clc

clear all

format short

%%phase 1: Input parameter

%c=[2 1]

A=[1 2;1 1;0 1];

B=[2000;1500;600];

c=[3,5];

%%phase 2: Plotting the graph

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 of x:');

ylabel('Value of y: ');

legend('x1+2x2=2000','x1+x2=1500','x2=600');

%%grid on

%%phase 3: find corner point with axes

cx1=find(y1==0)

c1=find(x11==0)

c2=find(x21==0)

c3=find(x31==0)

a1 = [ y1(:,[c1 cx1]) ; x11(:,[c1 cx1]) ]'

a2 = [ y1(:,[c2 cx1]); x21(:,[c2 cx1]) ]'

a3 = [ y1(:,[c3 cx1]); x31(:,[c3 cx1]) ]'

corner\_points=unique([a1;a2;a3],'rows')

%%phase 4: to find points of intersection

points=[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,:);

Af=[A1;A2];

Bf=[B1;B2];

X=inv(Af)\*Bf;

points=[points X];

end

end

points'

%phase 5: write all corner points

all\_points=[points';corner\_points];

X=unique(all\_points,'rows');

%phase 6: find feasibile region

x1=X(:,1);

x2=X(:,2);

con1=x1+2\*x2-2000;

h1=find(con1>0);

X(h1,:)=[];

x1=X(:,1);

x2=X(:,2);

con2=x1+x2-1500;

h2=find(con2>0);

X(h2,:)=[];

x1=X(:,1);

x2=X(:,2);

con3=x2-600;

h3=find(con3>0);

X(h3,:)=[];

p=X;

%p %to print feasibile region points

%phase 7: find maxima

max=0;

m1=0;

m2=0;

for i=1:size(p)

obj=3\*p(i,1)+5\*p(i,2);

if(obj>max)

max=obj;

m1=p(i,1);

m2=p(i,2);

end

end

max

m1

m2

%to plot region

plot(y1,x11,'r',y1,x21,'b',y1,x31,'g');

title('x1 vs x2');

xlabel('Value of of x:');

ylabel('Value of y: ');

legend('x1+2x2=2000','x1+x2=1500','x2=600');

hold on;

patch(p(:,1),p(:,2),'r')

%to find bfs

%if n-m=0 then solutions are non basic

clc

clear all

format short

A=[2 3 -1 4;1 -2 6 -7];

B=[8;-3];

C=[2 3 4 7];

n=size(A,2); % number of columns(variables)

m=size(A,1); % number of constraints (rows)

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

sol'

else

error('nCm does not exist')

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'};

optimal\_bfs

% to convert generall lpp into standard form

clc

clear all

format short

C=[3,5];

A=[1 2;1 1;0 1];

B=[2000;1500;600];

% to identify <= or >= sign

% 0 for <= sign

% 1 for >= sign

ineqsign=[0 0 1];

% to introduce slack or surplus variable

s=eye(size(A,1));

index=find(ineqsign>0);

s(index,:)=-s(index,:);

% to convert into standard form

objfn=array2table(C);

objfn.Properties.VariableNames(1:size(C,2))={'x\_1','x\_2'}

s1=[A s B];

cons=array2table(s1);

cons.Properties.VariableNames(1:size(s1,2))={'x\_1','x\_2','s\_1','s\_2','s\_3','b'}