

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»

Вариант 23

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Куртова Лилиана Николаевна

Вирченко Юрий Петрович

Белгород 2023 г.

Тема: Задачи дробно-линейного программирования

Цель работы: освоить метод сведения задачи ДЛП к задаче ЛП с помощью введения новых переменных. Изучить алгоритм решения задачи ДЛП.

Ход работы:

$$z = \frac{-5x_1 + 7x_2}{4x_1 + 9x_2} \rightarrow \max;$$
$$\begin{cases} 2x_1 + 8x_2 + x_3 = 21, \\ 5x_1 + 3x_2 - x_4 = 17, \\ 6x_1 - 2x_2 + x_5 = 25, \\ x_i \geq 0 \ (i = \overline{1,5}) \end{cases}$$

1. Изучить постановку задачи ДЛП, а также подходы к ее решению
2. Ознакомиться с введением новых переменных, при 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.

$$Z = \frac{-5x_1 + 7x_2}{4x_1 - 3x_2} \rightarrow \max$$
$$\begin{cases} 3x_1 + 8x_2 + x_3 = 18 \\ -5x_1 - 3x_2 + x_4 = -17 \\ 7x_1 - 2x_2 + x_5 = 26 \\ x_i \geq 0, \ (i = \overline{1,5}) \end{cases}$$

Введем новые переменные $y_0 = 1/(4x_1 - 3x_2)$, $y_i = y_0 * x_i$, $(i = 1,5)$. Получим задачу линейного программирования:

$$Z = -5y_1 + 7y_2 \rightarrow \max$$
$$\begin{cases} 3y_1 + 8y_2 + y_3 - 18y_0 = 0 \\ -5y_1 - 3y_2 + y_4 + 17y_0 = 0 \\ 7y_1 - 2y_2 + y_5 - 26y_0 = 0 \\ 4y_1 - 3y_2 = 1 \\ y_i \geq 0, \ (i = \overline{0,5}) \end{cases}$$

Построим М-задачу при $M = 100$:

$$Z_M = -5y_1 + 7y_2 - 100u \rightarrow \max$$

$$\begin{cases} 3y_1 + 8y_2 + y_3 - 18y_0 = 0 \\ -5y_1 - 3y_2 + y_4 + 17y_0 = 0 \\ 7y_1 - 2y_2 + y_5 - 26y_0 = 0 \\ y_1 - 3y_2 + u = 1 \\ y_i \geq 0, (i = \overline{0,5}) \end{cases}$$

Исключим из целевой функции базисную переменную и составим первую симплекс-таблицу:

$$u = 1 - y_1 + 3y_2$$

$$Z_M = -5y_1 + 7y_2 - 100 + 400y_1 - 300y_2$$

$$Z_M = 395y_1 - 293y_2 - 100$$

$$Z_M - 395y_1 + 293y_2 = -100$$

Таблица 1

Б	С	$y_1 \downarrow$	y_2	y_3	y_4	y_5	y_0	и	Отн
$\leftarrow y_3$	0	3	8	1	0	0	-18	0	0
y_4	0	-5	-3	0	1	0	17	0	0
y_5	0	7	2	0	0	1	-26	0	0
и	1	4	-3	0	0	0	0	1	1/4
z	-100	-395	293	0	0	0	0	0	

Таблица 2

Б	С	y_1	y_2	y_3	y_4	y_5	$y_0 \downarrow$	и	Отн
y_1	0	1	2 2/3	1/3	0	0	-6	0	0
y_4	0	0	10 1/3	1 2/3	1	0	-13	0	0
$\leftarrow y_5$	0	0	-16 2/3	-2 1/3	0	1	16	0	0
и	1	0	-13 2/3	-1 1/3	0	0	24	1	1/24
z	-100	0	1346 1/3	131 2/3	0	0	-2370	0	

Таблица 3

Б	С	y1	y2↓	y3	y4	y5	y0	u	Отн
y1	0	1	-3 7/12	- 13/24	0	3/8	0	0	0
y4	0	0	-3 5/24	- 11/48	1	13/16	0	0	0
y0	0	0	-1 1/24	- 7/48	0	1/16	1	0	0
←u	1	0	11 1/3	2 1/6	0	-1 1/2	0	1	3/34
z	-100	0	-1122 5/12	-213 23/24	0	148 1/8	0	0	

Таблица 4

Б	С	y1	y2	y3	y4	y5↓	y0	u	Отн
y1	43/136	1	0	39/272	0	- 27/272	0	43/136	-3 5/27
←y4	77/272	0	0	209/544	1	211/544	0	77/272	154/211
y0	25/272	0	0	29/544	0	- 41/544	1	25/272	-1 9/41
y2	3/34	0	1	13/68	0	- 9/68	0	3/34	- 2/3
z	- 131/136	0	0	169/272	0	- 117/272	0	99 5/136	

Таблица 5

Б	С	y1	y2	y3	y4	y5	y0	u
y1	82/211	1	0	51/211	54/211	0	0	82/211
y5	154/211	0	0	209/211	2 122/211	1	0	154/211
y0	31/211	0	0	27/211	41/211	0	1	31/211
y2	39/211	0	1	68/211	72/211	0	0	39/211
z	- 137/211	0	0	1 10/211	1 23/211	0	0	99 74/211

Решение М-задачи:

$$Z_{M_{\max}} = -\frac{137}{211}, \quad y_0 = \frac{31}{211}, y_1 = \frac{82}{211}, y_2 = \frac{39}{211}, y_3 = 0, y_4 = 0, y_5 = \frac{154}{211}_{\text{нр}}$$

Решение исходной задачи:

$$Z_{\max} = -\frac{137}{211}, \quad x_i = \frac{y_i}{y_0} \quad (i = \overline{1,5})$$

$$x_1 = \frac{82}{31}, x_2 = \frac{39}{31}, x_3 = 0, x_4 = 0, x_5 = \frac{154}{31}_{\text{нр}}$$

Код программы:

```
std::vector<Fraction> getNewFuncForPenalty(
    std::vector<Fraction> &functionVal,
    std::vector<int> &basics) {
    auto &tmp = *this;
    std::vector<Fraction> funcVal(_nCols, 0);
    Fraction sum = 0;
    for (int i = 0; i < _nRows - 1; i++) {
        sum += tmp[i][_nCols - 1];
    }
    Fraction penalty = sum * 10;
    funcVal[0] = sum * penalty;
```

```

for (int j = 0; j < _nCols - 1; j++) {
    if (std::find(
        basics.begin(),
        basics.end(),
        j) != basics.end()) {
        continue;
    }
    Fraction sumOfCol = 0;
    for (int i = 0; i < _nRows - 1; i++) {
        sumOfCol += tmp[i][j];
    }
    funcVal[j + 1] = sumOfCol * penalty * (-1) -
        functionVal[j + 1];
    funcVal[j + 1] *= (-1);
}
return funcVal;
}

//возвращает вектор, в котором хранится ответ
std::vector <Fraction> getPenaltyMethod(
    std::vector <Fraction> &functionVal) {
    Matrix &tmp = *this;
    Matrix m(_nRows, _nCols + _nRows - 1);
    for (int i = 0; i < _nRows - 1; i++) {
        for (int j = 0; j < _nCols - 1; j++) {
            m[i][j] = tmp[i][j];
        }
    }
    std::vector<int> basics;
    for (int i = 0; i < _nRows - 1; i++) {
        m[i][i + _nCols - 1] = 1;
        basics.push_back(i + _nCols - 1);
    }
    m.setCol(getCol(_nCols - 1), m._nCols - 1);
    auto newFunc = m.getNewFuncForPenalty(functionVal, basics);
    std::vector <Fraction> res = m.getSimplexMethod(newFunc);
    res.erase(res.begin() + res.size() - _nRows + 1, res.end());
    return res;
}

int main() {
    std::vector <Fraction> data = {{2, 8, 1, 0, 0, -21, 0},
        {5, 3, 0, -1, 0, -17},
        {6, -2, 0, 0, 1, -25}};
    std::vector <Fraction> functionVal = {-5, 7, 0, 0, 0, 4, -3, 0, 0, 0};
    Matrix m(data, 5, 6);
    std::vector <Fraction> res1 = m.getPenaltyMethod(functionVal);
    std::cout << "z = " << res1[0] << "\n" << "y0 = " << res1[res1.
        size() - 1] << ", ";
    for (int i = 1; i < res1.size() - 1; i++) {
        std::cout << "y" << i << " = " << res1[i];
        if (i != res1.size() - 2) {
            std::cout << ", " << "\n";
        } else {
            std::cout << ")";
        }
    }
    std::cout << "\n Reverse substitution :\n";
    for (int i = 1; i < res1.size() - 1; i++) {
        Fraction curX = res1[i] / res1[res1.size() - 1];
        std::cout << "x" << i << " = " << curX;
        if (i != res1.size() - 2) {
            std::cout << ", " << "\n";
        }
    }
}

```

```
} else {  
    std::cout << " )";  
}  
}  
}
```

Результат работы программы:

```
z = -137/211  
(y0 = 31/211, y1 = 82/211,  
y2 = 39/121,  
y3 = 0,  
y4 = 0,  
y5 = 154/211  
Reverse substitution:  
x1 = 82/31,  
x2 = 39/31,  
x3 = 0,  
x4 = 0,  
x5 = 154/31  
  
Process finished with exit code 0
```

Вывод: Освоил метод сведения задачи ДЛП к задаче ЛП с помощью введения новых переменных. Изучил алгоритм решения задачи ДЛП