

Федеральное агентство по образованию РФ
Государственное образовательное учреждение высшего профессионального
образования
Белгородский Государственный Технологический Университет им В. Г. Шухова

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем.

Лабораторная работа № 8
Задача дробно-линейного программирования

Выполнил:
Студент группы ВТ-212
Алёшин Артём Алексеевич

Приняли:
Куртова Л.Н.
Вирченко Ю.П.

Содержание отчета

1. Выполнение работы. Аналитическое решение
2. Программная реализация.
3. Листинг программы
4. Результат работы программы
5. Вывод

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.

Вариант 2.

$$z = \frac{9x_1 + 3x_2 + 2x_3 + x_4}{x_1 + x_2 + x_3 + x_4} \rightarrow \max;$$

$$\begin{cases} 4x_1 + x_2 + x_3 + 3x_4 \leq 250, \\ x_1 + 3x_3 + x_4 \leq 80, \\ x_1 + 3x_2 + x_3 \leq 340, \end{cases}$$

$$x_i \geq 0 \ (i = \overline{1, 4})$$

Аналитическое решение

Первым действием перейдем от неравенств в уравнениях к равенствам.

$$\begin{cases} 4x_1 + 1x_2 + x_3 + 3x_4 + x_5 = 250 \\ x_1 + 3x_3 + x_4 + x_6 = 80 \\ x_1 + 3x_2 + x_3 + x_7 = 340 \end{cases}$$

Чтобы перейти от задачи ДЛП к задаче ЛП, введем дополнительные переменные:

$$y_0 = \frac{1}{x_1 + x_2 + x_3 + x_4}$$

$$y_1 = x_1 y_0;$$

$$y_2 = x_2 y_0;$$

$$y_3 = x_3 y_0;$$

$$y_4 = x_4 y_0;$$

$$y_5 = x_5 y_0;$$

Пересчитаем целевую функцию:

$$\begin{aligned} z &= \frac{9x_1 + 3x_2 + 2x_3 + x_4}{x_1 + x_2 + x_3 + x_4} = (9x_1 + 3x_2 + 2x_3 + x_4) * \frac{1}{x_1 + x_2 + x_3 + x_4} = (9x_1 + 3x_2 + 2x_3 + x_4) * y_0 \\ &= 9x_1 y_0 + 3x_2 y_0 + 2x_3 y_0 + x_4 y_0 = 9y_1 + 3y_2 + 2y_3 + y_4 \end{aligned}$$

Из формулы y_0 следует:

$$y_0 = \frac{1}{x_1 + x_2 + x_3 + x_4}$$

$$(x_1 + x_2 + x_3 + x_4) * y_0 = 1$$

$$x_1 y_0 + x_2 y_0 + x_3 y_0 + x_4 y_0 = 1$$

$y_1 + y_2 + y_3 + y_4 = 1 \Rightarrow$ получили новое уравнение в нашу систему.

Новая система:

$$\begin{aligned} z &= 9y_1 + 3y_2 + 2y_3 + y_4 \rightarrow \max \\ \begin{cases} 4y_1 + 1y_2 + y_3 + 3y_4 + y_5 = 250 \\ y_1 + 3y_3 + y_4 + y_6 = 80 \\ y_1 + 3y_2 + y_3 + y_7 = 340 \\ y_1 + y_2 + y_3 + y_4 = 1 \end{cases} \end{aligned}$$

Заметим, что теперь не хватает одной базисной переменной. Чтобы перейти к базисному виду, используем *метод штрафов*.

Добавим недостающую переменную:

$$\begin{cases} 4y_1 + 1y_2 + y_3 + 3y_4 + y_5 = 250 \\ y_1 + 3y_3 + y_4 + y_6 = 80 \\ y_1 + 3y_2 + y_3 + y_7 = 340 \\ y_1 + y_2 + y_3 + y_4 + u = 1 \end{cases}$$

Теперь пересчитаем целевую функцию:

$$\begin{aligned} z &= 9y_1 + 3y_2 + 2y_3 + y_4 - M * u = 9y_1 + 3y_2 + 2y_3 + y_4 - 100 * (1 - y_1 - y_2 - y_3 - y_4) \\ &= -100 + 109y_1 + 103y_2 + 102y_3 + 101y_4 \end{aligned}$$

Нашли минимальное значение в последней строке, все большинство отношений свободных коэффициентов к элементам столбца равны нулю, берем первое отношение.

Б	С	y1	y2	y3	y4	y5	y6	y7	y0	u		
y5	0	4	1	1	3	1	0	0	-250	0	0	
y6	0	1	0	3	1	0	1	0	-80	0	0	
y7	0	1	3	1	0	0	0	1	-340	0	0	
u	1	1	1	1	1	0	0	0	0	1	1	
z	-100	-109	-103	-102	-101	0	0	0	0	0		

Минимальное значение находится в столбце y_0 , в нем содержится только один положительный элемент, поэтому выводим из базиса u .

Б	С	y1	y2	y3	y4	y5	y6	y7	y0	u		
y1	0	1	1/4	1/4	3/4	1/4	0	0	-62 1/2	0		
y6	0	0	- 1/4	2 3/4	1/4	- 1/4	1	0	-17 1/2	0		
y7	0	0	2 3/4	3/4	- 3/4	- 1/4	0	1	-277 1/2	0		
u	1	0	3/4	3/4	1/4	- 1/4	0	0	62 1/2	1		
z	-100	0	-75 3/4	-74 3/4	-19 1/4	27 1/4	0	0	-6812 1/2	0		

Симплекс-метод завершен.

Б	С	y1	y2	y3	y4	y5	y6	y7	y0	u
y1	1	1	1	1	1	0	0	0	0	1
y6	7/25	0	- 1/25	2 24/25	8/25	- 8/25	1	0	0	7/25
y7	4 11/25	0	6 2/25	4 2/25	9/25	-1 9/25	0	1	0	4 11/25
y0	2/125	0	3/250	3/250	1/250	- 1/250	0	0	1	2/125
z	9	0	6	7	8	0	0	0	0	109

Получили:

$$z = 9$$

$$y_0 = \frac{2}{125}; y_1 = 1; y_2 = 0; y_3 = 0; y_4 = 0; y_5 = 0; y_6 = \frac{7}{25}; y_7 = \frac{111}{125}$$

Ответ:

$$z = 9$$

$$x_1 = \frac{y_1}{y_0} = \frac{125}{2}; x_2 = \frac{y_2}{y_0} = 0; x_3 = \frac{y_3}{y_0} = 0; x_4 = \frac{y_4}{y_0} = 0; x_5 = \frac{y_5}{y_0} = 0;$$

$$x_6 = \frac{y_6}{y_0} = \frac{35}{2}; x_7 = \frac{y_7}{y_0} = \frac{555}{2}$$

Программная реализация

```
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 = {4, 1, 1, 3, 1, 0, 0, -250, 0,
                                     3, 1, 0, 3, 1, 0, 1, 0, -80, 0,
                                     4, 1, 3, 1, 0, 0, 0, 1, -340, 0,
                                     5, 1, 1, 1, 1, 0, 0, 0, 0, 1};

    std::vector< Fraction > functionVal = { 0, 9, 3, 2, 1, 0, 0, 0, 0 };

    Matrix m(data, 5, 9);

    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 << "\nReverse 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 << ")";
        }
    }
}

```

```
}  
}  
}
```

```
z = 9  
(y0 = 2/125, y1 = 1,  
y2 = 0,  
y3 = 0,  
y4 = 0,  
y5 = 0,  
y6 = 7/25,  
y7 = 111/25)  
Reverse substitution:  
x1 = 125/2,  
x2 = 0,  
x3 = 0,  
x4 = 0,  
x5 = 0,  
x6 = 35/2,  
x7 = 555/2)  
Process finished with exit code 0
```

Вывод: изучили решение задач ДЛП, написали программу. Ответ сошелся с ручным подсчетом.