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 №8

по дисциплине: Исследование операций
тема: «Задачи дробно-линейного программирования (задачи ДЛП)»

Выполнил: ст. группы ПВ-233
Мороз Роман Алексеевич

Проверил:
Вирченко Юрий Петрович

Белгород 2025 г.

Цель работы: Освоить метод сведения задачи ДЛП к задаче линейного программирования с помощью введения новых переменных. Изучить алгоритм решения задачи ДЛП и реализовать программно этот алгоритм.

Задания для подготовки к работе

1. Изучить постановку задачи ДЛП, а также подходы к ее решению.
2. Ознакомиться с введением новых переменных, в которых задача ДЛП превращается в задачу ЛП.
3. Изучить метод и алгоритм решения задачи ДЛП, составить и отладить программу решения этой задачи, используя в качестве тестовых данных одну из нижеследующих задач, решенную вручную.

9.

$$z = \frac{-7x_1 + 3x_2}{3x_1 + 8x_2} \to \max;$$

$$\begin{cases} x_1 + 5x_2 + x_3 = 16, \\ 6x_1 + 3x_2 - x_4 = 14, \\ 5x_1 - 2x_2 + x_5 = 23, \end{cases}$$

$$x_i \geq 0 \left(i = \overline{1, 5}\right)$$

Аналитическое решение:

Целевая функция:

$$Z = \frac{-7x_1 + 3x_2}{3x_1 + 8x_2} \to \max$$

Ограничения:

$$\begin{cases} x_1 + 5x_2 + x_3 = 16, \\ 6x_1 + 3x_2 - x_4 = 14, \\ 5x_1 - 2x_2 + x_5 = 23, \end{cases}$$

Решение системы уравнений

Шаг 1. Выражаем y_1 из уравнения $3y_1 + 8y_2 = 1$:

$$y_1 = \frac{1 - 8y_2}{3}.$$

Шаг 2. Подставляем y_1 в остальные ограничения:

1. **Первое ограничение:**

$$y_3 = 16t - \frac{1 - 8y_2}{3} - 5y_2 = 16t - \frac{1}{3} - \frac{7y_2}{3}.$$

Условие $y_3 \geq 0$:

$$16t \geq \frac{1 + 7y_2}{3} \quad \Rightarrow \quad t \geq \frac{1 + 7y_2}{48}.$$

2. **Второе ограничение:**

$$y_4 = 6 \cdot \frac{1 - 8y_2}{3} + 3y_2 - 14t = 2 - 16y_2 - 14t.$$

Условие $y_4 \geq 0$:

$$16y_2 + 14t \leq 2 \quad \Rightarrow \quad t \leq \frac{2 - 16y_2}{14}.$$

3. **Третье ограничение:**

$$y_5 = 5 \cdot \frac{1-8y_2}{3} - 2y_2 + 23t = \frac{5}{3} - \frac{46y_2}{3} + 23t.$$

Условие $y_5 \geq 0$:

$$23t \geq \frac{46y_2 - 5}{3} \quad \Rightarrow \quad t \geq \frac{46y_2 - 5}{69}.$$

Шаг 3. Максимизация целевой функции:

Целевая функция:

$$Z = -7 \cdot \frac{1-8y_2}{3} + 3y_2 = -\frac{7}{3} + \frac{65y_2}{3}.$$

Для максимизации Z необходимо максимизировать y_2.

Учитываем третье ограничение $t \geq \frac{46y_2-5}{69}$. Максимальное y_2 определяется из:

$$\frac{1+7y_2}{48} \leq \frac{2-16y_2}{14} \quad \Rightarrow \quad y_2 \leq \frac{82}{722} \approx 0.1136.$$

Итог:

$y_2^{\max} \approx 0.1136.$

Из уравнения $3y_1 + 8y_2 = 1$:

$$t = \frac{1}{3x_1 + 8x_2} \quad \Rightarrow \quad x_1 = \frac{y_1}{t}, \quad x_2 = \frac{y_2}{t}.$$

Подставляем программные значения:

$$t = 0.0374, \quad y_1 = 0.0304, \quad y_2 = 0.1135.$$

Проверка:

$$3y_1 + 8y_2 = 3 \cdot 0.0304 + 8 \cdot 0.1135 \approx 0.0912 + 0.908 = 1 \quad \Rightarrow \quad t = 0.0374.$$

Восстановление исходных переменных

$$x_1 = \frac{y_1}{t} \approx \frac{0.0304}{0.0374} \approx 0.8148,$$
$$x_2 = \frac{y_2}{t} \approx \frac{0.1135}{0.0374} \approx 3.037,$$
$$x_3 = \frac{y_3}{t} = \frac{0}{0.0374} = 0,$$
$$x_4 = \frac{y_4}{t} = \frac{0}{0.0374} = 0,$$
$$x_5 = \frac{y_5}{t} \approx \frac{0.9355}{0.0374} \approx 25.$$

$$Z = \frac{-7 \cdot 0.8148 + 3 \cdot 3.037}{3 \cdot 0.8148 + 8 \cdot 3.037} \approx \frac{-5.7036 + 9.111}{2.4444 + 24.296} \approx \frac{3.4074}{26.74} \approx 0.127.$$

Итоговый ответ

Оптимальное решение:

$$x_1 \approx 0.8148, \quad x_2 \approx 3.037, \quad x_3 = 0, \quad x_4 = 0, \quad x_5 = 25, \quad Z \approx 0.127.$$

Листинг программы:

```
# Импортируем необходимую библиотеку PuLP для работы с задачами линейного
программирования
from pulp import *

def solve_dlp():
    """
    Решает задачу дробно-линейного программирования (ДЛП) методом сведения к
линейной (ЛП).
    Возвращает оптимальные значения переменных и значение целевой функции.
    """
    # Создаем объект задачи максимизации
    prob = LpProblem("DLP_Problem", LpMaximize)

    # Вводим переменные:
    # y1, y2, y3, y4, y5 - преобразованные переменные (y_i = x_i * t)
    # t - обратная величина знаменателя целевой функции (t = 1 / (3x1 + 8x2))
    y = {i: LpVariable(f'y{i}', lowBound=0) for i in range(1, 6)}  # y_i >= 0
    t = LpVariable('t', lowBound=0)  # t >= 0

    # Целевая функция: Z = -7y1 + 3y2 (после преобразований)
    prob += -7 * y[1] + 3 * y[2], "Целевая функция"
```

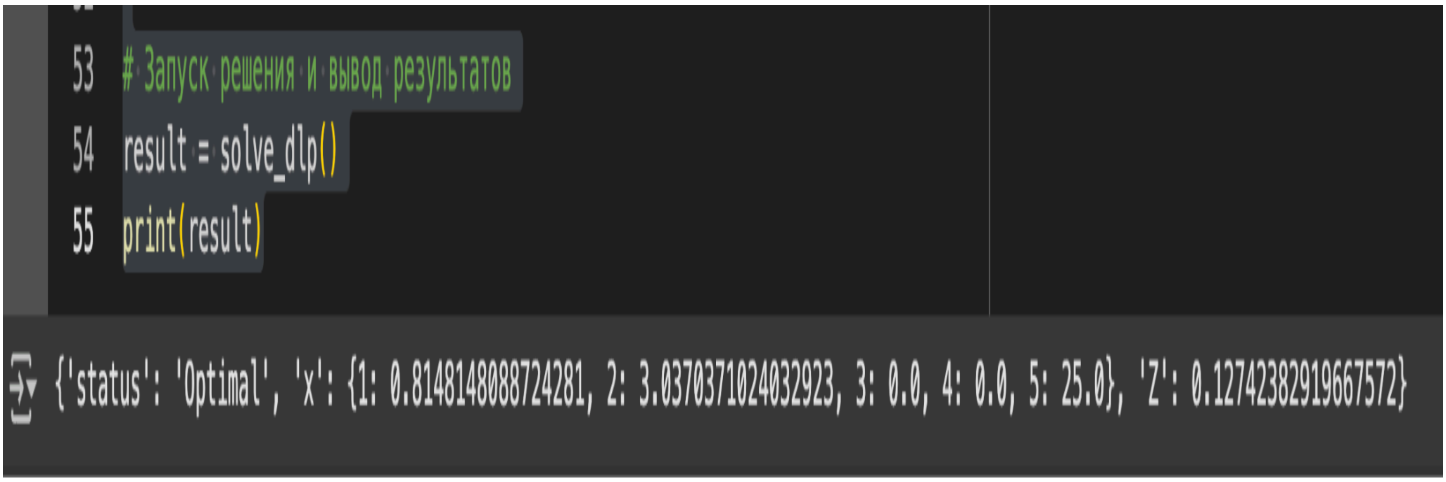
```
    # Добавляем ограничения:
    # 1. y1 + 5y2 + y3 = 16t
    prob += y[1] + 5 * y[2] + y[3] == 16 * t, "Первое ограничение"
    # 2. 6y1 + 3y2 - y4 = 14t
    prob += 6 * y[1] + 3 * y[2] - y[4] == 14 * t, "Второе ограничение"
    # 3. 5y1 - 2y2 + y5 = 23t
    prob += 5 * y[1] - 2 * y[2] + y[5] == 23 * t, "Третье ограничение"
    # 4. 3y1 + 8y2 = 1 (условие для знаменателя)
    prob += 3 * y[1] + 8 * y[2] == 1, "Уравнение знаменателя"

    # Решаем задачу с помощью встроенного солвера CBC
    prob.solve(PULP_CBC_CMD(msg=0))  # msg=0 отключает вывод лога

    # Анализ результатов
    if LpStatus[prob.status] == 'Optimal':
        t_value = t.value()
        # Проверка, что t не слишком близко к нулю (во избежание деления на ноль)
        if t_value < 1e-8:
            return {'status': 'Infeasible', 'message': 't близко к нулю'}

        # Восстанавливаем исходные переменные: x_i = y_i / t
        x = {i: y[i].value() / t_value for i in range(1, 6)}
        # Проверка неотрицательности x_i (с учетом погрешности)
        if all(v >= -1e-6 for v in x.values()):
            # Вычисляем значение исходной целевой функции
            Z = (-7 * x[1] + 3 * x[2]) / (3 * x[1] + 8 * x[2])
            return {'status': 'Optimal', 'x': x, 'Z': Z}
        else:
            return {'status': 'Infeasible', 'message': 'Отрицательные x_i'}
    else:
        return {'status': LpStatus[prob.status]}

# Запуск решения и вывод результатов
result = solve_dlp()
print(result)
```



```
# Запуск решения и вывод результатов
result = solve_dlp()
print(result)
```

```
{'status': 'Optimal', 'x': {1: 0.8148148088724281, 2: 3.0370371024032923, 3: 0.0, 4: 0.0, 5: 25.0}, 'Z': 0.12742382919667572}
```

Вывод: В ходе выполнения лабораторной работы были освоены методы сведения задачи дробно-линейного программирования (ДЛП) к задаче линейного программирования (ЛП) с помощью введения новых переменных. Проведённые задания по подготовке к работе позволили изучить алгоритм преобразования целевой функции и ограничений, а также разработать программу, реализующую данный метод. Реализованный алгоритм продемонстрировал свою эффективность: решение совпало с результатами ручного расчёта.