**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

|  |  |  |
| --- | --- | --- |
| Студент гр. 1304 |  | Шаврин А.П. |
| 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 (Вариант 24).

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 скалярного произведения: 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 не положительны, а последний элемент - отрицательный.

**Крайняя точка найдена**, если 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; [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 найдена**, если все элементы вектор-строки С >= 0 (при этом все элементы вектор-столбца B >= 0).

**Оптимальная точка не 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 ( см.выше ).

Координаты оптимальной точки определяются следующим образом:

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

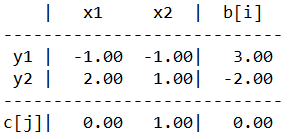
2) если x[i] находится на j-м месте верхней строки, то его значение равно 0.

## Выполнение работы.

1. Формальная постановка задачи
2. Решение задачи с помощью готовой программы и ее протокол

*А) Первая ветка решения*

* Шаг 1.

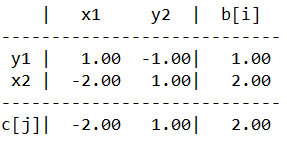


Листинг 1.

Из приведенного листинга, можно видеть, что **начальной точкой** **алгоритма является точка** . Как видно в таблице нет строки, все элементы которой не положительны, а последний элемент – отрицательный => **Крайняя точка существует**. Поскольку не все элементы вектора-столбца B больше нуля => **крайняя точка не найдена**.

Выберем 2ю строку (i = 2) с отрицательным свободным членом, а затем выберем 2й столбец (s = 2) для разрешающего элемента, т.к. в этом столбце есть неотрицательные значения. Выберем 2ю строку (r = 2) для разрешающего элемента, т.к. отрицательное отношение b[r]/a[r,s] является максимальным (в частности b[2] = -2, a[2, 2] = 1, -2/1 = -2 – максимальное отрицательное, другое соотношение для 1й строки будет -3, что тоже отрицательное, но не максимальное).

* Шаг 2.

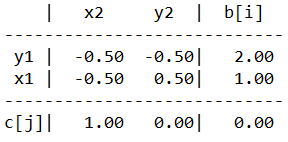


Листинг 2.

Из приведенного листинга, можно видеть, что **текущей точкой** **алгоритма является точка** . Как видно в таблице нет строки, все элементы которой не положительны, а последний элемент – отрицательный => **Крайняя точка существует**. Поскольку все элементы вектора-столбца B больше нуля => **крайняя точка найдена**. Так как не все элементы вектор-строки С >= 0 => **Оптимальная точка не найдена**. **Оптимальная точка существует**, т.к. в таблице есть столбец j=1, в котором c[j] < 0 (c[1] = -2 < 0), но не все a[i,j]>0 при любом i (в частности a[2, 1] = -2 < 0).

Выберем 1й столбец (s=1), в котором c[s] < 0 и выберем 2ю строку (r=2) для разрешающего элемента, т.к. отрицательное отношение b[r]/a[r,s] является максимальным (в частности b[2] = 2, a[2, 1] = -2, 2/-2 = -1 – максимальное отрицательное, другое соотношение для 1й строки будет положительным, что не подходит по условию).

* Шаг 3.

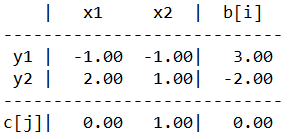


Листинг 3.

Из приведенного листинга, можно видеть, что **текущей точкой** **алгоритма является точка** . Как видно в таблице нет строки, все элементы которой не положительны, а последний элемент – отрицательный => **Крайняя точка существует.** Поскольку все элементы вектора-столбца B больше нуля => **крайняя точка найдена**. Так как все элементы вектор-строки С >= 0 => **Оптимальная точка найдена**.

*Б) Вторая ветка решения (отличается выбором столбца при рассмотрении разрешающего элемента)*

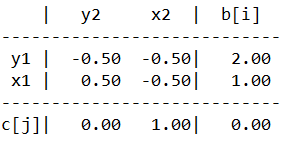
* Шаг 1.



Листинг 4.

Из приведенного листинга, можно видеть, что **начальной точкой** **алгоритма является точка** . Как видно в таблице нет строки, все элементы которой не положительны, а последний элемент – отрицательный => **Крайняя точка существует**. Поскольку не все элементы вектора-столбца B больше нуля => **крайняя точка не найдена**.

Выберем 2ю строку (i = 2) с отрицательным свободным членом, а затем выберем 1й столбец (s = 1) для разрешающего элемента, т.к. в этом столбце есть неотрицательные значения. Выберем 2ю строку (r = 2) для разрешающего элемента, т.к. отрицательное отношение b[r]/a[r,s] является максимальным (в частности b[2] = -2, a[2, 1] = 2, -2/2 = -1 – максимальное отрицательное, другое соотношение для 1й строки будет -3, что тоже отрицательное, но не максимальное).



Листинг 5.

Из приведенного листинга, можно видеть, что **текущей точкой** **алгоритма является точка** . Как видно в таблице нет строки, все элементы которой не положительны, а последний элемент – отрицательный => **Крайняя точка существует.** Поскольку все элементы вектора-столбца B больше нуля => **крайняя точка найдена**. Так как все элементы вектор-строки С >= 0 => **Оптимальная точка найдена**.

Как видно из листинга, результат такой же, как и в первой ветке решения, за исключением положения столбцов => не удалось прийти в другую точку программным способом.

1. Графическое решение задачи с отображением шагов выполнения программы

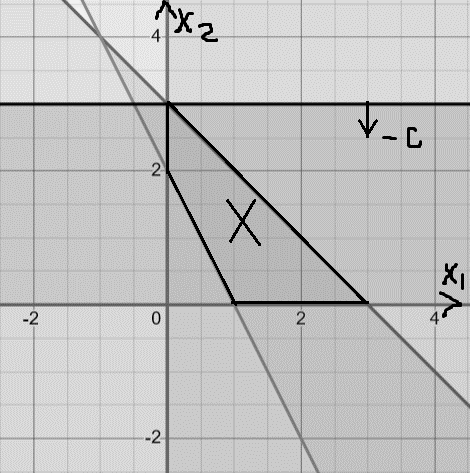


Рисунок 1. Начальные ограничения и линия уровня целевой функции с направлением ее минимизации.

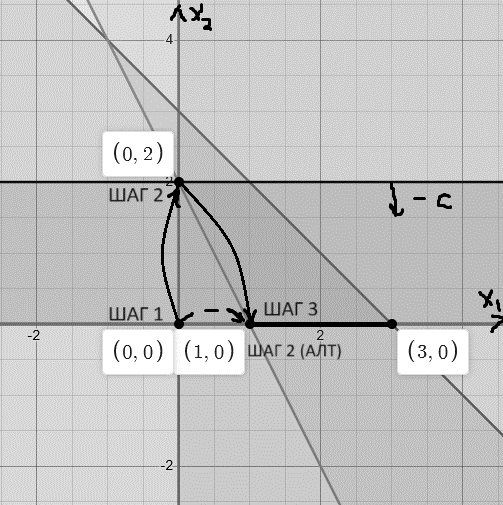


Рисунок 2. Шаги выполнения программы.

На рисунке отображены шаг 1, 2 и 3 для первой ветки решения и шаг 1 и 2 (альтернативный) для второй ветки решения программы.

При графическом решении мы начинаем с точки (0, 3), а не с (0, 0) как это делает программа, затем попадаем в точку (0, 2) а после в отрезок от (1, 0) до (3, 0) являющийся решением задачи.

1. Объяснение полученных результатов

Исходя из графического решения видно, что при движении линии уровня целевой функции в направлении ее минимизации ответом будет отрезок от точки (1, 0) до (3, 0). Таким образом, графическое решение задачи не совпадает с решением задачи с помощью готовой программы. Поскольку программа начинает работу из точки (0, 0) и по всем возможным веткам решений попадает в точку (1, 0), которая является частью верного решения, то она не дает возможности прийти в другую точку и завершает свое выполнение.

## Выводы.

Была изучена и решена задача линейного программирования симплекс методом с помощью стандартной программы, а также графическим методом.

Графическим методом найден ответ в виде отрезка от точки (1, 0) до (3, 0), а симплекс методом получен ответ в виде точки (1, 0).

Благодаря графическому методу мы можем видеть допустимую область и даже не рассматривать не крайние точки, а также произвольно выбирать начальную точку. Также данный подход позволяет визуально отобразить направление минимизации линии целевой функции и получить полный ответ.

Симплекс метод, реализованный с помощью стандартной программы, также приходит к ответу, однако он не всегда дает получить полный ответ из-за фиксированной начальной точки. Также данный метод требует рассмотрения не крайних точек.