

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 № 2

по дисциплине: Исследование операций

тема: «Симплекс-метод в чистом виде»

Выполнил: ст. группы ПВ-223

Игнатъев Артур Олегович

Проверил:

проф. Вирченко Юрий Петрович

Белгород 2024г.

Лабораторная работа №2

«Симплекс-метод в чистом виде»

Цель работы: изучение симплекс-метода для решения задачи линейного программирования с использованием симплекс-таблиц, получение навыков кодирования изученного алгоритма, отладки и тестирования соответствующих программ.

Вариант 3

$$z = 7x_2 + 8x_4 + x_6 \rightarrow \max;$$

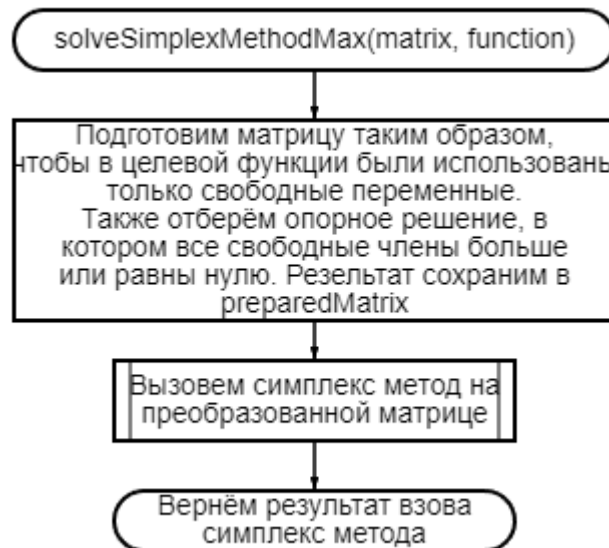
$$\begin{cases} x_1 - 3x_2 - 4x_4 - 2x_6 = 10 \\ 5x_2 + 5x_4 + x_5 + x_6 = 26 \\ 4x_2 + x_3 - 6x_4 - 3x_6 = 12 \end{cases}$$

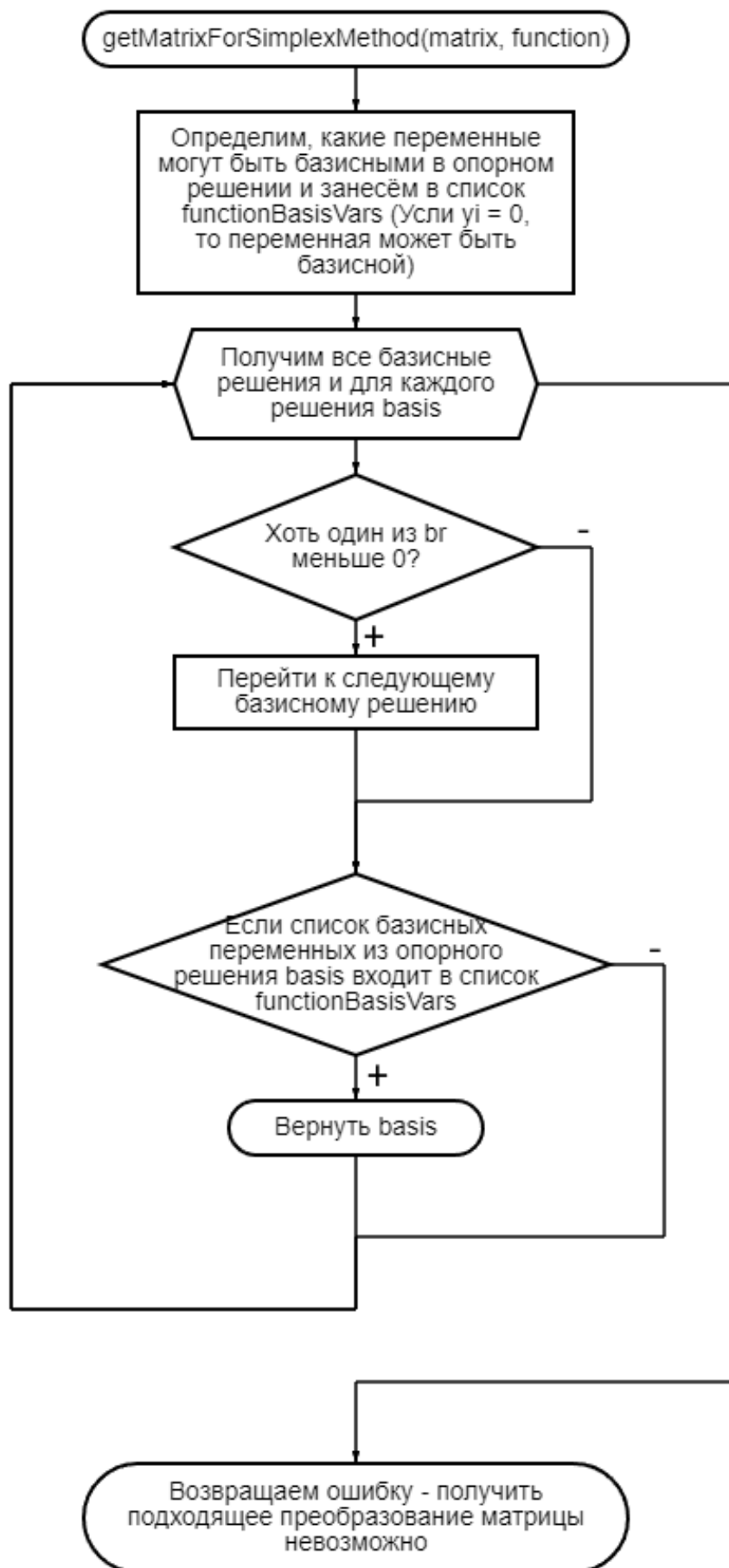
$$x_i \geq 0 (i = \overline{1,6})$$

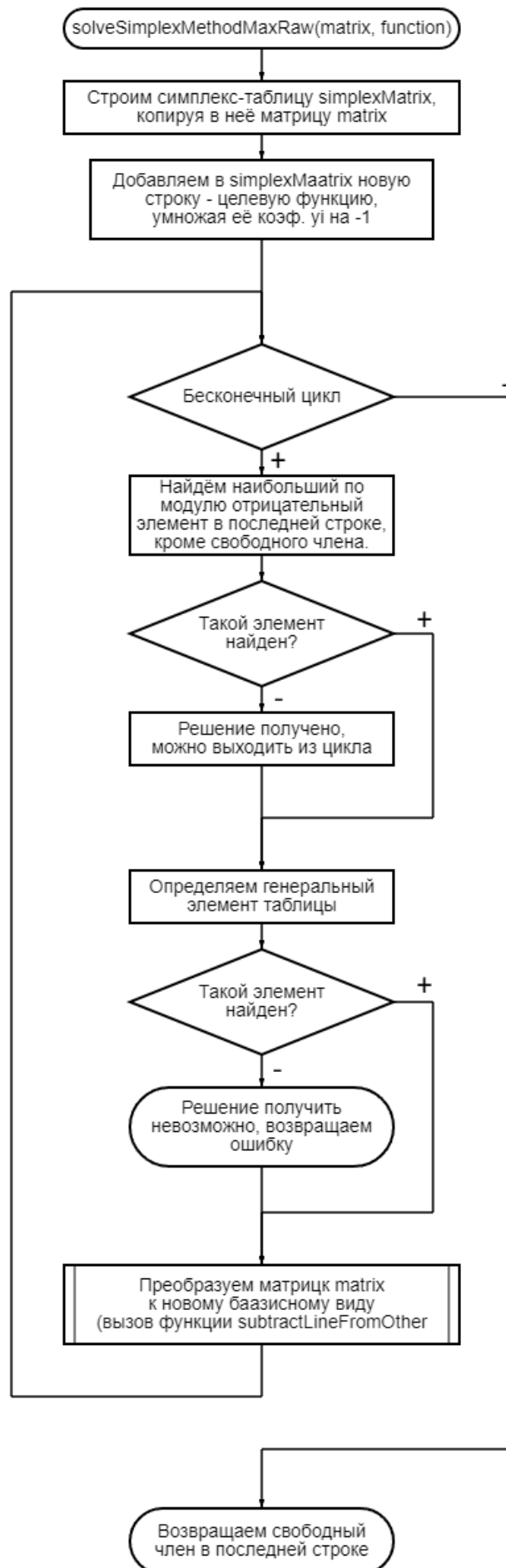
Ход выполнения лабораторной работы:

Запрограммировать и отладить изученный алгоритм.

Блок-схемы







Исходный код программы:

Файл task1.hpp

```
#ifndef OPERATIONS_RESEARCH_TASK1_HPP
#define OPERATIONS_RESEARCH_TASK1_HPP

#pragma once

#include <optional>
#include <vector>
#include <algorithm>
#include <iomanip>

#include "../lab1/task1.hpp"

template <std::size_t T>
auto getMatrixForSimplexMethod(std::vector<std::array<double, T>>& matrix,
std::array<double, T>& function) {
    // Определим, какие переменные могут быть базисными в опорном решении и
    // занесём их в список functionBasisVars
    // (Если yi = 0, то переменная может быть базисной)
    std::vector<int> functionBasisVars;
    for (int i = 0; i < function.size() - 1; i++)
        if (std::abs(function[i]) < EPS)
            functionBasisVars.push_back(i);

    // Получим все базисные решения и для каждого решения basis
    for (auto& basis : getAllBasises(matrix)) {
        bool isAllBsMoreOrEqualToZero = true;
        for (int i = 0; i < basis.matrix.size() && isAllBsMoreOrEqualToZero;
i++) {
            if (basis.matrix[i].back() < EPS)
                isAllBsMoreOrEqualToZero = false;
        }

        // Хотя один из br меньше 0?
        if (!isAllBsMoreOrEqualToZero) {
            // Перейти к следующему базисному решению
            continue;
        }

        // Если список базисных переменных из опорного решения basis входит в
        // список
        // functionBasisVars
        std::sort(basis.indices.begin(), basis.indices.end());
        if (std::includes(basis.indices.begin(), basis.indices.end(),
functionBasisVars.begin(), functionBasisVars.end()))
            // Возвращаем искомый базис
            return basis;
        }

        // Возвращаем ошибку - получить подходящее преобразование матрицы невоз-
        // можно
        throw std::invalid_argument("No basis for function found");
    }
}

template <std::size_t T>
double solveSimplexMethodMaxRow(std::vector<std::array<double, T>>& matrix,
std::array<double, T>& function, std::vector<int>* baseIndices = nullptr) {
    // Строим симплекс-таблицу, копируя в неё матрицу matrix
    std::vector<std::array<double, T>> simplexMatrix(matrix);
    // Добавляем новую строку - целевую функцию, умножая её коэф. yi на -1
```

```

simplexMatrix.push_back(function);
for (int i = 0; i < T; i++)
    simplexMatrix.back()[i] *= -1;

// Бесконечный цикл
while (true) {
    for (int i = 0; i < simplexMatrix.size(); i++) {
        for (int j = 0; j < simplexMatrix[i].size(); j++) {
            std::cout << std::setw(10) << simplexMatrix[i][j];
        }

        std::cout << std::endl;
    }
    std::cout << std::endl;

    // Найдём наибольший по модулю отрицательный элемент в последней
    строке, кроме свободного члена.
    int minColumnIndex = -1;
    for (int i = 0; i < T - 1; i++) {
        if (simplexMatrix.back()[i] < 0 && (minColumnIndex == -1 || simplexMatrix.back()[minColumnIndex] > simplexMatrix.back()[i]))
            minColumnIndex = i;
    }

    // Такой элемент найден?
    if (minColumnIndex == -1) {
        // Решение получено, можно выходить из цикла
        break;
    }

    // Определим генеральный элемент таблицы
    int minRowIndex = -1;
    for (int i = 0; i < simplexMatrix.size() - 1; i++) {
        if (simplexMatrix[i][minColumnIndex] <= EPS) continue;
        if (minRowIndex == -1) minRowIndex = i;
        else if (simplexMatrix[minRowIndex].back() / simplexMatrix[minRowIndex][minColumnIndex] >
                 simplexMatrix[i].back() / simplexMatrix[i][minColumnIndex]) minRowIndex = i;
    }

    // Такой элемент найден?
    if (minRowIndex == -1) {
        // Решение получить невозможно, возвращаем ошибку
        throw std::invalid_argument("No solution");
    }

    if (baseIndices)
        (*baseIndices)[minRowIndex] = minColumnIndex;

    // Преобразуем матрицу к новому базисному виду
    subtractLineFromOther(simplexMatrix, minRowIndex, minColumnIndex);
}

for (int i = 0; i < simplexMatrix.size(); i++) {
    for (int j = 0; j < simplexMatrix[i].size(); j++) {
        std::cout << std::setw(10) << simplexMatrix[i][j];
    }

    std::cout << std::endl;
}
std::cout << std::endl;

for (int i = 0; i < matrix.size(); i++)

```

```

        for (int j = 0; j < matrix[i].size(); j++)
            matrix[i][j] = simplexMatrix[i][j];

        // Возвращаем свободный член в последней строке
        std::cout << "Zmax: ";
        return {simplexMatrix.back().back()};
    }

template <std::size_t T>
double solveSimplexMethodMax(std::vector<std::array<double, T>>& matrix,
std::array<double, T>& function) {
    // Подготовим матрицу таким образом, чтобы в целевой функции были исполь-
зованы только свободные переменные.
    // Также отберём опорное решение, в котором все свободные члены больше
или равны нулю
    auto preparedMatrix = getMatrixForSimplexMethod(matrix, function);

    // Вызовем симплекс метод на преобразованной матрице
    return solveSimplexMethodMaxRaw(preparedMatrix.matrix, function);
}

#endif //OPERATIONS_RESEARCH_TASK1_TPP

```

Файл task1.cpp

```

#include <iostream>
#include <iomanip>
#include <array>

#include "../libs/alg/alg.h"

int main() {
    // Инициализируем матрицу и функцию
    std::vector<std::array<double, 7>> matrix = {
        {1, -3, 0, -4, 0, -2, 10},
        {0, 5, 0, 5, 1, 1, 26},
        {0, 4, 1, -6, 0, -3, 12}};
    std::array<double, 7> function{{0.0, 7.0, 0.0, 8.0, 0.0, 1.0, 0.0}};

    // Вывод ответа
    std::cout << "Zmax: " << solveSimplexMethodMax(matrix, function);
}

```

Результат работы программы:

```
Zmax: 41.6
```

Дополнительно проведём тестирование при помощи программы отбора базисных решений (см. лабораторная работа №1 задание 2).

Результат выполнения программы:

```
=====
Обнаружено опорное решение: {62; 0; 90; 0; 0; 26; 0}
```

```
Значение функции z(B): 26
=====
```

```
Обнаружено опорное решение: {10; 0; 12; 0; 26; 0; 0}
```

```
Значение функции z(B): 0
=====
```

```
Обнаружено опорное решение: {30.8; 0; 43.2; 5.2; 0; 0; 0}
```

```
Значение функции z(B): 41.6
=====
```

```
Обнаружено опорное решение: {28.8421; 4.73684; 0; 0; 0; 2.31579; 0}
```

```
Значение функции z(B): 35.4737
=====
```

```
Обнаружено опорное решение: {19; 3; 0; 0; 11; 0; 0}
```

```
Значение функции z(B): 21
=====
```

```
Обнаружено опорное решение: {26.48; 4.32; 0; 0.88; 0; 0; 0}
```

```
Значение функции z(B): 37.28
=====
```

```
Zmax: 41.6
```

```
Оптимальный план: {30.8; 0; 43.2; 5.2; 0; 0; 0}
```

Результаты вычислений:

Имеем целевую функцию $z = 7x_2 + 8x_4 + x_6 \rightarrow \max$. И систему уравнений

$$\begin{cases} x_1 - 3x_2 - 4x_4 - 2x_6 = 10 \\ 5x_2 + 5x_4 + x_5 + x_6 = 26 \\ 4x_2 + x_3 - 6x_4 - 3x_6 = 12 \end{cases}$$

Приведём систему уравнений к базисному виду. Базисными переменными выберем x_1, x_3 и x_6 .

Привели систему к базисному виду

$$\begin{cases} x_1 = 10 + 3x_2 + 4x_4 + 2x_6 \\ x_3 = 12 - 4x_2 + 6x_4 + 3x_6 \\ x_6 = 26 - 5x_2 - 5x_4 - x_5 \end{cases}$$

Построим симплекс- таблицу

Баз. Пер.	Св. чл.	x_1	x_2	x_3	x_4	x_5	x_6
x_1	10	1	-3	0	-4	0	-2
x_3	12	0	4	1	-6	0	-3
x_6	26	0	5	0	5	1	1
z	0	0	-7	0	-8	0	-1

Баз. Пер.	Св. чл.	x_1	x_2	x_3	x_4	x_5	x_6
x_1	$30\frac{4}{5}$	1	1	0	0	$\frac{4}{5}$	$-1\frac{1}{5}$
x_4	$43\frac{1}{5}$	0	10	1	0	$1\frac{1}{5}$	$-1\frac{4}{5}$
x_5	$5\frac{1}{5}$	0	1	0	1	$\frac{1}{5}$	$\frac{1}{5}$
z	$41\frac{3}{5}$	0	1	0	0	$1\frac{3}{5}$	$\frac{3}{5}$

Ответ: $z_{\max} = 41,6$. Координаты точки максимума: $x_1 = 30\frac{4}{5}, x_2 = 0, x_3 = 43\frac{1}{5}, x_4 = 5\frac{1}{5}, x_5 = 0, x_6 = 0$. Результаты ручных вычислений совпали с результатами выполнения программы.

Вывод: в ходе лабораторной работы разработали и отладили программу, находящую оптимальное решение в системе линейных уравнений для целевой функции, и использующей симплекс метод.