BFS

```
clc
clear all
format short
A=[1 1 1 0;2 1 0 1]
c=[3 4 0 0]
b=[450;600]
n=size(A,2)
m=size(A,1)
if(n>m)
    ncm=nchoosek(n,m)
    pair=nchoosek(1:n,m)
    sol=[];
    for i=1:ncm
        y=zeros(n,1)
        x=A(:,pair(i,:))\b
        if all(x>=0 & x~=inf & x~=-inf)
            y(pair(i,:)) =x
            sol=[sol, y]
        end
    end
else
    error('ncm does not exists')
end
z=c*sol
[zmax, zindex]=max(z)
bfs=sol(:, zindex)
optimal_value=[bfs' zmax];
optimal_bfs=array2table(optimal_value)
optimal_bfs.Properties.VariableNames(1:size(optimal_bfs,2))={'x_1','x_2','x_3','x_4','z'}
```

Simplex

```
clc
clear all

a=[1 1;3 2]
B=[5;12]
c=[6 5]
Noofvariables=2
s=eye(size(a,1))
A=[a s B]
cost=zeros(1,size(A,2))
cost(1:Noofvariables)=c
bv=Noofvariables+1:1:size(A,2)-1
zjcj=cost(bv)*A-cost
zcj=[zjcj;A];
simptable=array2table(zcj)
simptable.Properties.VariableNames(1:size(zcj,2))={'x_1' ,'x_2' ,'s_1','s_2','sol'}
RUN=true
while RUN
zc=zjcj(1:end-1);
if any(zc<0);
    fprintf('the current BFS is not optimal \n')

    [Enter_val, pvt_col]= min(zc)
    if all(A(:,pvt_col)<0)
        error('LPP is unbounded all entries are <=0 in column %d',pvt_col);
    else
        sol=A(:,end)
        column=A(:,pvt_col)
        for i=1:size(A,1)
            if column(i)>0
                ratio(i)= sol(i)./column(i)
```

```
            else
                ratio(i)=inf
            end
        end
        [leaving_val,pvt_row]=min(ratio)
    end
    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];
    table=array2table(zcj);
    table.Properties.VariableNames(1:size(zcj,2))={'x_1','x_2','s_1','s_2','sol'}
else
    RUN=false;
    fprintf('The current BFS is optimal \n')
end
end
```

Standard

```
clc
clear all
c=[7,4]
A=[1,-3;1,2;1,0]
B=[4;5;3]
S=eye(size(A,1))
I=[0,0,1]
index=find(I==1)
S(index,index)= -S(index,index)
mat=[A S B]
obj=array2table(c)
obj.Properties.VariableNames(1:size(c,2))={'x_1','x_2'}
cons=array2table(mat);
cons.Properties.VariableNames(1:size(mat,2))={'x_1','x_2','s1','s2','s3','B'}
```

Graphical

```
clc
clear
format rat

c=[2,1];
A=[1 2; 1 1; 1 -2];
B=[10;6;1];
p=max(B);
y1=0:1:max(B);
x11=(B(1)-A(1,1).*y1)./A(1,2);
x21=(B(2)-A(2,1).*y1)./A(2,2);
x31=(B(3)-A(3,1).*y1)./A(3,2);
x11=max(0,x11);
x21=max(0,x21);
x31=max(0,x31);
plot(y1,x11,'r',y1,x21,'b',y1,x31,'g')
title('x1 vs x2')
xlabel('value of x1')
ylabel('value of x2')
cx1=find(y1==0)
c1=find(x11==0)
```

```
Line1=[y1(:,[c1 cx1]); x11(:,[c1 cx1])]'

c2=find(x21==0)
Line2=[y1(:,[c2 cx1]); x21(:,[c2 cx1])]' ;

c3=find(x31==0)
Line3=[y1(:,[c3 cx1]); x31(:,[c3 cx1])]' ;
corpt=unique([Line1;Line2;Line3],'rows')
pt=[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,:);
     A4=[A1;A2]
     B4=[B1;B2]
     X=A4\B4
     pt=[pt X]
     end
     end
     ptt=pt'

 allpt=[ptt;corpt]
 points=unique(allpt,'rows')

 PT=constraint(points)
 P=unique(PT,'rows')
```

Big M

```
clc
clear
M=1000;
art_var=[5 6];
A=[1 3 -1 0 1 0; 1 1 0 -1 0 1];
b=[3; 2];
c=[-3 -5 0 0 -M -M 0];
a=[A b];
array2table(a,'VariableNames',{'x1','x2','s1','s2','A1','A2', 'b'});
bv=[5 6];
z=c(bv)*a-c;
simplex_table=[z;a];
Var={'x1','x2','s1','s2','A1','A2', 'b'};
array2table(simplex_table,'VariableNames',Var)

for k=1:15
if all(z(1:end-1)>=0)
   if any(bv==art_var(1))||any(bv==art_var(2))

       fprintf('Infeasible solution');
       break;

   end
    fprintf('The current table is optimal\n');
    optimal_value=z(end);
    fprintf('The optimal value of the current lpp is %f',optimal_value);
    break;
else
    fprintf('The current table is not optimal');
     [entering_var_value, pvt_col]=min(z(1:end-1));
     if all(a(:,pvt_col)<=0)
         error('The lpp is unbounded');
     else
```

```
        sol=a(:,end);
        column=a(:,pvt_col);
        for i=1:size(a,1)
            if column(i)>0
                ratio(i)=sol(i)/column(i);
            else
                ratio(i)=inf;
            end
        end
        [leaving_var_value,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
        z=c(bv)*a-c;
        simplex_table=[z;a];
array2table(simplex_table,'VariableNames',Var)

    end
end
end
```

Two Phase

```
clc
clear all
Variables={'x_1','x_2','s_1','s_2','A_1','A_2','sol'};
OVariables={'x_1','x_2','s_1','s_2','sol'};
OrigC=[-4 -5 0 0 -1 -1 0]
a=[3 1 1 0 0 0;3 2 0 -1 1 0; 5 5 0 0 0 1]
b=[27;  3; 60];
A=[a b]

fprintf('** PHASE-1 ** \n')
cost=[0 0 0 0 -1 -1 0]
Artifical_var=[5 6]
bv=[3 5 6];

zjcj=cost(bv)*A-cost;
simplex_table=[zjcj;A];
array2table(simplex_table,'VariableNames',Variables)
RUN=true;
while RUN
if any(zjcj(1:end-1)<0)
 fprintf(' the current BFS is not optimal \n')
 zc=zjcj(1:end-1);
 [Enter_val, pvt_col]= min(zc);
 if all(A(:,pvt_col)<=0)
 error('LPP is Unbounded all enteries are <=0 in column %d',pvt_col);
 else
 sol=A(:,end);
 column=A(:,pvt_col);
 for i=1:size(A,1)
 if column(i)>0
 ratio(i)= sol(i)./column(i);
 else
 ratio(i)=inf;
 end
 end
 [leaving_val, pvt_row]=min(ratio);
 end
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=cost(bv)*A-cost;
 zcj=[zjcj;A];
 table=array2table(zcj,'VariableNames',Variables)
else
 RUN=false;
 if any(bv==Artifical_var(1)) || any(bv==Artifical_var(2))
     error('Infeasible solution');
 else
  fprintf('optimal table of phase-1 is achieved \n');
 end
end
end

fprintf('** PHASE-2 ** \n')
A(:,Artifical_var)=[];
OrigC(:,Artifical_var)=[];
cost=OrigC;
zjcj=cost(bv)*A-cost;
simplex_table=[zjcj;A];
array2table(simplex_table,'VariableNames',OVariables)

RUN=true;
while RUN
if any(zjcj(1:end-1)<0)
 fprintf(' the current BFS is not optimal \n')
 zc=zjcj(1:end-1);
 [Enter_val, pvt_col]= min(zc);
 if all(A(:,pvt_col)<=0)
 error('LPP is Unbounded all enteries are <=0 in column %d',pvt_col);
 else
 sol=A(:,end);
 column=A(:,pvt_col);
 for i=1:size(A,1)
 if column(i)>0
 ratio(i)= sol(i)./column(i);
 else
 ratio(i)=inf;
 end
 end
 [leaving_val, pvt_row]=min(ratio);
 end
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=cost(bv)*A-cost;
 zcj=[zjcj;A];
 table=array2table(zcj,'VariableNames',OVariables)
else
 RUN=false;
  fprintf('The current BFS is optimal \n');
  z=input(' Enter 0 for minimization and 1 for max \n');
    if z==0
        Obj_value=-zjcj(end);
    else
```

```
        Obj_value=zjcj(end);
    end
    fprintf('The final optimal value is %f\n',Obj_value);
 end
end
```