

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 №3

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: Модификации симплекс метода. Методы искусственного базиса и больших штрафов

Цель работы: изучение методов искусственного базиса и больших штрафов решения задач ЛП в канонической форме, не подготовленных к работе симплекс-методом в чистом виде.

3.

$$\begin{aligned} z &= x_1 - 3x_2 + 4x_3 + 5x_4 - x_5 + 8x_6 \rightarrow \max; \\ \begin{cases} x_1 + 5x_2 - 3x_3 - 4x_4 + 2x_5 + x_6 = 14, \\ 2x_1 + 9x_2 - 5x_3 - 7x_4 + 4x_5 + 2x_6 = 30, \\ x_i \geq 0 \ (i = \overline{1, 6}). \end{cases} \end{aligned}$$

1. Изучить метод и алгоритм искусственного базиса и составить программу решения задачи ЛП этим методом.
2. Изучить метод и алгоритм больших штрафов и составить программу решения задачи ЛП этим методом.

Метод искусственного базиса

Код библиотеки:

```
#include "simplexTable.h"

SimplexTable SimplexTable::findArtificialBasis() {
    SimplexTable processedTable;
    this->copy(processedTable);
    //Делаем свободные члены положительными
    for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
        if (processedTable.matrix.getData(row, processedTable.matrix.getColumns() - 1) < 0) {
            processedTable.matrix.multiplyRowBy(row, -1);
        }
    }
    //Определяем значения коэффициентов целевой ф-ции подзадачи
    for (int column = 0; column < processedTable.matrix.getColumns(); column++) {
        float sum = 0;
        for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
            sum += processedTable.matrix.getData(row, column);
        }
        processedTable.matrix.setData(processedTable.matrix.getRows() - 1, column, -sum);
    }
    //Вводим искусственные переменные
    for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
        processedTable.matrix.pushBackZeroColumn();
        processedTable.matrix.setData(row, processedTable.matrix.getColumns() - 1, 1);
    }
    //Применяем симплекс метод
    bool hasSolution = true;
    bool isSolution = false;
    while (hasSolution && !isSolution) {
        processedTable = processedTable.simplexMethodStep(hasSolution, isSolution);
    }
    //Приводим к подходящему для решения исходной задачи виду
    processedTable.matrix.deleteRow(processedTable.matrix.getRows() - 1);
    for (int i = 0; i < processedTable.matrix.getRows(); i++) {
        processedTable.matrix.deleteColumn(processedTable.matrix.getColumns() - 1);
    }
}
```

```

}
vector<float> targetRow(this->matrix.getRow(this->matrix.getRows() - 1));
processedTable.matrix.addNewRow(targetRow);
return processedTable;
}

```

Код основной программы:

```

#include <iostream>
#include <vector>
#include "simplexTable.h"
int main() {
    SimplexTable simplexTable;
    simplexTable.inputTable(2, 6,
        {{1, 5, -3, -4, 2, 1, 14},
         {2, 9, -5, -7, 4, 2, 30}
        },
        {1, -3, 4, 5, -1, 8, 0});
    SimplexTable newTable = simplexTable.findArtificialBasis();
    float maxFValue;
    vector<float> solution = newTable.findSolutionBySimplexMethod(maxFValue);
    if (solution.empty())
        cout << "Решений нет";
    else {
        cout << "{";
        for (auto i: solution) {
            cout << i << ", ";
        }
        cout << "\b\b}\n";
        cout << "Fmax = " << maxFValue;
    }
}

```

Результат работы программы:

```

C:\B6TU\B6TU\IsOp\Lab3\Code\cmake-build-debug\Code.exe
{0; 0; 0; 22; 0; 2}
Fmax = 186
Process finished with exit code 0

```

Метод больших штрафов

Код библиотеки:

```

vector<float> SimplexTable::findSolutionBySimplexBigPenalty(float &maxFValue) {
    SimplexTable processedTable;
    this->copy(processedTable);
    //Делаем свободные члены положительными
    for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
        if (processedTable.matrix.getData(row, processedTable.matrix.getColumns() - 1)
            < 0) {
            processedTable.matrix.multiplyRowBy(row, -1);
        }
    }
    float penalty = 10000;
    //Вводим искусственные переменные
    for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
        processedTable.matrix.pushBackZeroColumn();
        processedTable.matrix.setData(row, processedTable.matrix.getColumns() - 1, 1);
    }
}

```

```

processedTable.matrix.setData(processedTable.matrix.getRows() - 1,
                               processedTable.matrix.getColumns() - 1,
                               penalty);
processedTable.matrix.subtractMultipliedRow(processedTable.matrix.getRows() -
                                             1,
                                             row, penalty);
}
for (int row = 0; row < processedTable.matrix.getRows() - 1; row++) {
    processedTable.matrix.deleteColumn(processedTable.matrix.getColumns() - 1);
}
return processedTable.findSolutionBySimplexMethod(maxFValue);
}

```

Код основной программы:

```

#include <iostream>
#include "libs/matrix/matrix.h"
#include "libs/simplexTable/simplexTable.h"
int main() {
    SimplexTable simplexTable;
    simplexTable.inputTable(2, 6,
                           {{1, 5, -3, -4, 2, 1, 14},
                            {2, 9, -5, -7, 4, 2, 30}},
                           {1, -3, 4, 5, -1, 8, 0});
    float maxFValue;
    vector<float> solution = simplexTable.findSolutionBySimplexBigPenalty(maxFValue);
    if (solution.empty())
        cout << "Решений нет";
    else {
        cout << "{";
        for (auto i: solution) {
            cout << i << ", ";
        }
        cout << "\b\b}\n";
        cout << "Fmax = " << maxFValue;
    }
}

```

Результат работы программы:

```

C:\BGTU\BGTU\IsOp\Lab3\Code\cmake-build-debug\Code.exe
{0; 0; 0; 22; 0; 2}
Fmax = 186.007
Process finished with exit code 0

```

Задание 3. Запрограммировать изученные алгоритмы и отладить соответствующие программы. В рамках подготовки тестовых данных решить вручную одну из следующих ниже задач.

Нахождение искусственного базиса:

	<u>b</u>	x1	x2	x3	x4	x5	x6	y1	y2		<u>b / x</u>
<--y1	14	1	5	-3	-4	2	1	1	0		2,8
y2	30	2	9	-5	-7	4	2	0	1		3,33333
F	-44	-3	-14	8	11	-6	-3	0	0		
	<u>b</u>	x1	x2	x3	x4	x5	x6	y1	y2		
x2	2,8	0,2	1	-0,6	-0,8	0,4	0,2	0,2	0		
<--y2	4,8	0,2	0	0,4	0,2	0,4	0,2	-1,8	1		
F	-4,8	-0,2	0	-0,4	-0,2	-0,4	-0,2	2,8	0		
	<u>b</u>	x1	x2	x3	x4	x5	x6	y1	y2		
x2	10	0,5	1	0	-0,5	1	0,5	-2,5	1,5		
x3	12	0,5	0	1	0,5	1	0,5	-4,5	2,5		
F	0	0	0	0	1	0	0	1	1		

Решение исходной задачи

	<u>b</u>	x1	x2	x3	x4	x5	x6		<u>b / x</u>
x2	10	0,5	1	0	-0,5	1	0,5		2
x3	12	0,5	0	1	0,5	1	0,5		2
Z	0	-1	3	-4	-5	1	-8		
	<u>b</u>	x1	x2	x3	x4	x5	x6		
x6	20	1	2	0	-1	2	1		
x3	2	0	-1	1	1	0	0		
Z	160	7	19	-4	-13	17	0		
	<u>b</u>	x1	x2	x3	x4	x5	x6		
x6	22	1	1	1	0	2	1		
x4	2	0	-1	1	1	0	0		
Z	186	7	6	9	0	17	0		

Метод больших штрафов

Приведение к М-задаче											
	b	x1	x2	x3	x4	x5	x6	y1	y2		
y1	14	1	5	-3	-4	2	1	1	0		
y2	30	2	9	-5	-7	4	2	0	1		
F	0	-1	3	-4	-5	1	-8	1000	1000		
	b	x1	x2	x3	x4	x5	x6	y1	y2		
y1	14	1	5	-3	-4	2	1	1	0		
y2	30	2	9	-5	-7	4	2	0	1		
F	-14000	-1001	-4997	2996	3995	-1999	-1008	0	1000		
	b	x1	x2	x3	x4	x5	x6	y1	y2		
y1	14	1	5	-3	-4	2	1	1	0		
y2	30	2	9	-5	-7	4	2	0	1		
F	-44000	-3001	-13997	7996	10995	-5999	-3008	0	0		
Решение М-задачи											
	b	x1	x2	x3	x4	x5	x6	y1	y2	b / x	
<--y1	14	1	5	-3	-4	2	1	1	0	2,8	
y2	30	2	9	-5	-7	4	2	0	1	3,33333333	
F	-44000	-3001	-13997	7996	10995	-5999	-3008	0	0		
	b	x1	x2	x3	x4	x5	x6	y1	y2		
x2	2,8	0,2	1	-0,6	-0,8	0,4	0,2	0,2	0		
<--y2	4,8	0,2	0	0,4	0,2	0,4	0,2	-1,8	1		
F	-4808,4	-201,6	0	-402,2	-202,6	-400,2	-208,6	2799,4	0		
	b	x1	x2	x3	x4	x5	x6	y1	y2	b / x	
<--x2	10	0,5	1	0	-0,5	1	0,5	-2,5	1,5	20	
x3	12	0,5	0	1	0,5	1	0,5	-4,5	2,5	24	
F	18	-0,5	0	0	-1,5	2	-7,5	989,5	1005,5		
	b	x1	x2	x3	x4	x5	x6	y1	y2		
x6	20	1	2	0	-1	2	1	-5	3		
<--x3	2	0	-1	1	1	0	0	-2	1		
F	168	7	15	0	-9	17	0	952	1028		
	b	x1	x2	x3	x4	x5	x6				
x6	22	1	1	1	0	2	1	-7	4		
x4	2	0	-1	1	1	0	0	-2	1		
F	186	7	6	9	0	17	0	934	1037		

Вывод: в ходе лабораторной работы мы изучили методы искусственного базиса и больших штрафов решения задач ЛП в канонической форме, не подготовленных к работе симплекс-методом в чистом виде.