UMA035 - Optimization Techniques Lab (MATLAB)

Submitted by:

Aryaman

Kalia

(102003099)

(COE 6)

Submitted to -

Dr. Bhuvaneshvar Kumar

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department TIET, Patiala

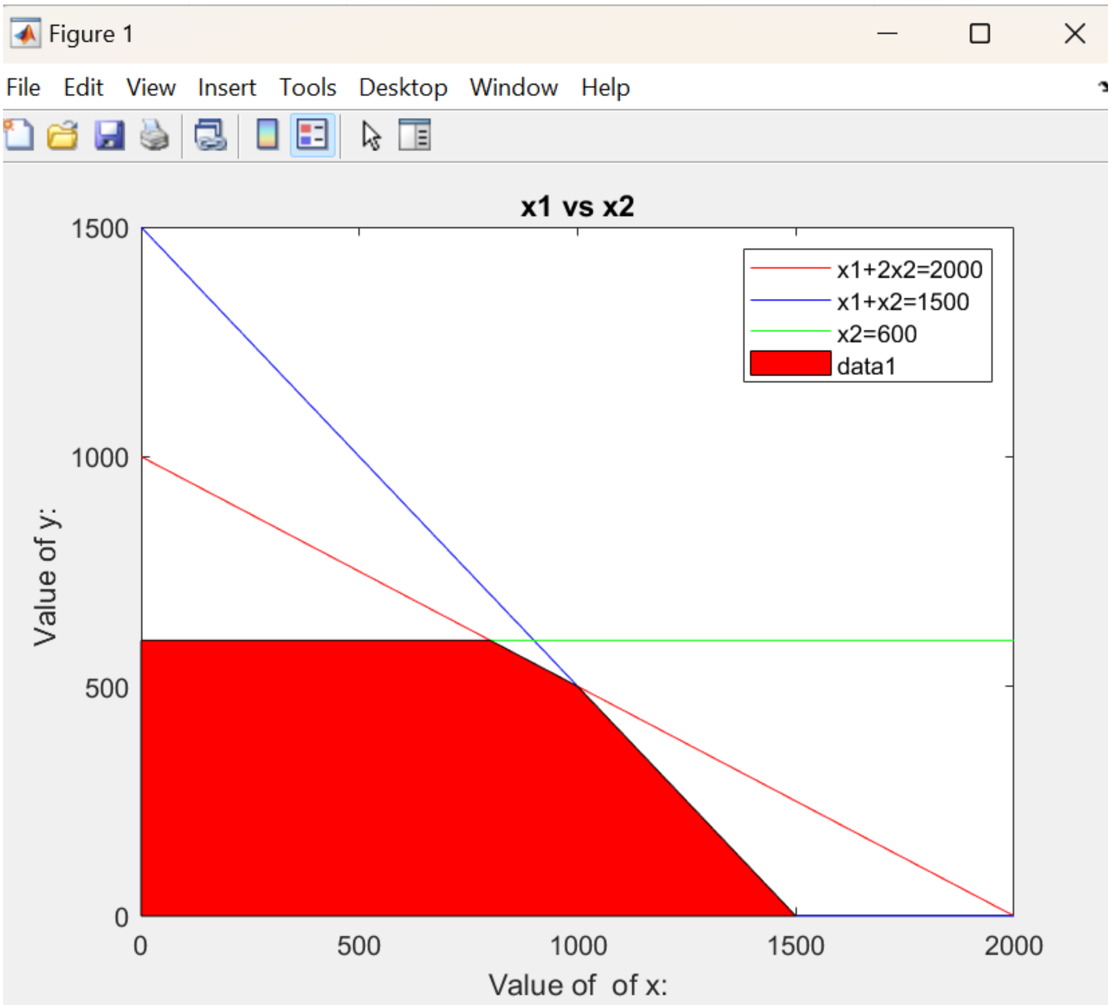
(January-May) 2023

TABLE OF CONTENTS

1 GRAPHICAL METHOD :-

```
clc
clear all
format short
c=[3 5];
A=[1 2;1 1;0 1];
b=[2000; 1500; 600];
y1=0:1:max(b);
x21=(b(1)-A(1,1)*y1)/A(1,2);
x22=(b(2)-A(2,1)*y1)/A(2,2);
x23=(b(3)-A(3,1)*y1)/A(3,2);
x21=max(0,x21);
x22=max(0,x22);
x23=max(0,x23);
plot(y1,x21,'r',y1,x22,'k',y1,x23,'b')
xlabel('Value of  x1')
ylabel('Value of x2')
cx1=find(y1==0)
c1=find(x21==0
Line1=[y1(:,[c1 cx1]);x21(:,[c1 cx1])]'
c2=find(x22==0)
Line2=[y1(:,[c2 cx1]);x22(:,[c2 cx1])]'
c3=find(x23==0)
Line3=[y1(:,[c3 cx1]);x23(:,[c3 cx1])]'
corpt=unique([Line1;Line2;Line3],'rows');
HG=[0;0];
for i=1:size(A,1)
    a1=A(i,:);
    b1=b(i);
    for j=i+1:size(A,1)
        a2=A(j,:);
        b2=b(j);
        A_solve=[a1;a2];
        B_solve=[b1;b2];
        soln=inv(A_solve)*B_solve;
        HG=[HG soln];
    end
end
pt=HG';
all_points=[pt;corpt];
```

```
all_final=unique(all_points,'rows');
PT=constraint(all_final);
PT=unique(PT,'rows');
for i=1:size(PT,1)
   Fx(i,:)=sum(PT(i,:).*c);
end
final_vtx=[PT Fx];
[fxval,indfx]=max(Fx);
optval=final_vtx(indfx,:);
optimal_bfs=array2table(optval,'VariableNames',{'X1','X2',’Z’})
```



OUTPUT :-

```
ans =

        0              0
     1000            500
      800            600
      900            600


max =

     5500


m1 =

     1000


m2 =

      500
```

2 BFS :-

```
format short
clear all
clc
c=[2 3 4 7];
a=[2,3,-1,4; 1,-2,6,-7];
b=[8;-3];
n=size(a,2);
m=size(a,1);
basic_var=nchoosek(n,m);
basic_pair=nchoosek(1:n,m);
sol=[];
if n>m
    for i=1:basic_var
        y=zeros(n,1);
        x=a(:,basic_pair(i,:))\b;
        if all(x>=0 & x~=inf & x~= -inf)
            y(basic_pair(i,:))=x;
            sol=[sol y];
        end
    end
else
```

```
    error('Cannot Evaluate The result !!, more values than required')
    end z=c*sol;
[Zmax Zind]= max(z);
bfs=sol(:,Zind);
optval=[bfs' Zmax]
optimal_bfs=array2table(optval);
optimal_bfs.Properties.VariableNames(1:size(optimal_bfs,2))={'x_1','x_2','x_3','x_4','
z_val'};
```

OUTPUT :-

```
optimal_value =

         0         0    2.5882    2.6471   28.8824


optimal_bfs =

  1×5 table
```

x_1	x_2	x_3	x_4	z
0	0	2.5882	2.6471	28.882

3. SIMPLEX :-

```
clc
clear all
format short
c=[-1 3 -2];
info=[3 -1 2;-2 4 0;-4 3 8];
b=[7; 12; 10];
s=eye(size(info,1));
A=[info  s b];
var=3;
cost=zeros(1,size(A,2));
cost(1:3)=c;
BV=var+1:size(A,2)-1;
zjcj=cost(BV)*A-cost;
zcj=[zjcj;A];
simptable=array2table(zcj)
Run=true;
```

```
while Run
    if(any(zjcj<0))
        fprintf('Current BFS is not optimal\n');
        fprintf('Current values of BV are\n');
        disp(BV);
        zc=zjcj(1:end-1);
        [Entercol,pvt_col]=min(zc);
        if all(A(:,pvt_col)<=0)
            fprintf('LPP is Unbounded');
        end
        sol=A(:,end);
        column=A(:,pvt_col);
        for i=1:size(column,1)
            if column(i)>0
                ratio(i)=sol(i)./column(i);
            else
                ratio(i)=inf;
           end
        end
        [Minratio,pvt_row]=min(ratio);
        BV(pvt_row)=pvt_col;
        pvt_key=A(pvt_row,pvt_col);
         A(pvt_row,:)=A(pvt_row,:)./pvt_key;
         for i=1:size(A,1)
             if i~=pvt_row
                 A(i,:)=A(i,:)-A(i,pvt_col).*A(pvt_row,:);
             end
         end
         zjcj=zjcj-zjcj(pvt_col).*A(pvt_row,:);
         zcj=[zjcj;A];
         simptable=array2table(zcj)
         BFS=zeros(1,size(A,2));
         BFS(BV)=A(:,end);
         BFS(end)=sum(BFS.*cost);
         curr_bfs=array2table(BFS)
    else
        Run=false;
        fprintf('optimal solution !!!!!')
    end
end
```

OUTPUT :-

```
CurrentBFS =

  1×7 table

    x_1    x_2    x_3    s_1    s_2    s_3    Sol
    ___    ___    ___    ___    ___    ___    ___

     4      5      0      0      0     11     11


Run =

  logical

   0

The current BFS is optimal and Optimality is reached
```

4. Two Phase:

```
clc
clear all
format short
M=100;
cost=[-2 -1 0 0 -M -M 0];
info=[3 1 0 0 1 0;4 3 -1 0 0 1;1 2 0 1 0 0];
b=[3;6;3];
A=[info b];
BV=[5 6 4];
zjcj=cost(BV)*A-cost;
disptable=array2table([zjcj;A],'VariableNames',{'x1','x2','x3','s1','s2','s3','sol'})
Run=true;
while Run
    if any(zjcj(1:end-1)<0)
        fprintf('Solution is not optimal\n');
        fprintf('Current basic variables are\t');
        disp(BV);
        zc=zjcj(1:end-1);
        [value,pvt_col]=min(zc);
        sol=A(:,end);
        column=A(:,pvt_col);
        if all(column<=0)
            fprintf('Solution is unbounded');
        else
```

```
            for i=1:size(column,1)
                if column(i)<=0
                    ratio(i)=inf;
                else
                    ratio(i)=sol(i)./column(i);
                end
            end
            [element,pvt_row]=min(ratio);
            pvt_key=A(pvt_row,pvt_col);
            A(pvt_row,:)=A(pvt_row,:)./pvt_key;
            for i=1:size(A,1)
                if i~=pvt_row
                    A(i,:)=A(i,:)-A(i,pvt_col).*A(pvt_row,:);
                end
            end
            BV(pvt_row)=pvt_col;
            zjcj=cost(BV)*A-cost;
            disp([zjcj;A])
        end
    else
        fprintf('Solution is optimal\n');
        fprintf('Current basic variables are\t');
        disp(BV);
        Run=false;
    end
end
fprintf('Optimal solution is ')
b=cost(BV)*A-cost;
disp(b(end))
```

OUTPUT :-

```
TABLE =

  3×8 table

    x_1    x_2     x_3        s_1         s_2      A_1      A_2     sol
    ___    ____    ___    ___________    _____    _____    ____    ____

     0        0     0     5.5511e-17        0        1       1       0
     1     -0.5     0          -0.25    -0.25     0.25    0.25    1.25
     0     -0.5     1           0.25    -0.75    -0.25    0.75    0.75


BFS =

    1.2500         0    0.7500         0         0         0    1.0000

 Current BFS is Optimal
Phase End
fx >>
```

5. Big-M:

```
clc
clear all
format short
cost=[0 0 0 0 0 -1 -1 0];
info=[3 -1 -1 -1 0 1 0;1 -1 1 0 -1 0 1];
b=[3;2];
A=[info b];
BV=[6 7];
zjcj=cost(BV)*A-cost;
disptable=[zjcj;A];
disp(disptable)
Run=true;
while Run
  zc=zjcj(1:end-1)
  if all(zc>=0)
    Run=false;
    fprintf(' Current BFS is Optimal \n');
    fprintf('Phase End \n')
    disp(sol)
  else
    [element,pvt_col]=min(zc);
    if all(A(:,pvt_col)<=0)
      fprintf('Unbounded solution');
```

```
    else
      column=A(:,pvt_col);
      solution=A(:,end);
      for i=1:size(column,1)
        if(column(i)<=0)
          ratio(i)=inf;
        else
          ratio(i)=solution(i)./column(i);
        end
      end
      [element,pvt_row]=min(ratio);
      BV(pvt_row)=pvt_col;
      A(pvt_row,:)=A(pvt_row,:)./A(pvt_row,pvt_col);
      for i=1:size(A,1)
        if i~=pvt_row
          A(i,:)=A(i,:)-A(pvt_row,:).*A(i,pvt_col);
          end
          end
      zjcj=cost(BV)*A-cost;
      disptable=[zjcj;A];
      disp(disptable)
      sol=A(:,end);
    end
  end
end
```

OUTPUT :-

```
TABLE =

  4×7 table

    x_1    x_2    s_1     s_2    A_1       A_2       sol
    ___    ___    ____    ___    ______    ______    ____

     0      0      0.2     0     9999.6    9999.8    -2.4
     1      0      0.2     0        0.6      -0.2     0.6
     0      1     -0.6     0       -0.8       0.6     1.2
     0      0        1     1          1        -1       0


ZC =

   1.0e+03 *

         0         0    0.0002         0    9.9996    9.9998

 Current BFS is Optimal
>>
```

6. Dual Simplex:

```
variables={'x_1','x_2','x_3','s_1','s_2','sol'};
cost=[-2 0 -1 0 0 0];
A=[-1 -1 1 1 0 -5; -1 2 -4 0 1 -8];
s=eye(size(A,1))
BV=[];
for j=1:size(s,2)
   for i=1:size(A,2)
      if A(:,i)==s(:,j)
         BV=[BV i]
      end
   end
end
ZjCj=cost(BV)*A-cost
ZCj=[ZjCj;A];
simpletable=array2table(ZCj);
simpletable.Properties.VariableNames(1:size(ZCj,2))=variables
while true
   sol=A(:,end);
   if any(sol<0)
      fprintf('The current BFS is not feasible \n');
      [LeavingVal,pvt_row]=min(sol);
```

```
    fprintf('The leaving variable is %d and the pivot row is %d\n',LeavingVal,pvt_row);
    ROW=A(pvt_row,1:end-1) ZJ=ZjCj(1,1:end-1)
    for i=1:size(ROW,2)
      if ROW(i)<0
        ratio(i)=abs(ZJ(i)./ROW(i));
      else
        ratio(i)=inf;
      end
    end
    [EnteringVal,pvt_col]=min(ratio);
    fprintf('The entering variable is %d and the pivot col is %d \n',EnteringVal,
pvt_col);
    BV(pvt_row)=pvt_col;
    fprintf('the basic variable')
    disp(variables(BV))
    pvt_key=A(pvt_row,pvt_col);
    A(pvt_row,:)=A(pvt_row,:)./pvt_key;
    for i=1:size(A,1)
      if i~=pvt_row
        A(i,:)=A(i,:)-A(i,pvt_col).*A(pvt_row,:);
      end
    end
    ZjCj=ZjCj-ZjCj(pvt_col)*A(pvt_row,:);
    final_BFS=zeros(1,size(A,2))
    final_BFS(BV)=A(:,end);
    final_BFS(end)=sum(final_BFS.*cost);
    optimal_BFS=array2table(final_BFS);
    optimal_BFS.Properties.VariableNames(1:size(optimal_BFS,2))=variables
  else
    fprintf('The current BFS is feasible and optimal ');
    break;
  end
end
```

OUTPUT :-

```
SimpTable =

  3×6 table

    x_1    x_2    x_3    s_1    s_2     Sol
    ___    ___    ___    ___    ____    ___

    0.5     0      0      1      0.5    -9
    2.5     1      0     -2     -0.5    14
    1.5     0      1     -1     -0.5     9

The current BFS is Feasible and opimal
>>
```

7. LCM:

```
clc
clear all
format short
cost=[11 20 7 8;21 7 10 12; 8 12 18 9];
A=[50 40 70];
B=[30 25 35 40];
if sum(A) == sum(B)
   fprintf('Given Transportation is balanced');
else
   fprintf('Given Transportation problem is not balanced')
end
if sum(A) <sum(B)
   cost(end+1,:)=zeros(1,size(B,2));
   A(end+1)=sum(B)-sum(A);
else
   cost(:,end+1)=zeros(1,size(A,2));
   B(end+1)=sum(A)-sum(B)
end
Icost=cost;
X=zeros(size(cost))
[m,n]=size(cost);
BV=m+n-1;
for i=1:size(cost,1)
   for j=1:size(cost,2)
      hh=min(cost(:));
      [Row_index,col_index]=find(hh==cost);
```

```
        x11=min(A(Row_index),B(col_index));
        [value,index]=max(x11);
        ii=Row_index(index);
        jj=col_index(index);
        y11=min(A(ii),B(jj));
        X(ii,jj)=y11;
        A(ii)=A(ii)-y11;
        B(jj)=B(jj)-y11;
        cost(ii,jj)=Inf;
    end
end
fprintf('Initial BFS\n');
TotalBFS=length(nonzeros(X));
if TotalBFS==BV
    fprintf('Initial BFS is non degenerate\n');
else
    fprintf('Initial BFS is degenerate\n');
end
InitialCost=sum(sum(Icost.*X));
fprintf('Initial BFS Cost is %d \n',InitialCost);
```

OUTPUT :-

```
Command Window
Unbalanced transportation problem
A =

    50    40    70


B =

    30    25    35    40    30


cost =

    11    20     7     8     0
    21     7    10    12     0
     8    12    18     9     0


cost =

   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
```

```
X =

     0     0    20     0    30
     0    25    15     0     0
    30     0     0    40     0

Initial BFS =
Initial BFS is Degenerate
Initial BFS Cost is = 1065>>
```