Российский государственный университет нефти и газа (НИУ)

имени И.М. Губкина

ОТЧЕТ

о выполнении домашнего задания № 6

по дисциплине «Методы оптимизации»

Выполнил: Хуснутдинов Эдуард Тимурович

Группа: АМ-17-06

Москва, 2020

**Постановка задачи**

Решить задачу линейного программирования модифицированным 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. Для приведения системы к допустимому базисному виду (под допустимым базисом понимается такой базис, которому соответствует опорное решение системы), необходимо вывести все искусственные переменные. Этому соответствует задача максимизации функции .

Максимизации функции проводится аналогично симплекс-методу, разница лишь в том, что в модифицированном симплекс-методе на каждой итерации в текущем базисе происходит перерасчет только коэффициентов функции , разрешающего столбца, столбца правой части и обратной матрицы.

Выпишем коэффициенты перед соответствующими переменными.

??? Вы тут любого запутаете!! У Вас здесь жуткая нумерация!!! (исправил)  
где первые 3 вектора – это коэффициенты перед базисными переменными,

Коэффициенты функции :

Вектор правой части:

Значение функции  **Итерация 1.**

Базис системы составляют столбцы коэффициентов при переменных

Поэтому первый базис будет единичным, как и обратная к нему матрица.

коэффициенты при небазисных переменных по формуле:

Наименьший отрицательный коэффициент Следовательно, переменная пойдет в базис, s=3. Посчитаем столбец в данном базисе и вектор правой части:

Разрешающую строку выбираем по минимальному отношению элемента вектора правой части к элементу разрешающей строки (элемент разрешающей строки должен быть положительными, иначе, если таких элементов не будет, целевая функция будет не ограничена на допустимом множестве), в данном случае, разрешающей будет строка 3, так как , строка 3 соответствует переменной , следовательно она выходит из базиса.

Вычисляем элементарную матрицу:

Она будет равна единичной матрице за тем лишь исключением, что на столбце, “имеющим” номер разрешающей строки, элементы будут рассчитываться следующем образом: (далее, r – номер разрешающей строки)

*где*

Тогда матрица

Для получения матрицы, обратной новой матрице умножаем матрицу на матрицу слева:

Коэффициенты целевой функции в новом базисе:

**Итерация 2.**

Теперь базис системы составляют столбцы (переменные ), посчитаем с помощью коэффициенты при небазисных переменных, коэффициенты при небазисных искусственных переменных вычислять не будем, так как их из системы мы выводим.

Наименьший отрицательный коэффициент Следовательно, переменная пойдет в базис, s=1. Посчитаем столбец в данном базисе и вектор правой части:

Разрешающей будет строка 2, так как , строка 2 соответствует переменной , следовательно она выходит из базиса.

Вычисляем элементарную матрицу

Для получения матрицы, обратной новой матрице умножаем матрицу на матрицу слева:

Коэффициенты целевой функции в новом базисе:

**Итерация 3.**

На данном шаге базис системы составляют столбцы (переменные ), посчитаем с помощью коэффициенты при небазисных переменных.

Наименьший отрицательный коэффициент Следовательно, переменная пойдет в базис, s=2. Посчитаем столбец в данном базисе и вектор правой части:

Разрешающей будет строка 1, так как это единственная строка, в которой элемент положительный, строка 1 соответствует переменной , следовательно она выходит из базиса.

Вычисляем элементарную матрицу

Вычислим матрицу :

Коэффициенты целевой функции в новом базисе:

**Итерация 4.**

Теперь базис системы составляют столбцы Посчитаем с помощью коэффициенты при небазисных переменных:

Все коэффициенты получили неотрицательное значение. Следовательно, получено оптимальное решение.

Считаем значение функции по формуле:

Получили допустимый опорный базис.

**2 этап работы:**

Следующий этап работы заключается в максимизации целевой функции аналогично первому этапу (вместо ). Тогда получаем:

коэффициенты перед соответствующими переменными:

Коэффициенты целевой функции:

Вектор правой части:

Матрица

**Итерация 1.**

Базис системы составляют переменные посчитаем с помощью коэффициенты при небазисных переменных по формуле:

Наименьший отрицательный коэффициента равен (-2,873). Следовательно переменная пойдет в базис. Посчитаем столбец и вектор правой частив новом базисе по формулам:

Разрешающую строку также, как и на 1 этапе выбираем по минимальному отношению элемента вектора правой части к элементу разрешающей строки, в данном случае, разрешающей будет строка 1, так как , строка 1 соответствует переменной , следовательно она выходит из базиса.

Вычисляем элементарную матрицу:

Для получения матрицы, обратной новой матрице умножаем матрицу на матрицу слева:

Коэффициенты целевой функции в новом базисе:

**Итерация 2.**

Теперь базис системы составляют столбцы , посчитаем с помощью коэффициенты при небазисных переменных.

Наименьший отрицательный коэффициента равен (-2,9091). Следовательно переменная пойдет в базис. Посчитаем столбец и вектор правой частив новом базисе:

Разрешающей будет строка 2, так как , строка 2 соответствует переменной , следовательно она выходит из базиса.

Вычисляем элементарную матрицу

Для получения матрицы, обратной новой матрице умножаем матрицу на матрицу слева:

Коэффициенты целевой функции в новом базисе:

**Итерация 3.**

Теперь базис системы составляют столбцы , посчитаем с помощью коэффициенты при небазисных переменных.

Все коэффициенты получили неотрицательное значение. Следовательно, получено оптимальное решение. Его координаты для базисных переменных равны вектору правой частив данном базисе.

Значение функции цели ищется по формуле:

Итак, получено оптимальное решение , значение функции цели на котором равно .

Проведем проверку решения с помощью инструмента «Поиск решения» в MS Office Excel.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Результат: Решение найдено. Все ограничения и условия оптимальности выполнены.** | | | | | | |  |
| **Модуль поиска решения** | | | |  |  |  |  |
|  | Модуль: Поиск решения лин. задач симплекс-методом | | | |  |  |  |
|  | Время решения: 0,032 секунд. | | |  |  |  |  |
|  | Число итераций: 5 Число подзадач: 0 | | | |  |  |  |
| **Параметры поиска решения** | | | |  |  |  |  |
|  | Максимальное время Без пределов, Число итераций Без пределов, Precision 0,000001 | | | | | |  |
|  | Максимальное число подзадач Без пределов, Максимальное число целочисленных решений Без пределов, Целочисленное отклонение 1%, Считать неотрицательными | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Ячейка целевой функции (Максимум) | | | | |  |  |  |
|  | **Ячейка** | **Имя** | **Исходное значение** | **Окончательное значение** |  |  |  |
|  | $D$1 | L= | 0 | 7,194444444 |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Ячейки переменных | | | |  |  |  |  |
|  | **Ячейка** | **Имя** | **Исходное значение** | **Окончательное значение** | **Целочисленное** |  |  |
|  | $B$1 | X1 | 0 | 0 | Продолжить |  |  |
|  | $B$2 | X2 | 0 | 0 | Продолжить |  |  |
|  | $B$3 | X3 | 0 | 0,222222222 | Продолжить |  |  |
|  | $B$4 | X4 | 0 | 2,444444444 | Продолжить |  |  |
|  | $B$5 | X5 | 0 | 1,027777778 | Продолжить |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Ограничения | | |  |  |  |  |  |
|  | **Ячейка** | **Имя** | **Значение ячейки** | **Формула** | **Состояние** | **Допуск** |  |
|  | $D$4 | R1= | 22 | $D$4=$E$4 | Привязка | 0 |  |
|  | $D$5 | R2= | 12 | $D$5=$E$5 | Привязка | 0 |  |
|  | $D$6 | R3= | 3 | $D$6=$E$6 | Привязка | 0 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ячейки переменных | | | |  |  |  |  |
|  |  |  | **Окончательное** | **Приведенн.** | **Целевая функция** | **Допустимое** | **Допустимое** |
|  | **Ячейка** | **Имя** | **Значение** | **Стоимость** | **Коэффициент** | **Увеличение** | **Уменьшение** |
|  | $B$1 | X1 | 0 | -3,555555556 | 5 | 3,555555556 | 1E+30 |
|  | $B$2 | X2 | 0 | -0,638888889 | -9 | 0,638888889 | 1E+30 |
|  | $B$3 | X3 | 0,222222222 | 0 | 0 | 1,6 | 0,442307692 |
|  | $B$4 | X4 | 2,444444444 | 0 | 0 | 1E+30 | 0,71875 |
|  | $B$5 | X5 | 1,027777778 | 0 | 7 | 0,534883721 | 2,909090909 |
|  |  |  |  |  |  |  |  |
| Ограничения | | |  |  |  |  |  |
|  |  |  | **Окончательное** | **Тень** | **Ограничение** | **Допустимое** | **Допустимое** |
|  | **Ячейка** | **Имя** | **Значение** | **Цена** | **Правая сторона** | **Увеличение** | **Уменьшение** |
|  | $D$4 | R1= | 22 | -0,388888889 | 22 | 18,5 | 4 |
|  | $D$5 | R2= | 12 | 1,75 | 12 | 1,333333333 | 4,111111111 |
|  | $D$6 | R3= | 3 | -1,75 | 3 | 4,111111111 | 0,666666667 |

**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]';

bcopy=b;

ccopy=c;

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;

bcopy=b;

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(:,:))

NB=[];

iINDEX=[];

ni=[];

AA=[];

b=[b;0]

A=[A;-c']

AA=eye(length(b));

AA=[AA A];

c1=1:length(AA(1,:));

for i=1:length(c1)

if i>length(b)

c1(i)=L1(i-length(b));

else

c1(i)=0;

end;

end;

L1=L1(length(L1))

Bp=1:length(b); %%% basisnie peremennie

NB=length(b)+1:length(AA(1,:)) %%nebasisnie peremennie

cb=zeros(1,length(Bp));

iter=1

cdo=-c1;

cc=-c1;

bb=b;

B=eye(length(Bp));

for i=1:length(Bp)

B(:,i)=AA(:,Bp(i));

end;

iB=inv(B);

coefS=[-1 -1];

ni=NB;

while min(coefS)<0

if iter~=1

cb(ms)=cdo(oldB); %coef pri basisnih peremennih

end;

cs=[];

for i=1:length(NB)

cs=[cs cb\*iB\*AA(:,NB(i))-cc(NB(i))]; %%coeff pri nebazis perem

end;

coefS=cs;

for i=1:length(coefS)

if abs(coefS(i))<0.00000000001

coefS(i)=0;

end;

end;

[coef mm]=min(coefS);

if coef>=0

XB=iB\*bb;

LLL=cb\*iB\*bb+L1

display 'optimalnoe reshenie';

break;

end;

ib=NB(mm);

As=iB\*AA(:,NB(mm));

if max(As)<=0

display 'Celevay funcia ne ogranichena na dopust mnogestve';

return

end;

bbs=iB\*bb;

MB=[];

for j=1:length(bbs)

MB=[MB bbs(j)/As(j)];

end;

for j=1:length(bb)

if As(j)<0

fg=MB(j);

MB(j)=999999;

%if As(j)>-0.000001

%MB(j)=fg;

%end;

end;

MB(4)=999999;

end;

[s ms]=min(MB);

RazS=ni(ms);

Et=B;

for i=1:length(bbs)

if i==ms

e(i)=1/As(ms);

else

e(i)=-As(i)/As(ms)

end

end

Et(:,ms)=e; %% elementarnay matrica

iB2=Et\*iB;

iB=iB2;

oldB=NB(mm);

NB(mm)=Bp(ms);

Bp(ms)=oldB;

ni=NB;

iter=iter+1;

end;

iB(:,length(iB))=[];

iB(length(iB),:)=[];

B=eye(length(bcopy));

Bp=Bp(Bp~=length(b1))

NB=NB(NB>3)

for i=1:length(Bp)

Bp(i)=Bp(i)-4

end;

for i=1:length(NB)

NB(i)=NB(i)-4

end;

ni=NB;

iter=1;

cc=1:length(c);

bb=XB(1:3)

AAA3=AA(1:3,5:9)

AA=AA(1:3,5:9)

cc=c;

EY=eye(length(Bp));

for i=1:length(Bp)

AA(:,Bp(i))=EY(:,i); %%tablica coefficentov pered peremennimi

cc(Bp(i))=0; %% coefficienti v func celi (vida L=)

end;

L=XB(length(XB));

for i=1:length(NB)

cdo=cc(NB(i));

end;

cdo=cc;

cc=c;

cdo=c;

B=eye(length(Bp));

for i=1:length(Bp)

B(:,i)=AA(:,Bp(i));

%B(:,i)=AA(:,i);

end;

cb=zeros(1,length(Bp));

coefS=[-1 -1];

AA=AAA3;

for i=1:3

for j=1:5

if Bp(i)==j

cb(i)=ccopy(j);

end

end

end

while min(coefS)<0

%if iter==1

%cb=zeros(1,length(Bp));

%end;

if iter~=1

%cb(iter-1)=cdo(NB(mm));

cb(ms)=cdo(oldB);

end;

cs=[];

for i=1:length(NB)

cs=[cs cb\*iB\*AA(:,NB(i))-cc(NB(i))]; %%coeff pri nebazis perem

end;

coefS=cs;

[coef mm]=min(coefS);

if coef>=0

XB=iB\*bb;

LLLk=cb\*iB\*bcopy

display 'optimalnoe reshenie';

return

end;

ib=NB(mm);

As=iB\*AA(:,NB(mm));

if max(As)<=0

display 'Celevay funcia ne ogranichena na dopust mnogestve';

return

end;

%bbs=iB\*bb;

bbs=iB\*bcopy

MB=[];

for j=1:length(bbs)

MB=[MB bbs(j)/As(j)];

end;

for j=1:length(bb)

if As(j)<0

MB(j)=999999;

end;

if bbs(j)<0

MB(j)=999999;

end;

end;

[s ms]=min(MB);

%RazS=ni(ms);

Et=B;

e=[];

for i=1:length(bbs)

if i==ms

e(i)=1/As(ms);

else

e(i)=-As(i)/As(ms)

end

end

Et(:,ms)=e; %% elementarnay matrica

iB2=Et\*iB;

iB=iB2;

iter=iter+1;

oldB=NB(mm);

NB(mm)=Bp(ms);

Bp(ms)=oldB;

ni=NB;

end;