**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Методы оптимизации»**

Тема: Симплексный метод

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Преподаватель |  | Мальцева Н.В. |

Санкт-Петербург

2024

## Цель работы.

1. Решение задачи линейного программирования симплекс методом с помощью стандартной программы.
2. Решение задачи линейного программирования графически.
3. Сравнение результатов решения задачи обоими способами.

## Задание.

Рассматривается следующая задача линейного программирования.

Найти минимум линейной функции *f(x1,x2,...,xn):*

*f = c[1]\*x[1] + c[2]\*x[2] +...+ c[n]\*x[n] ,*

где *c[i]* - постоянные коэффициенты, на множестве, заданном набором линейных ограничений:

*a[1,1]\*x[1] + ... + a[1,n]\*x[n] >= b[1]*

*...*

*a[m,1]\*x[1] + ... + a[m,n]\*x[n] >= b[m]*

*x[1]>=0,...,x[n]>=0 ,*

где *a[i,j],b[i]* - постоянные коэффициенты .

В матричной форме ограничения записываются следующим образом:

*AX>=B, X>=0.*

Целевая функция может быть представлена в виде скалярногопроизведения:

*f = (C, X).*

**Основные теоретические положения.**

Симплексный метод решения задачи линейного программирования состоит из двух этапов:

1) поиск крайней точки допустимого множества,

2) поиск оптимальной точки путем направленного перебора крайних точек.

Крайняя точка не существует, если в таблице существует строка, все элементы которой неположительные, а последний элемент – отрицательный.

Крайняя точка н а й д е н а, ели все элементы вектора-столбца B больше нуля.

Чтобы найти крайнюю точку, надо:

1) выбрать строку i, в которой b[i] < 0;

2) выбрать столбец s, в котором a[i, s] >= 0;

3) в столбце s задать номер строки r разрешающего элемента так, чтобы отрицательное отношение b[r]/a[r,s] было максимальным .

4) поменять местами имена координат в таблице из строки r и столбца s;

5) рассматривая элемент a[r,s] как разрешающий, необходимо преобразовать таблицу по формулам :

ARS:= a[r,s];

z1[r,s]:= 1/ARS;

z1[r,j]:= -z[r,j]/ARS , j<>s;

z1[i,s]:= z[i,s]/ARS , i<>r;

z1[i,j]:= (z[i,j]\*ARS - z[i,s]\*z[r,j])/ARS , i<>r,j<>s;

z:=z1,

где под z и z1 понимается соответственно первоначальное и преобразованное значение таблицы (кроме левого столбца и верхней строки).

Оптимальная точка н а й д е н а, если все элементы вектор-строки С >= 0 (при этом все элементы вектор-столбца B >= 0 ).

Оптимальная точка н е существует, если в таблице есть столбец j, в котором c[j] < 0 , а все a[i,j]>0 при любом i .

Чтобы найти оптимальную точку, надо:

1) выбрать столбец s, в котором c[s] < 0;

2) в столбце s задать номер строки r разрешающего элемента так, чтобы отрицательное отношение b[r]/a[r,s] было максимальным ;

3) поменять местами имена координат в таблице из строки r и столбца s;

4) рассматривая элемент a[r,s] как разрешающий, необходимо преобразовать таблицу по формулам (смотри выше).

Координаты оптимальной точки определяются следующим образом:

1) если x[j] находится на i-м месте левого столбца, то его значение равно b[i];

2) если x[i] находится на j-м месте верхней строки, то его значение равно 0.

## Выполнение работы.

*Вариант 21.*

* Графическое решение задачи.

Для графического решения поставленной задачи построим допустимое множество, описанное выше, и обозначим его X. Также отобразим линию уровня целевой функции, например, , а также направление её антиградиента.

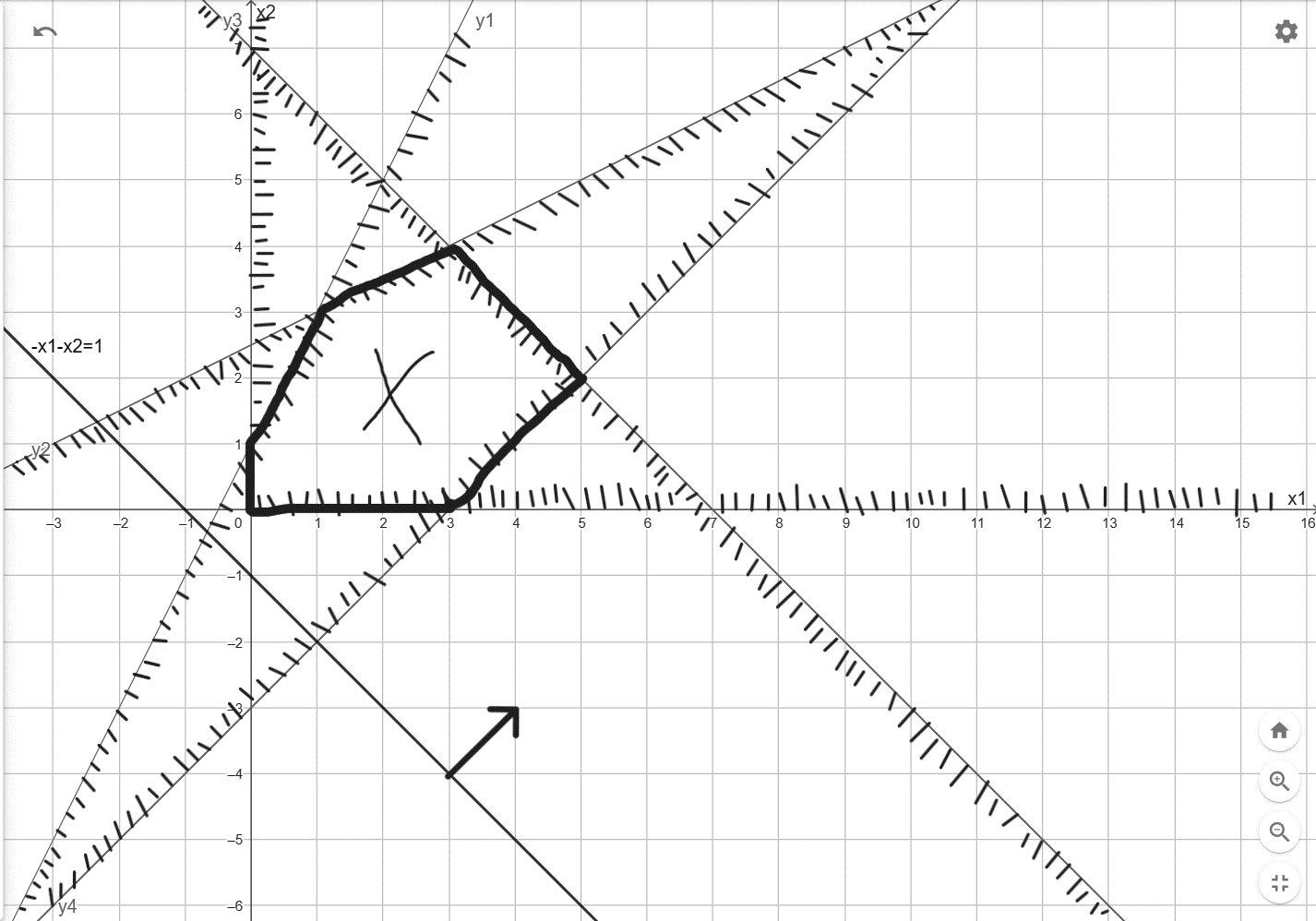


Рисунок 1. Графическое решение задачи

Заметим, что при движении линии уровня функции вдоль направления антиградиента мы придём в отрезок с концами {x1 = 3, x2 = 4} и {x1 = 5, x2 = 2}. На данном отрезке и достигается минимум целевой функции равный -7.

Далее в ходе программного решения будут добавлены рисунки, на которых будут отмечены шаги работы программы.

* Программное решение задачи.

**Первая ветка решения.**

*Первый шаг.*



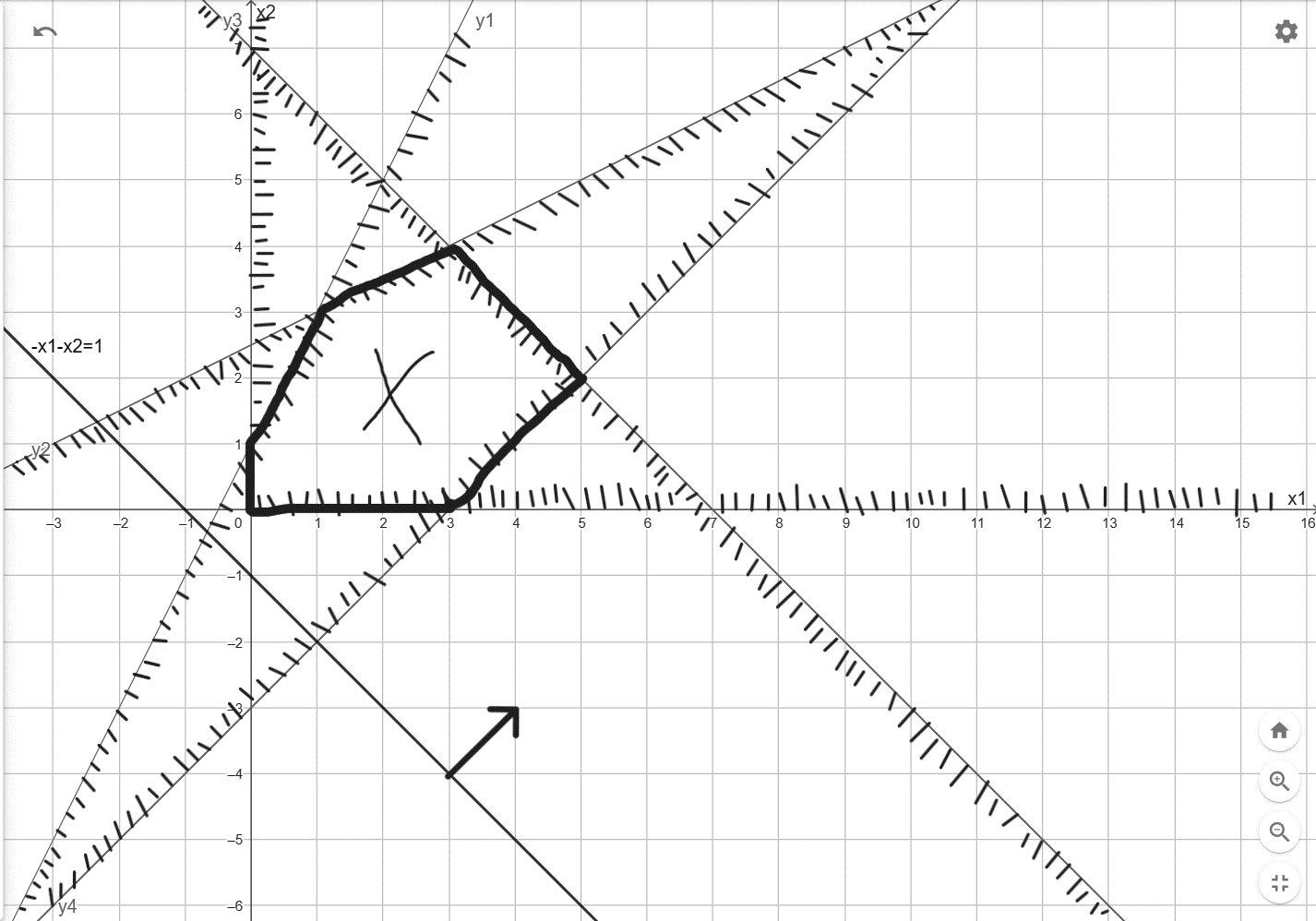


Рисунок 2. Первая ветка решения, первый шаг

Если *xi* находится на *j*-ом месте верхней строки, то его значение равно 0. Поэтому на данном шаге алгоритма рассматривается точка (0, 0).

Так как все элементы вектора-столбца B больше нуля => точка (0, 0) является крайней.

Заметим, что существует (). Из этого можно сделать вывод, что точка (0, 0) не является оптимальной.

Также существуют

( для ).

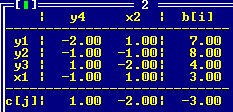
Значит, выбираем разрешающий элемент. Для этого выберем в начале столбец, так как и отрицательные, то можно выбрать любой из этих двух столбцов. В данной ветке решения возьмем столбец *r = 1.*

При определении разрешающего руководствуемся следующим правилом выбора:

.

Среди полученных отношений максимальным отрицательным является . Таким образом столбец разрешающего элемента 1, строка разрешающего элемента 4.

*Второй шаг.*



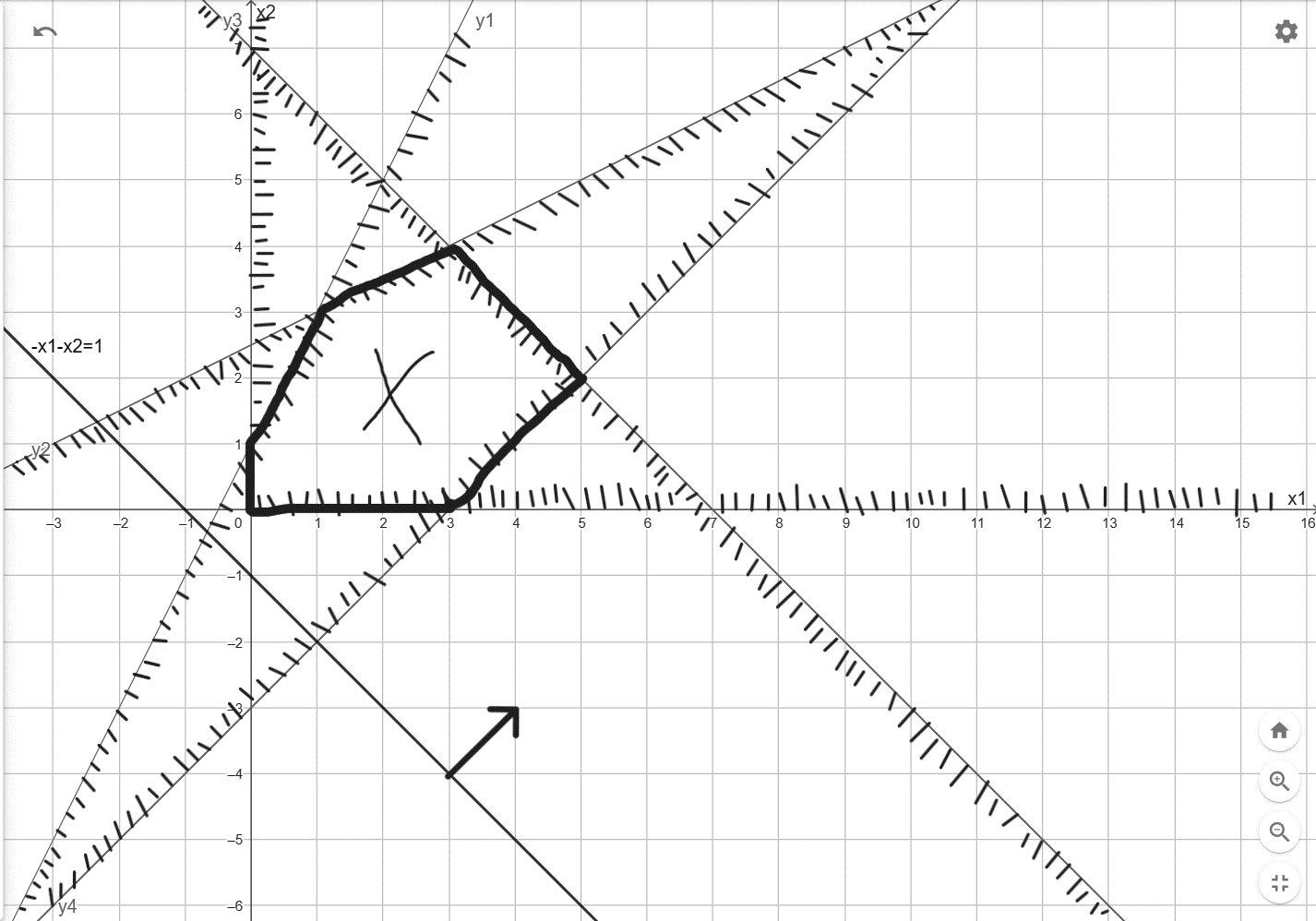


Рисунок 3. Первая ветка решения, второй шаг

Если *xi* находится на *j*-ом месте верхней строки, то его значение равно 0. Если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*. Поэтому на данном шаге алгоритма рассматривается точка (3, 0).

Так как все элементы вектора-столбца B больше нуля и точка получена из крайней точки предыдущего шага (первого) => точка (3, 0) является крайней.

Заметим, что существует (). Из этого можно сделать вывод, что точка (3, 0) не является оптимальной.

Также существуют

().

Значит, выбираем разрешающий элемент. Для этого выберем в начале столбец, так как отрицательное и во *2* столбце существуют отрицательные *а*, то выбираем столбец *r = 2.*

При определении разрешающего руководствуемся следующим правилом выбора:

.

Среди полученных отношений максимальным отрицательным является . Таким образом столбец разрешающего элемента 2, строка разрешающего элемента 3.

*Третий шаг.*



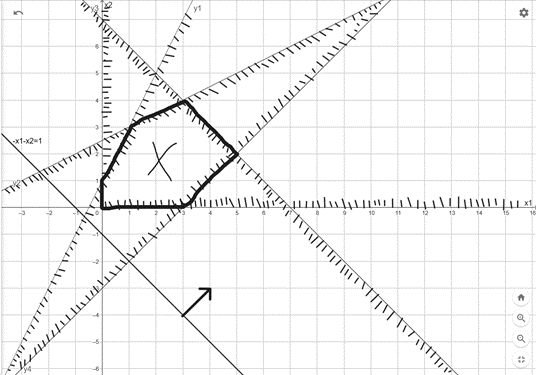


Рисунок 4. Первая ветка решения, третий шаг

Если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*. Поэтому на данном шаге алгоритма рассматривается точка (5, 2).

Так как все элементы вектора-столбца B больше нуля и точка получена из крайней точки предыдущего шага (второго) => точка (5, 2) является крайней.

Заметим, что существует ∀*j* *cj* ≥ 0 (). Из этого можно сделать вывод, что оптимальная точка найдена и это точка (5, 2). Оптимальная крайняя точка найдена за 3 шага. Значение функции в данной точке -7.

**Вторая ветка решения (отличается выбором столбца на первом шаге).**

*Первый шаг.*



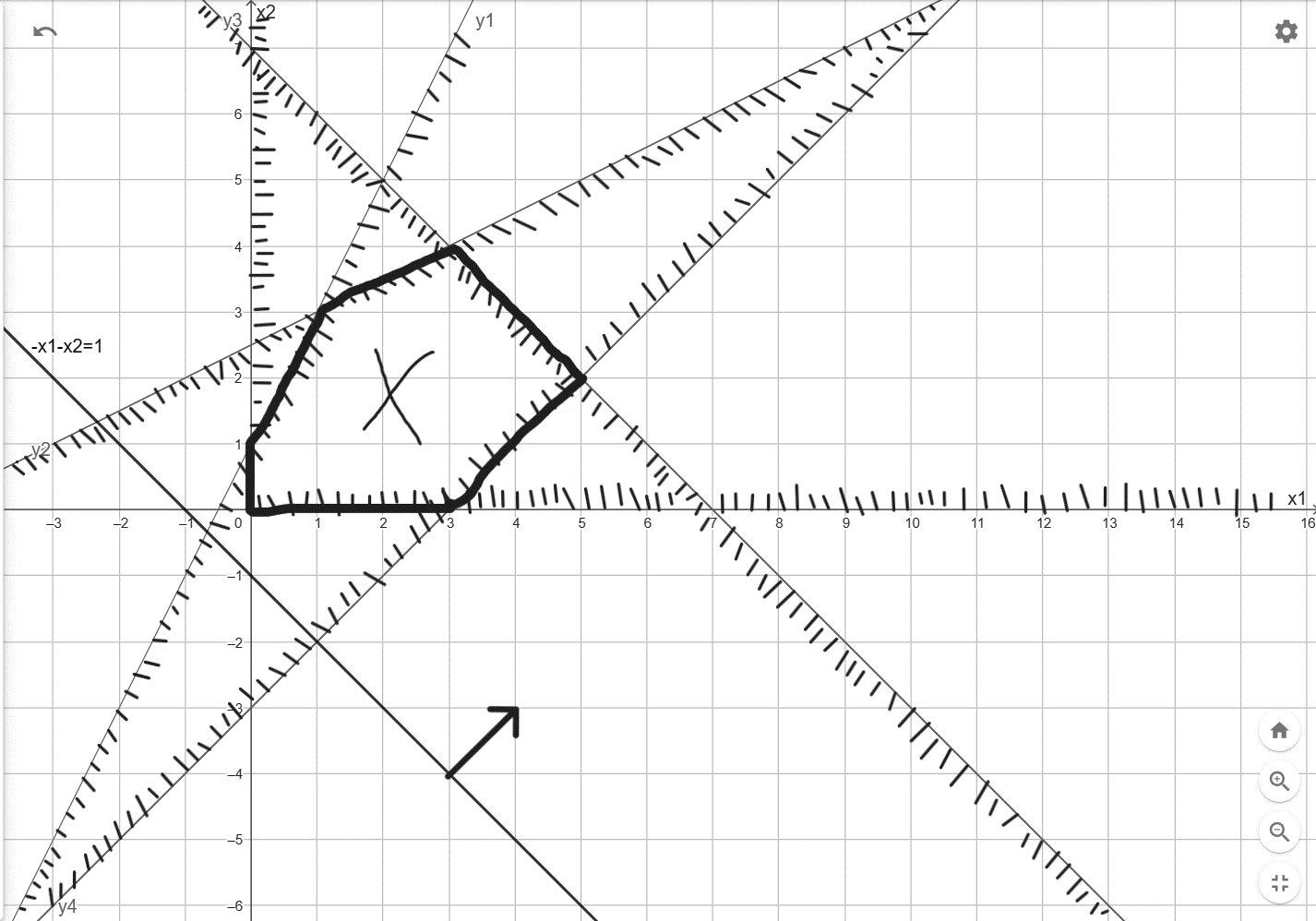


Рисунок 5. Вторая ветка решения, первый шаг

Если *xi* находится на *j*-ом месте верхней строки, то его значение равно 0. Поэтому на данном шаге алгоритма рассматривается точка (0, 0).

Так как все элементы вектора-столбца B больше нуля => точка (0, 0) является крайней.

Заметим, что существует (). Из этого можно сделать вывод, что точка (0, 0) не является оптимальной.

Также существуют

( для ).

Значит, выбираем разрешающий элемент. Для этого выберем в начале столбец, так как и отрицательные, то можно выбрать любой из этих двух столбцов. В данной ветке решения возьмем столбец *r = 2.*

При определении разрешающего руководствуемся следующим правилом выбора:

.

Среди полученных отношений максимальным отрицательным является . Таким образом столбец разрешающего элемента 2, строка разрешающего элемента 1.

*Второй шаг.*



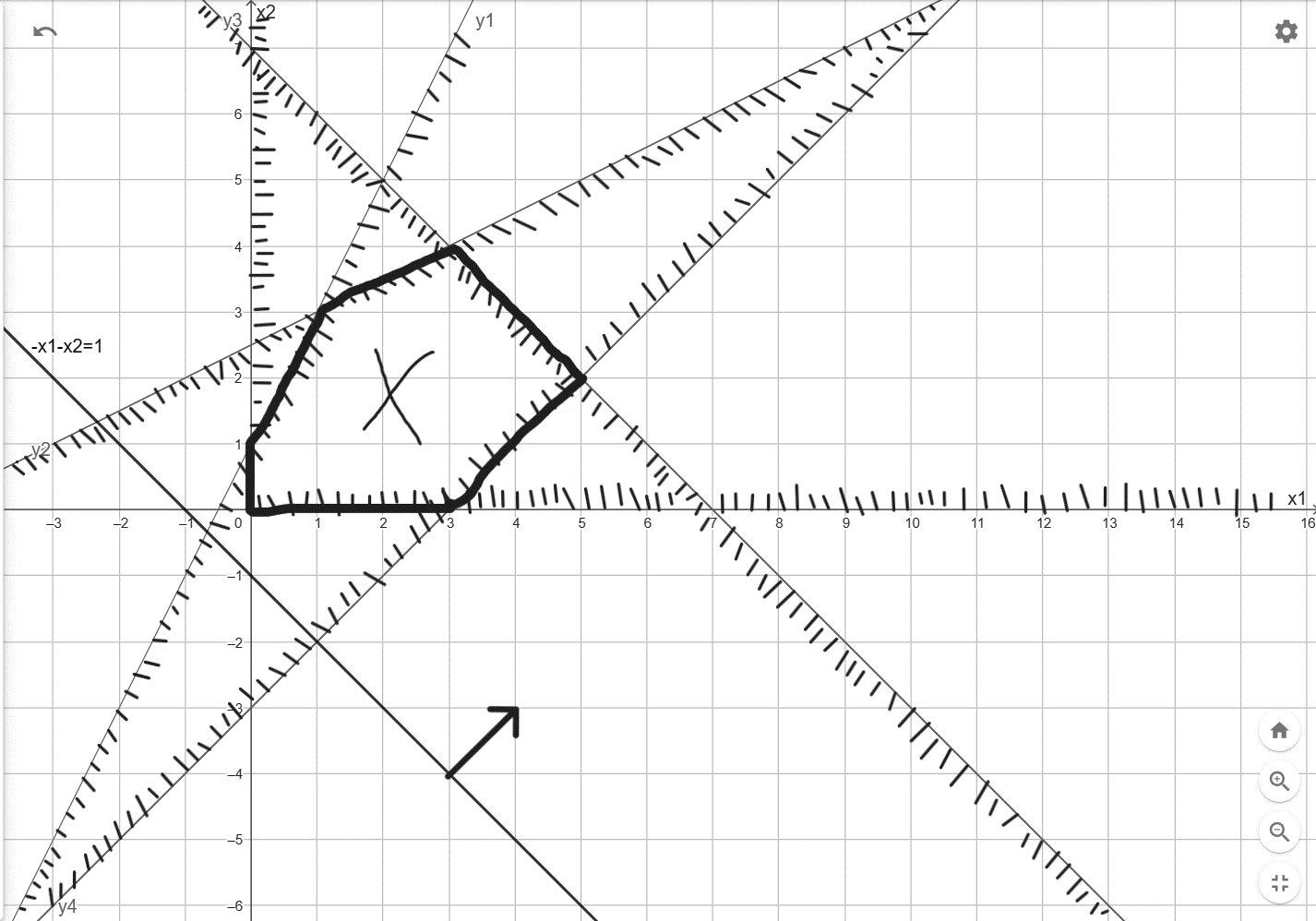


Рисунок 6. Вторая ветка решения, второй шаг

Если *xi* находится на *j*-ом месте верхней строки, то его значение равно 0. Если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*. Поэтому на данном шаге алгоритма рассматривается точка (0, 1).

Так как все элементы вектора-столбца B больше нуля и точка получена из крайней точки предыдущего шага (первого) => точка (0, 1) является крайней.

Заметим, что существует (). Из этого можно сделать вывод, что точка (0, 1) не является оптимальной.

Также существуют

().

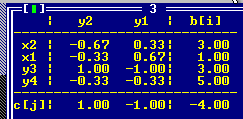
Значит, выбираем разрешающий элемент. Для этого выберем в начале столбец, так как отрицательное и в *1* столбце существуют отрицательные *а*, то выбираем столбец *r = 1.*

При определении разрешающего руководствуемся следующим правилом выбора:

.

Среди полученных отношений максимальным отрицательным является . Таким образом столбец разрешающего элемента 1, строка разрешающего элемента 2.

*Третий шаг.*



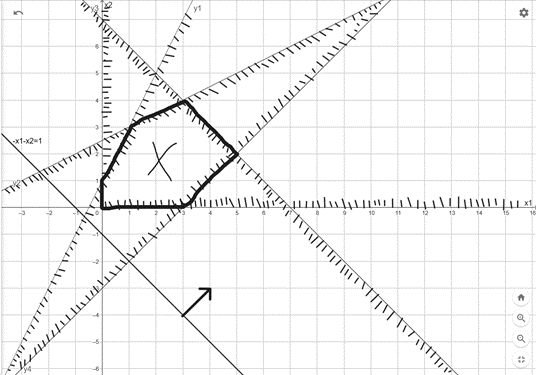


Рисунок 7. Вторая ветка решения, третий шаг

Если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*. Поэтому на данном шаге алгоритма рассматривается точка (1, 3).

Так как все элементы вектора-столбца B больше нуля и точка получена из крайней точки предыдущего шага (второго) => точка (1, 3) является крайней.

Заметим, что существует (). Из этого можно сделать вывод, что точка (1, 3) не является оптимальной.

Также существуют

().

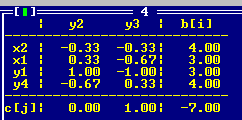
Значит, выбираем разрешающий элемент. Для этого выберем в начале столбец, так как отрицательное и во *2* столбце существуют отрицательные *а*, то выбираем столбец *r = 2.*

При определении разрешающего руководствуемся следующим правилом выбора:

.

Среди полученных отношений максимальным отрицательным является . Таким образом столбец разрешающего элемента 2, строка разрешающего элемента 3.

*Четвёртый шаг.*



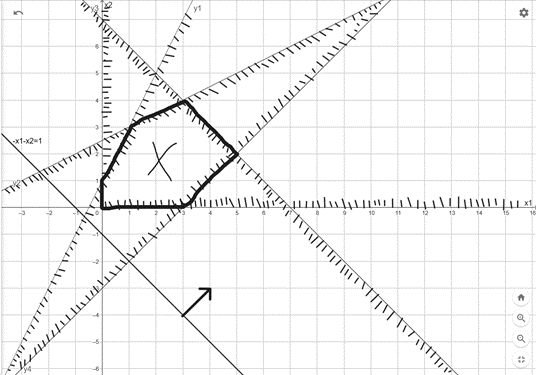


Рисунок 8. Вторая ветка решения, четвёртый шаг

Если *xj* находится на *i*-ом месте левого столбца, то его значение равно *bi*. Поэтому на данном шаге алгоритма рассматривается точка (3, 4).

Так как все элементы вектора-столбца B больше нуля и точка получена из крайней точки предыдущего шага (третьего) => точка (3, 4) является крайней.

Заметим, что существует ∀*j* *cj* ≥ 0 (). Из этого можно сделать вывод, что оптимальная точка найдена и это точка (3, 4). Оптимальная крайняя точка найдена за 4 шага. Значение целевой функции в данной точке -7.

* Сравнение результатов графического и программного решения

По рисункам с графиками, представленными вместе с работой программы, можно наблюдать, как программа переходит из одной крайней точки в другую, пока не будет достигнута оптимальная. Передвижение по точкам осуществлялось таким образом, что значение целевой функции уменьшалось. Направление антиградиента как раз и подсказывало, куда направлено наискорейшее убывание функции.

Заметим, что программное решение можно представить в виде двух веток в зависимости от выбора столбца на первом шаге. Так каждая из веток решения по отдельности в результате получает точки, которые на графическом решении являются лишь концами отрезка, на котором целевая функция достигает своего минимума на допустимом множестве.

То есть, графическое решение и программное не совпадают в точности, так как минимум функции достигается на отрезке, что возможно отследить лишь на графике, а не программно.

## Выводы.

В ходе выполнения лабораторной работы был рассмотрен симплексный метод решения основной задачи линейного программирования. Для целевой функции была решена задача поиска минимума на допустимом множестве. Программный способ решения задачи использует такие шаги симплексного метода: ответ на вопросы найдена ли уже крайняя точка и существует ли она, найдена ли уже оптимальная точка и существует ли она, выбор разрешающего элемента и преобразования таблиц. Графический же метод позволяет наглядно определить множество допустимых значений, а также направление убывания значения функции, продвигая вдоль которого линию уровня функции также можно найти решение задачи. В итоге, графическое решение и программное не совпадают совсем точно, так как графически видно, что функция достигает минимума на отрезке, в то время как программа выдает только точки, где достигается минимум, и при том одну, в зависимости от выбора столбца на первом шаге. В общем, минимальное значение целевой функции на допустимом множестве -7.