Российский государственный университет нефти и газа (НИУ)

имени И.М. Губкина

ОТЧЕТ

о выполнении домашнего задания № 3

по дисциплине «Методы оптимизации»

Выполнил: Хуснутдинов Эдуард Тимурович

Группа: АМ-17-06

Москва, 2020

**Постановка задачи**

Решить задачу линейного программирования симплекс-методом.

**Исходные данные**

A = =

=

Имеем задачу линейного программирования:

maxL=

**Решение**

Предварительно проведём проверку на неотрицательность вектора ограничений , в случае, если < 0, то домножим строку матрицы А на -1.

Далее к задаче применяем метод искусственного базиса, добавляя в уравнения искусственные переменные , полагаем, что они неотрицательны. Приводя систему к базисному виду, необходимо вывести все искусственные переменные. Этому соответствует задача максимизации функции

Выбирая разрешающий столбец будем руководствоваться критериями симплекс-метода, то есть выбирать разрешающим тот столбец, коэффициент для L1 которого будет наименьшим и отрицательным.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| + | x1 | x2 | x3 | x4 | x5 | = |
| u1 | 5 | 8 | 0 | 9 | 0 | 22 |
| u2 | -3 | 0 | 6 | 1 | 8 | 12 |
| u3 | -9 | 3 | 6 | -1 | 4 | 3 |
| L | -5 | 9 | 0 | 0 | -7 | 0 |
| L1 | 7 | -11 | -12 | -9 | -12 | -37 |

Выбираем в качестве разрешающего 3-ий столбец:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| + | x1 | x2 | u3 | x4 | x5 | = |
| u1 | 5 | 8 | 0 | 9 | 0 | 22 |
| u2 | 6 | -3 | -1 | 2 | 4 | 9 |
| x3 | -1,5 | 0,5 | 0,1667 | -0,1667 | 0,6667 | 0,5 |
| L | -5 | 9 | 0 | 0 | -7 | 0 |
| L1 | -11 | -5 | 2 | -11 | -4 | -31 |

Избавляемся от искусственной переменной :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| + | x1 | x2 | x4 | x5 | = |
| u1 | 5 | 8 | 9 | 0 | 22 |
| u2 | 6 | -3 | 2 | 4 | 9 |
| x3 | -1,5 | 0,5 | -0,1667 | 0,6667 | 0,5 |
| L | -5 | 9 | 0 | -7 | 0 |
| L1 | -11 | -5 | -11 | -4 | -31 |

Тем же путем выводим из базиса переменные , получаем:

|  |  |  |  |
| --- | --- | --- | --- |
| + | x4 | x5 | = |
| x2 | 0,698 | -0,317 | 1,381 |
| x1 | 0,682 | 0,507 | 2,190 |
| x3 | 0,507 | 1,587 | 3,095 |
| L | -2,873 | -1,603 | -1,476 |
| L1 | 0 | 0 | 0 |

Имеем базисный вид системы:

Точка **А**

|  |  |  |  |
| --- | --- | --- | --- |
| + | x4 | x5 | = |
| x2 | 0,698 | -0,317 | 1,381 |
| x1 | 0,682 | 0,507 | 2,190 |
| x3 | 0,507 | 1,587 | 3,095 |
| L | -2,873 | -1,603 | -1,476 |

Для данной таблицы задача имеет вид:

Обнуляя небазисные переменные , получаем начальное опорное решение:

Соответствующее значение целевой функции:

2 этап работы:

Следующий этап работы заключается в максимизации целевой функции, согласно симплекс-методу мы выбираем разрешающий столбец (аналогично предыдущему пункту), а в качестве разрешающей строки выбираем ту, в которой достигается минимум отношения константы ограничений к элементу разрешающего столбца .

|  |  |  |  |
| --- | --- | --- | --- |
| + | x4 | x5 | = |
| x2 | 0,698 | -0,317 | 1,381 |
| x1 | 0,682 | 0,507 | 2,190 |
| x3 | 0,507 | 1,587 | 3,095 |
| L | -2,873 | -1,603 | -1,476 |

(Синим выделены разрешающий столбец и строка)

Точка **B**

|  |  |  |  |
| --- | --- | --- | --- |
| + | x2 | x5 | = |
| x4 | 1,431 | -0,454 | 1,977 |
| x1 | -0,977 | 0,818 | 0,840 |
| x3 | -0,727 | 1,818 | 2,090 |
| L | 4,113 | -2,909 | 4,204 |

Обнуляя небазисные переменные , получаем промежуточное опорное решение:

Соответствующее значение целевой функции:

Так как при L остались отрицательные коэффициенты продолжаем алгоритм, выводя из базиса и вводя в него . Получаем:

Точка **C**

|  |  |  |  |
| --- | --- | --- | --- |
| + | x2 | x1 | = |
| x4 | 0,888 | 0,555 | 2,444 |
| x5 | -1,194 | 1,222 | 1,027 |
| x3 | 1,444 | -2,222 | 0,222 |
| L | 0,638 | 3,555 | 7,194 |

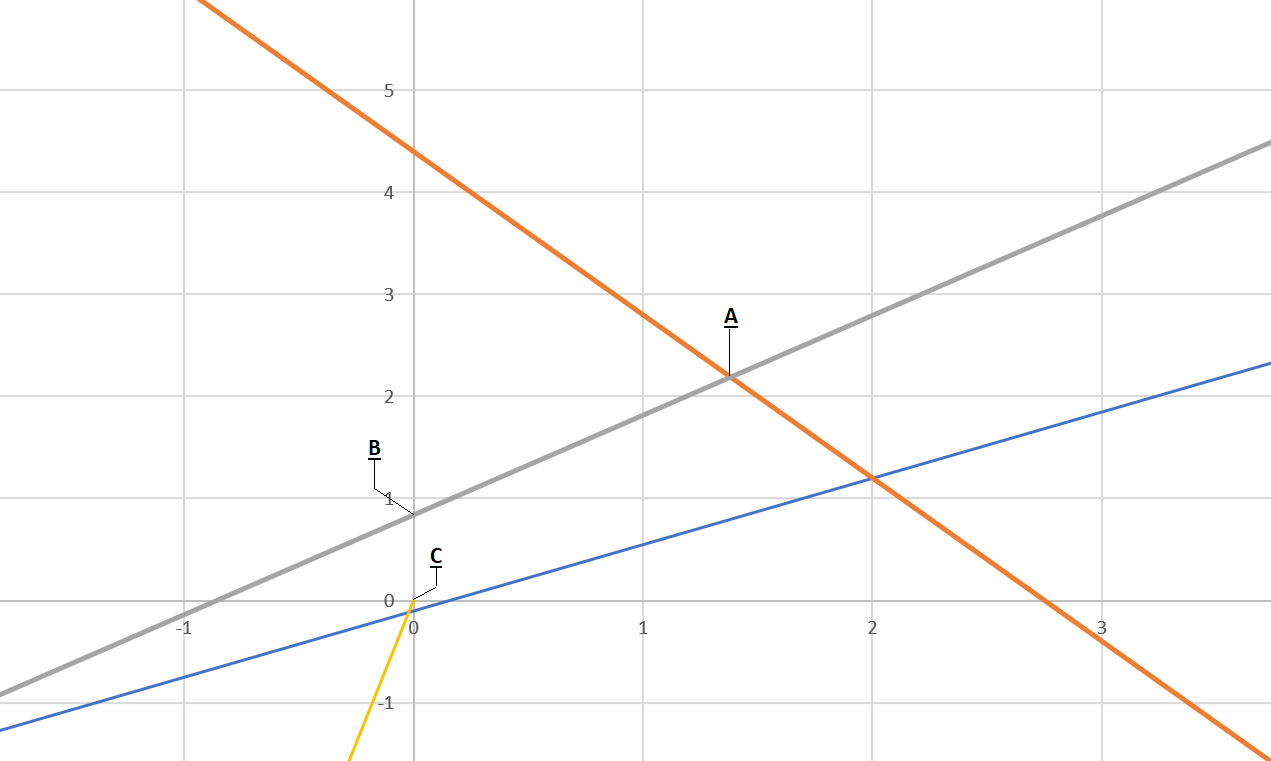
Отсутствие отрицательных коэффициентов при небазисных переменных в L свидетельствует о достижении оптимального решения:

Соответствующее значение целевой функции:

То есть

Проверка полученных результатов была проведена с помощью инструмента «Поиск решения» в MS Office Excel.

Ниже представлен график геометрического метода решения данной задачи в координатах , отмечены точки, полученные в ходе выполнения симплекс-метода.



**Программная реализация**

При написании программы я руководствовался теми же соображениями, что и в пункте “Решение”.

После подачи на вход программы исходных данных в цикле проводится проверка на неотрицательность вектора ограничений b. Далее собирается таблица из исходных данных размера (в моем случае) 5x6, 5 первых столбцов представляют собой соответственно, последний столбец – равенство (=), первые 3 строки – это искусственные переменные , 4-ая строка – функция , последняя - , между строками и столбцами (кроме последнего столбца) подразумевается суммирование, элементы – коэффициенты при переменных столбца.

Затем, выполняется первый этап работы, метод искусственного базиса, это происходит в цикле (3 итерации, которые соответствуют трем искусственным переменным). На каждой итерации высчитываются отношения ограничений к элементам разрешающего столбца, то есть происходит выбор разрешающей строки (за это отвечает переменная h). В конце каждого шага цикла удаляется столбец с выведенной из базиса искусственной переменной. В переменную ni записываются номера переменных введенных в базис (для подстановки нулей в соответствующие места небазисных переменных). После данного цикла вычисляются номера, соответствующие небазисным переменным, и на их места ставятся нули, выдавая начальное опорное решение (XOpor).

Следующий цикл отвечает за 2 этап работы, условие цикла: двигаться дальше, пока хоть один элемент (кроме последнего, так как это ответ функции при обнуленных небазисных переменных) последней строки таблицы отрицательный. Так как по условию необходимо фиксировать промежуточные таблицы, то их все записывал в одну переменную, в роли разделителя выступал столбец нулей. В целом, данный цикл работает аналогично предыдущему. Когда происходит выход из цикла, то есть все коэффициенты при небазисных переменных в L неотрицательны, ответ записывается в переменную XOptimal, далее на нужные места ставятся нули, и получаем оптимальное решение, записанное в переменную OTBET.

**Приложение**

Листинг программного кода:

clc;

clear all;

A=[5 8 0 9 0;-3 0 6 1 8;-9 3 6 -1 4];

c=[5 -9 0 0 7]';

b=[22 12 3]';

n=length(A);

L=c;

%%%proverka na otricatelnost

for i=1:length(b)

if b(i)<0

A(i,:)=-A(i,:);

b(i)=-b(i);

end;

end;

Acopy=A;

b1=[b;0]; % 0 if L=0

L1=-sum(Acopy(:,:));

c1=-c;

J=[Acopy;c1'];

J=[J b1];

Awithb=[Acopy b];

L1=-sum(Awithb(:,:))

J=[J;L1] %%% J - table with synthetic basis unknown

Jcopy=J;

k=0;

NB=[];

iINDEX=[];

ni=[];

for e=1:length(J(1,:))-3

ni=[ni 0]; %%unknown for calculate position x=0

end;

for ik=(length(J(1,:))-3):-1:1 %%

h=[];

[minn j]=min(J(length(J(:,1)),1:length(J(1,:))-1)); %% razreshausiy stolbec

%%%

F=0;

for g=1:(length(J(:,1))-2)

h=[h J(g,length(J(1,:)))/J(g,j)]; %% otnoshenie b/a\_is

end;

for g=1:(length(J(:,1))-2)

if J(g,j)<0

h(g)=1/0; %% tak kak vibor tolko dly pologitelnih znachenii a\_is

end; %% to dly otricatelnih obrashaem v beskonechnost (budem schitati minimum)

end;

while F~=1 %%% cikl, dly togo chtobi ne povtorylis vivedennie i vvedennie v bazis peremennie

[minh i]=min(h)

BBBBB=isempty(find(iINDEX==i))

if isempty(find(iINDEX==i))==0

h(i)=1/0;

else

F=1;

end;

end;

iINDEX=[iINDEX i]; %%% i - razreshaushay stroka (3 cikla sverhu dly ee vichisleniy)

% % %

RE=J(i,j);

J1=J;

J(:,j)=-J(:,j);

J(i,j)=1

for ii=1:n

for jj=1:n+1-k

if (ii~=i && jj~=j)

J(ii,jj)=J1(ii,jj)\*RE-J1(ii,j)\*J1(i,jj) %% algoritm

end;

end;

end;

J(:,:)=J(:,:)/RE;

k=k+1;

ni(i)=j; %%% nomera peremennih vvedenih v bazis

if k~=1

if j>=NB(k-1)

ni(i)=j+1;

end;

end;

if k==1

NB=[NB j]

end;

if k~=1

if j>=NB(k-1)

NB=[NB j+1]

else

NB=[NB j]

end;

end;

J(:,j)=[]

end;

NB=ni; %%% nomera bazisnih peremennih

J(length(J(:,1)),:)=[];

XBasis=NB

Xb=J(1:length(J(:,1))-1,length(J(1,:)))

XneBasis=[];

for i=1:length(c)

if isempty(find(XBasis==i))==1

XneBasis=[XneBasis i]; %%% nomera nebazisnih peremennih

end;

end;

FX=[Xb XBasis'];

FX=sortrows(FX,2);

FX=FX(:,1)';

for i=length(XneBasis):-1:1

p=ceil(XneBasis(i));

if p>length(FX)

FX=[FX 0];

else

FX=[FX(1:p-1), 0, FX(p:length(FX))];

end;

end;

XOpor=FX; %% nachalnoe opornoe reshenie

Lprom=XOpor\*c

sc=0;

PromTable=[];

Razdelitel=[];

for u=1:length(J(:,1))

Razdelitel=[Razdelitel;0]; %%%% tablici budut razdelytsy nulymi (0)

end;

while min(J(length(J(:,1)),1:length(J(1,:))-1))<0 %% 2-oi etap simplex-metoda

sc=sc+1;

PromTable=[PromTable J];

PromTable=[PromTable Razdelitel];

h=[];

[minn j]=min(J(length(J(:,1)),1:length(J(1,:))-1));

for g=1:(length(J(1,:)))

h=[h J(g,length(J(1,:)))/J(g,j)];

end;

for g=1:(length(J(1,:)))

if J(g,j)<0

h(g)=1/0;

end;

end;

[minh i]=min(h)

a=XBasis(i);

b=XneBasis(j);

XBasis(i)=b;

XneBasis(j)=a;

RE=J(i,j);

J1=J;

J(:,j)=-J(:,j);

J(i,j)=1

for ii=1:length(J(:,1)) %n-1

for jj=1:length(J(1,:)) % n-2

if (ii~=i && jj~=j)

J(ii,jj)=J1(ii,jj)\*RE-J1(ii,j)\*J1(i,jj)

end;

end;

end;

J(:,:)=J(:,:)/RE;

end;

PromTable=[PromTable J];

XOptimal=J(1:length(J(:,1))-1,length(J(1,:)))';

XOptimal=[XOptimal;XBasis]';

XOptimal=sortrows(XOptimal,2)';

XOptimal=XOptimal(1,:);

XneBasis=sort(XneBasis);

for i=1:length(XneBasis)

p=ceil(XneBasis(i));

if p>length(XOptimal)

XOptimal=[XOptimal 0];

else

XOptimal=[XOptimal(1:p-1), 0, XOptimal(p:length(XOptimal))];

end;

end;

OTBET=XOptimal