \frac{49}{41} \\ 0 & 1 & -\frac{33}{41} & -\frac{48}{41} & \frac{2}{41} & \frac{11}{41} & -\frac{64}{41} \\ 0 & 23 & -18 & -15 & 1 & 11 & -35 \end{array}\right)$$

   – Умножаем вторую строку на 23 и вычитаем из третьей:

$$\left(\begin{array}{cccccc|c} 1 & 0 & -\frac{1}{41} & \frac{6}{41} & \frac{10}{41} & \frac{14}{41} & \frac{49}{41} \\ 0 & 1 & -\frac{33}{41} & -\frac{48}{41} & \frac{2}{41} & \frac{11}{41} & -\frac{64}{41} \\ 0 & 0 & \frac{21}{41} & \frac{489}{41} & -\frac{5}{41} & \frac{198}{41} & \frac{37}{41} \end{array}\right)$$

3. $a_{33} = \frac{21}{41}$

   – Делим третью строку на $\frac{21}{41}$:

$$\left(\begin{array}{cccccc|c} 1 & 0 & -\frac{1}{41} & \frac{6}{41} & \frac{10}{41} & \frac{14}{41} & \frac{49}{41} \\ 0 & 1 & -\frac{33}{41} & -\frac{48}{41} & \frac{2}{41} & \frac{11}{41} & -\frac{64}{41} \\ 0 & 0 & 1 & \frac{163}{7} & -\frac{5}{21} & \frac{66}{7} & \frac{37}{21} \end{array}\right)$$

   – Умножаем третью строку на $-\frac{33}{41}$ и вычитаем из второй:

$$\left(\begin{array}{cccccc|c} 1 & 0 & -\frac{1}{41} & \frac{6}{41} & \frac{10}{41} & \frac{14}{41} & \frac{49}{41} \\ 0 & 1 & 0 & -\frac{123}{7} & -\frac{1}{7} & \frac{55}{7} & -\frac{1}{7} \\ 0 & 0 & 1 & \frac{163}{7} & -\frac{5}{21} & \frac{66}{7} & \frac{37}{21} \end{array}\right)$$

   – Умножаем третью строку на $-\frac{1}{41}$ и вычитаем из первой:

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & \frac{5}{7} & \frac{5}{21} & \frac{4}{7} & \frac{26}{21} \\ 0 & 1 & 0 & -\frac{123}{7} & -\frac{1}{7} & \frac{55}{7} & -\frac{1}{7} \\ 0 & 0 & 1 & \frac{163}{7} & -\frac{5}{21} & \frac{66}{7} & \frac{37}{21} \end{array}\right)$$

- Матрица в упрощенном ступенчатом виде:

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & \frac{5}{7} & \frac{5}{21} & \frac{4}{7} & \frac{26}{21} \\ 0 & 1 & 0 & -\frac{123}{7} & -\frac{1}{7} & \frac{55}{7} & -\frac{1}{7} \\ 0 & 0 & 1 & \frac{163}{7} & -\frac{5}{21} & \frac{66}{7} & \frac{37}{21} \end{array}\right)$$

- После всех преобразований матрица имеет вид:

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & a_{14} & a_{15} & a_{16} & b_1 \\ 0 & 1 & 0 & a_{24} & a_{25} & a_{26} & b_3 \\ 0 & 0 & 1 & a_{34} & a_{35} & a_{36} & b_2 \end{array}\right)$$

- Базисные переменные: x_1, x_2, x_3. Выражаем их через свободные x_4, x_5, x_6:

$$\begin{cases} x_1 = b_1 - a_{14}x_4 - a_{15}x_5 - a_{16}x_6 \\ x_2 = b_2 - a_{24}x_4 - a_{25}x_5 - a_{26}x_6 \\ x_3 = b_3 - a_{34}x_4 - a_{35}x_5 - a_{36}x_6 \end{cases}$$

- Итоговое решение:

$$\begin{cases} x_1 = \dfrac{26}{21} - \dfrac{5}{7}x_4 - \dfrac{5}{21}x_5 - \dfrac{4}{7}x_6 \\ x_2 = -\dfrac{1}{7} - \dfrac{123}{7}x_4 + \dfrac{1}{7}x_5 - \dfrac{55}{7}x_6 \\ x_3 = \dfrac{37}{21} - \dfrac{163}{7}x_4 + \dfrac{5}{21}x_5 - \dfrac{66}{7}x_6 \end{cases}$$

Найдем один из опорных планов (Всего их $C_6^3 = 20$):

- Для базисных переменных x_1, x_2, x_4
- Расширенная матрица:

$$\begin{pmatrix} -1 & 5 & -6 & -4 & 0 & 1 & | & -9 \\ 8 & 1 & 0 & -1 & 2 & 3 & | & 8 \\ 4 & 3 & 9 & -2 & 1 & 7 & | & 1 \end{pmatrix}$$

1. a_{11} = -1
   - Делим первую строку на -1:

$$\begin{pmatrix} 1 & -5 & 6 & 4 & 0 & -1 & | & 9 \\ 8 & 1 & 0 & -1 & 2 & 3 & | & 8 \\ 4 & 3 & 9 & -2 & 1 & 7 & | & 1 \end{pmatrix}$$

   - Умножаем первую строку на 8 и вычитаем из второй:

$$\begin{pmatrix} 1 & -5 & 6 & 4 & 0 & -1 & | & 9 \\ 0 & 41 & -48 & -33 & 2 & 11 & | & -64 \\ 4 & 3 & 9 & -2 & 1 & 7 & | & 1 \end{pmatrix}$$

   - Умножаем первую строку на 4 и вычитаем из третьей:

$$\begin{pmatrix} 1 & -5 & 6 & 4 & 0 & -1 & | & 9 \\ 0 & 41 & -48 & -33 & 2 & 11 & | & -64 \\ 0 & 23 & -15 & -18 & 1 & 11 & | & -35 \end{pmatrix}$$

2. a_{22} = 41
   - Делим вторую строку на 41:

$$\begin{pmatrix} 1 & -5 & 6 & 4 & 0 & -1 & | & 9 \\ 0 & 1 & -\frac{48}{41} & -\frac{33}{41} & \frac{2}{41} & \frac{11}{41} & | & -\frac{64}{41} \\ 0 & 23 & -15 & -18 & 1 & 11 & | & -35 \end{pmatrix}$$

   - Умножаем вторую строку на -5 и вычитаем из первой:

$$\begin{pmatrix} 1 & 0 & \frac{6}{41} & -\frac{1}{41} & \frac{10}{41} & \frac{14}{41} & | & \frac{49}{41} \\ 0 & 1 & -\frac{48}{41} & -\frac{33}{41} & \frac{2}{41} & \frac{11}{41} & | & -\frac{64}{41} \\ 0 & 23 & -15 & -18 & 1 & 11 & | & -35 \end{pmatrix}$$

   - Умножаем вторую строку на 23 и вычитаем из третьей:

$$\begin{pmatrix} 1 & 0 & \frac{6}{41} & -\frac{1}{41} & \frac{10}{41} & \frac{14}{41} & | & \frac{49}{41} \\ 0 & 1 & -\frac{48}{41} & -\frac{33}{41} & \frac{2}{41} & \frac{11}{41} & | & -\frac{64}{41} \\ 0 & 0 & \frac{489}{41} & \frac{21}{41} & -\frac{5}{41} & \frac{198}{41} & | & \frac{37}{41} \end{pmatrix}$$

3. $a_{33} = \frac{21}{41}$

   – Делим третью строку на $\frac{489}{41}$:

$$\begin{pmatrix} 1 & 0 & \frac{6}{41} & -\frac{1}{41} & \frac{10}{41} & \frac{14}{41} & | & \frac{49}{41} \\ 0 & 1 & -\frac{48}{41} & -\frac{33}{41} & \frac{2}{41} & \frac{11}{41} & | & -\frac{64}{41} \\ 0 & 0 & 1 & \frac{7}{163} & -\frac{5}{489} & \frac{66}{163} & | & \frac{37}{489} \end{pmatrix}$$

   – Умножаем третью строку на $-\frac{48}{41}$ и вычитаем из второй:

$$\begin{pmatrix} 1 & 0 & \frac{6}{41} & -\frac{1}{41} & \frac{10}{41} & \frac{14}{41} & | & \frac{49}{41} \\ 0 & 1 & 0 & -\frac{123}{163} & \frac{6}{163} & \frac{121}{163} & | & \frac{240}{163} \\ 0 & 0 & 1 & \frac{7}{163} & -\frac{5}{489} & \frac{66}{163} & | & \frac{37}{489} \end{pmatrix}$$

   – Умножаем третью строку на $-\frac{6}{41}$ и вычитаем из первой:

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1020}{6683} & -\frac{173}{6683} & \frac{2678}{6683} & | & \frac{8061}{6683} \\ 0 & 1 & 0 & -\frac{123}{163} & \frac{6}{163} & \frac{121}{163} & | & \frac{240}{163} \\ 0 & 0 & 1 & \frac{7}{163} & -\frac{5}{489} & \frac{66}{163} & | & \frac{37}{489} \end{pmatrix}$$

- После всех преобразований матрица имеет вид:

$$\begin{pmatrix} 1 & 0 & 0 & a_{14} & a_{15} & a_{16} & | & b_1 \\ 0 & 1 & 0 & a_{24} & a_{25} & a_{26} & | & b_3 \\ 0 & 0 & 1 & a_{34} & a_{35} & a_{36} & | & b_2 \end{pmatrix}$$

- Базисные переменные: x_1, x_2, x_3. Выражаем их через свободные x_4, x_5, x_6:

$$\begin{cases} x_1 = b_1 - a_{14}x_4 - a_{15}x_5 - a_{16}x_6 \\ x_2 = b_2 - a_{24}x_4 - a_{25}x_5 - a_{26}x_6 \\ x_3 = b_3 - a_{34}x_4 - a_{35}x_5 - a_{36}x_6 \end{cases}$$

- Итоговое решение:

$$\begin{cases} x_1 = \dfrac{8061}{6683} - \dfrac{1020}{6683}x_4 + \dfrac{173}{6683}x_5 - \dfrac{2678}{6683}x_6 \\ x_2 = \dfrac{240}{163} + \dfrac{123}{163}x_4 - \dfrac{6}{163}x_5 - \dfrac{121}{163}x_6 \\ x_3 = \dfrac{37}{489} - \dfrac{7}{163}x_4 + \dfrac{5}{489}x_5 - \dfrac{66}{163}x_6 \end{cases}$$

Вывод: в ходе выполнения лабораторной работы я составил программу для отыскания всех базисных решений системы уравнений с помощью метода Гаусса-Жордана, вывод которой совпал с ответом в моем аналитическом решении.