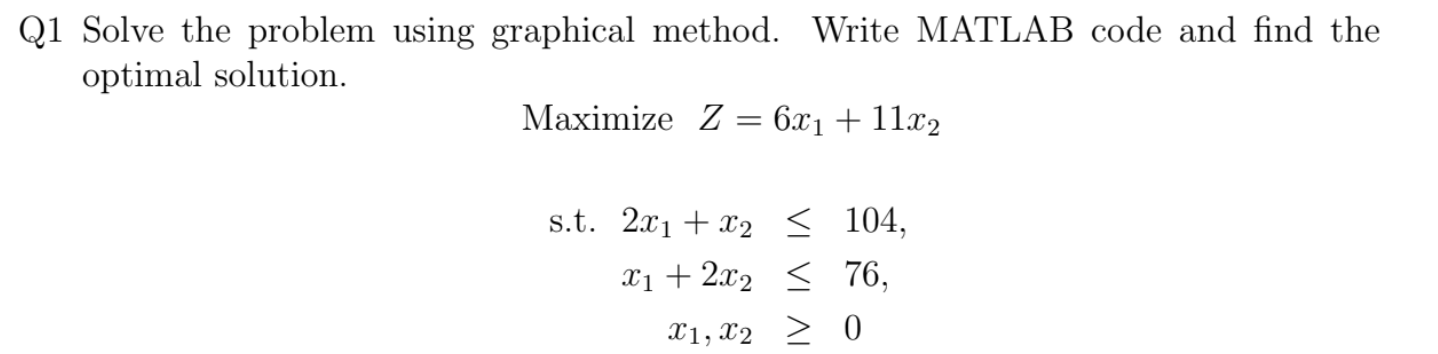
**OT LAB ASSIGNMENT 2-3: UPDATED**

****

Code:

%% QUESTION 1: Graphical method to solve

% Max Z= 6x1+11x2

% 2x1+x2<=104

% x1+2x2<=76

% x2>=0

% x1>=0

clc

clear all

format short

%INPUT PARAMETERS

c=[6,11]; %cost objective function

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

B=[104;76;0;0];

% 1 for <=const and -1 for >= const

const=[1;1];%for lesser than function we have 1 and -1 for greater than

objective=1;%1 for maximization and -1 for minimization

n=size(A,1);

x1=0:0.01:max(B);

for i=1:n-2 %we take n-2 since we are also taking x1=0 and x2=0 as they have no significance in our graph

y(i,:)=(B(i)-A(i,1)\*x1)/A(i,2);

end

%DRAWING THE LINES

for i=1:n-2

y(i,:)=max(0,y(i,:));

plot(x1,y(i,:),'linewidth',4)

hold on

end

hold on

%FINDING THE POINT OF INTERSECTION

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,:);

A3=[A1;A2];

B3=[B1;B2];

%X3=inv(A3)\*B3

X3=A3\B3;

if(X3>=0)%since the number of chairs can never be negative

pt= [pt X3];

end

end

end

X=pt';

X=unique(X,'rows')%solution

hold on

% KEEP ONLY FEASIBLE POINTS

x1=X(:,1);

x2=X(:,2);

for i=1:n-2 %n=size(A,1)-2

%for greater than(1) equation we remove A\*X'>B and for less than(-1) we do A(i,:)\*X'<B(i)

if(const(i)>0)

ind=find(A(i,:)\*X'>B(i));

X(ind,:)=[];% indexes that are not satisfying the constraint are being replaced with empty set

else

ind=find(A(i,:)\*X'<B(i));

X(ind,:)=[];

end

end

% EVALUATE THE OBJECTIVE FUNCTION VALUE

if(objective == 1)

obj\_val=c\*X';

[value, ind]=max(obj\_val);

fprintf("The max optimal value is : %f \n",value)

fprintf("The max optimal point is : (%g,%g) \n",X(ind,:))

else

obj\_val=c\*X';

[value, ind]=min(obj\_val);

fprintf("The min optimal value is : %f \n",value)

fprintf("The min optimal point is : (%g,%g) \n",X(ind,:))

end

X(ind,:);

Optimal=[X(ind,:) value];

% Shaded feasible region

x=X(:,1);

y=X(:,2);

scatter(X(:,1),X(:,2),'\*')

hold on

k=convhull(x,y);%the shaded region where a and y is satisfied

fill(x(k),y(k),'m')

% setting the axes

xlim([0 max(x)+1])

ylim([0 max(y)+1])

xlabel('x-axis')

ylabel('y-axis')

title('Feasible region of the linear programming problem')

% legend('2x\_1+x\_2\leq104','x\_1+2x\_2\leq76','x\_1,x\_2\geq0')

%phase 7: Verification

x=0:0.1:max(B);

for z=0:8:value

y=(z-c(1)\*x)/c(2);

plot(x,y)

hold on

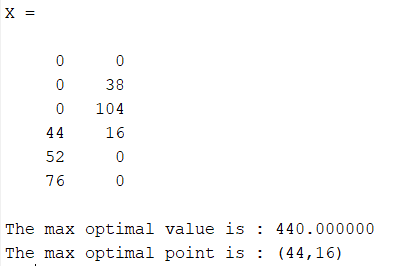
drawnow

pause(0.001)

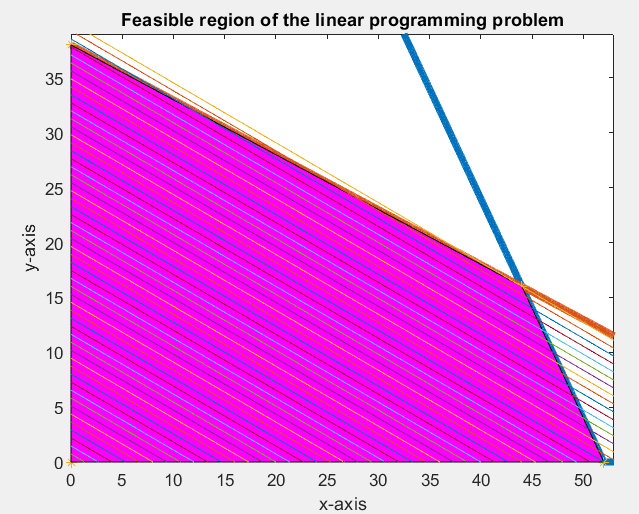
end

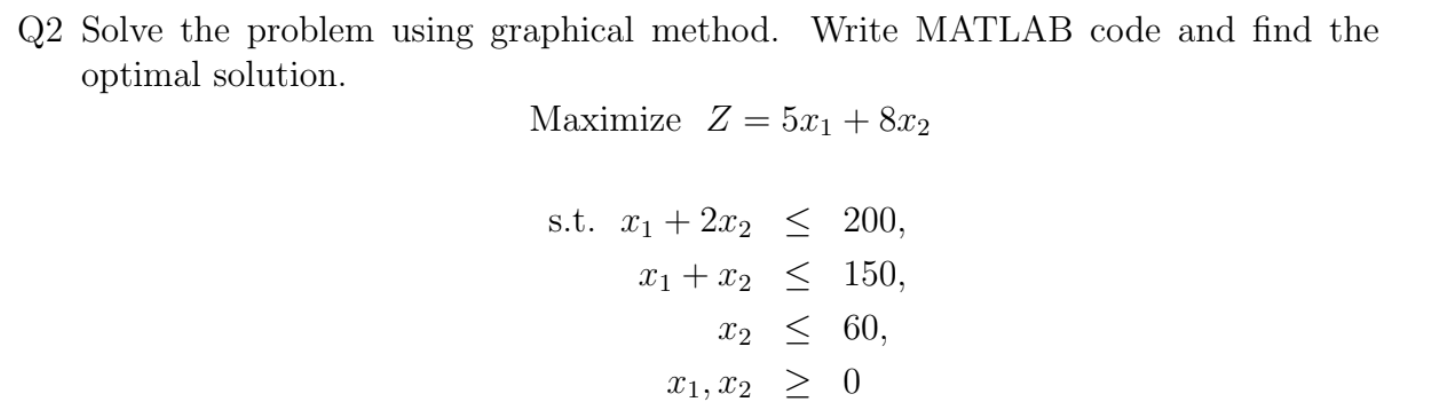
hold on

Output:

****

Graph:



****

Code:

%% QUESTION 2: Graphical method to solve

% Max Z= 5x1+8x2

% 2x1+x2<=200

% x1+2x2<=150

% x2<=60

% x1,x2>=0

clc

clear all

format short

%INPUT PARAMETERS

c=[5,8]; %cost objective function

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

B=[200;150;60;0;0];

% 1 for <=const and -1 for >= const

const=[1;1;1];%for lesser than function we have 1 and -1 for greater than

objective=1;%1 for maximization and -1 for minimization

n=size(A,1);

x1=0:0.01:max(B);

for i=1:n-2 %we take n-2 since we are also taking x1=0 and x2=0 as they have no significance in our graph

y(i,:)=(B(i)-A(i,1)\*x1)/A(i,2);

end

%DRAWING THE LINES

for i=1:n-2

y(i,:)=max(0,y(i,:));

plot(x1,y(i,:),'linewidth',4)

hold on

end

hold on

%FINDING THE POINT OF INTERSECTION

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,:);

A3=[A1;A2];

B3=[B1;B2];

%X3=inv(A3)\*B3

X3=A3\B3;

if(X3>=0)%since the number of chairs can never be negative

pt= [pt X3];

end

end

end

X=pt';

X=unique(X,'rows')%solution

hold on

% KEEP ONLY FEASIBLE POINTS

x1=X(:,1);

x2=X(:,2);

for i=1:n-2 %n=size(A,1)-2

%for greater than(1) equation we remove A\*X'>B and for less than(-1) we do A(i,:)\*X'<B(i)

if(const(i)>0)

ind=find(A(i,:)\*X'>B(i));

X(ind,:)=[];% indexes that are not satisfying the constraint are being replaced with empty set

else

ind=find(A(i,:)\*X'<B(i));

X(ind,:)=[];

end

end

% EVALUATE THE OBJECTIVE FUNCTION VALUE

if(objective == 1)

obj\_val=c\*X';

[value, ind]=max(obj\_val);

fprintf("The max optimal value is : %f \n",value)

fprintf("The max optimal point is : (%g,%g) \n",X(ind,:))

else

obj\_val=c\*X';

[value, ind]=min(obj\_val);

fprintf("The min optimal value is : %f \n",value)

fprintf("The min optimal point is : (%g,%g) \n",X(ind,:))

end

X(ind,:);

Optimal=[X(ind,:) value];

% Shaded feasible region

x=X(:,1);

y=X(:,2);

scatter(X(:,1),X(:,2),'\*')

hold on

k=convhull(x,y);%the shaded region where a and y is satisfied

fill(x(k),y(k),'m')

% setting the axes

xlim([0 max(x)+1])

ylim([0 max(y)+1])

xlabel('x-axis')

ylabel('y-axis')

title('Feasible region of the linear programming problem')

% legend('2x\_1+x\_2\leq104','x\_1+2x\_2\leq76','x\_1,x\_2\geq0')

%phase 7: Verification

x=0:0.1:max(B);

for z=0:8:value

y=(z-c(1)\*x)/c(2);

plot(x,y)

hold on

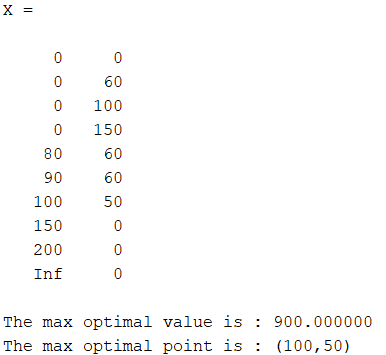
drawnow

pause(0.001)

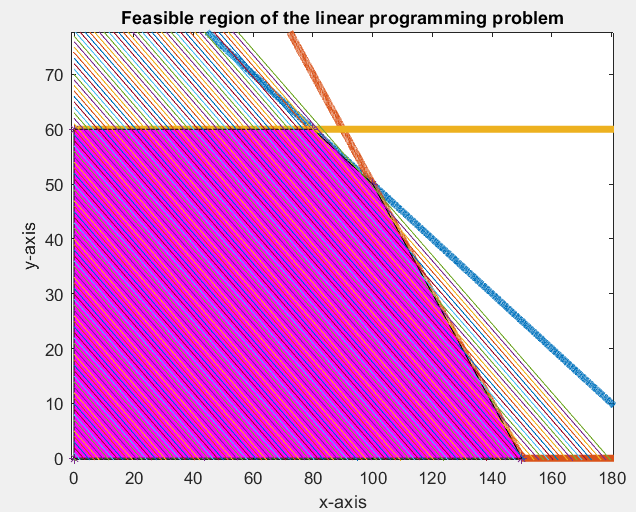
end

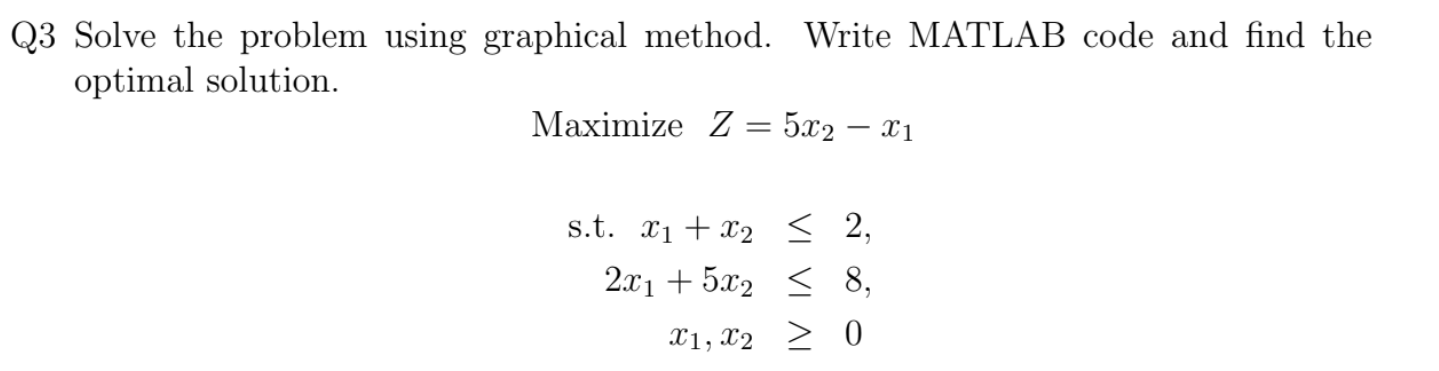
hold on

Output:



Graph:

****

****

Code:

%% QUESTION 3: Graphical method to solve

% Max Z= 5x1-x2

% x1+x2<=2

% 2x1+5x2<=8

% x2>=0

% x1>=0

clc

clear all

format short

%INPUT PARAMETERS

c=[5,-1]; %cost objective function

A=[1,1;2,5;0,1;1,0];

B=[2;8;0;0];

% 1 for <=const and -1 for >= const

const=[1;1];%for lesser than function we have 1 and -1 for greater than

objective=1;%1 for maximization and -1 for minimization

n=size(A,1);

x1=0:0.01:max(B);

for i=1:n-2 %we take n-2 since we are also taking x1=0 and x2=0 as they have no significance in our graph

y(i,:)=(B(i)-A(i,1)\*x1)/A(i,2);

end

%DRAWING THE LINES

for i=1:n-2

y(i,:)=max(0,y(i,:));

plot(x1,y(i,:),'linewidth',4)

hold on

end

hold on

%FINDING THE POINT OF INTERSECTION

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,:);

A3=[A1;A2];

B3=[B1;B2];

%X3=inv(A3)\*B3

X3=A3\B3;

if(X3>=0)%since the number of chairs can never be negative

pt= [pt X3];

end

end

end

X=pt';

X=unique(X,'rows')%solution

hold on

% KEEP ONLY FEASIBLE POINTS

x1=X(:,1);

x2=X(:,2);

for i=1:n-2 %n=size(A,1)-2

%for greater than(1) equation we remove A\*X'>B and for less than(-1) we do A(i,:)\*X'<B(i)

if(const(i)>0)

ind=find(A(i,:)\*X'>B(i));

X(ind,:)=[];% indexes that are not satisfying the constraint are being replaced with empty set

else

ind=find(A(i,:)\*X'<B(i));

X(ind,:)=[];

end

end

% EVALUATE THE OBJECTIVE FUNCTION VALUE

if(objective == 1)

obj\_val=c\*X';

[value, ind]=max(obj\_val);

fprintf("The max optimal value is : %f \n",value)

fprintf("The max optimal point is : (%g,%g) \n",X(ind,:))

else

obj\_val=c\*X';

[value, ind]=min(obj\_val);

fprintf("The min optimal value is : %f \n",value)

fprintf("The min optimal point is : (%g,%g) \n",X(ind,:))

end

X(ind,:);

Optimal=[X(ind,:) value];

% Shaded feasible region

x=X(:,1);

y=X(:,2);

scatter(X(:,1),X(:,2),'\*')

hold on

k=convhull(x,y);%the shaded region where a and y is satisfied

fill(x(k),y(k),'m')

% setting the axes

xlim([0 max(x)+1])

ylim([0 max(y)+1])

xlabel('x-axis')

ylabel('y-axis')

title('Feasible region of the linear programming problem')

% legend('2x\_1+x\_2\leq104','x\_1+2x\_2\leq76','x\_1,x\_2\geq0')

%phase 7: Verification

x=0:0.1:max(B);

for z=0:0.5:value

y=(z-c(1)\*x)/c(2);

plot(x,y)

hold on

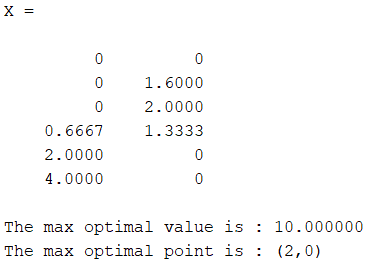
drawnow

pause(0.001)

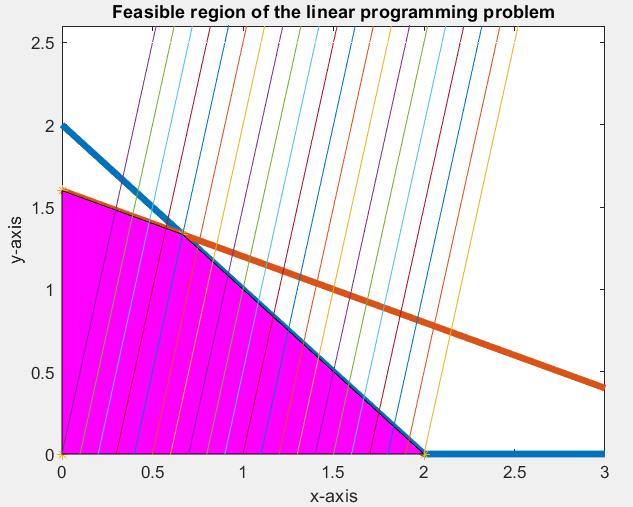
end

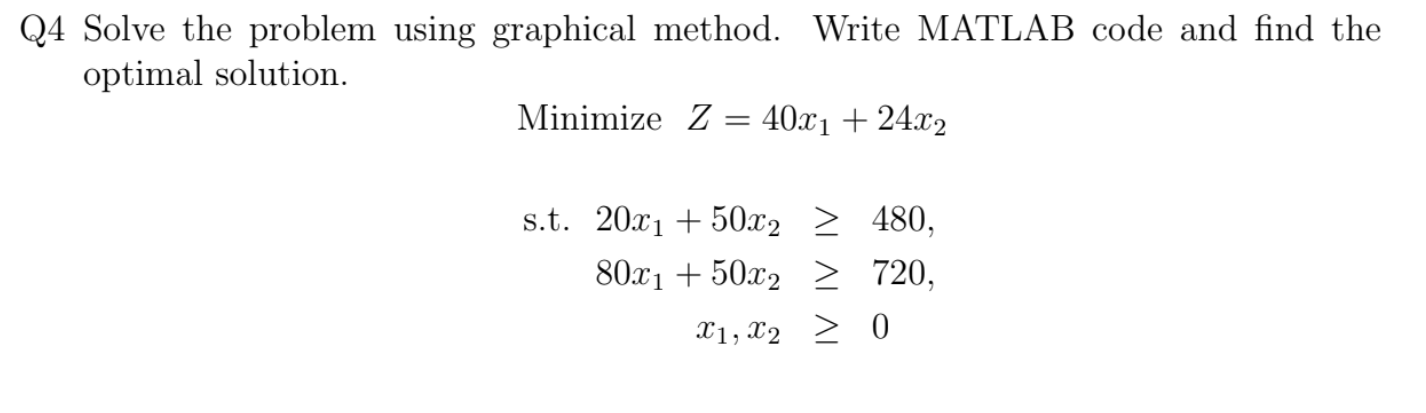
hold on

Output:



Graph:



****

Code:

%% QUESTION 4: Graphical method to solve

% Min Z= 40x1+24x2

% 20x1+50x2>=480

% 80x1+50x2>=720

% x2>=0

% x1>=0

clc

clear all

format short

%INPUT PARAMETERS

c=[40,24]; %cost objective function

A=[20,50;80,50;0,1;1,0];

B=[480;720;0;0];

% 1 for <=const and -1 for >= const

const=[-1;-1];%for lesser than function we have 1 and -1 for greater than

objective=-1;%1 for maximization and -1 for minimization

n=size(A,1);

x1=0:0.01:max(B);

for i=1:n-2 %we take n-2 since we are also taking x1=0 and x2=0 as they have no significance in our graph

y(i,:)=(B(i)-A(i,1)\*x1)/A(i,2);

end

%DRAWING THE LINES

for i=1:n-2

y(i,:)=max(0,y(i,:));

plot(x1,y(i,:),'linewidth',4)

hold on

end

hold on

%FINDING THE POINT OF INTERSECTION

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,:);

A3=[A1;A2];

B3=[B1;B2];

%X3=inv(A3)\*B3

X3=A3\B3;

if(X3>=0)%since the number of chairs can never be negative

pt= [pt X3];

end

end

end

X=pt';

X=unique(X,'rows')%solution

hold on

% KEEP ONLY FEASIBLE POINTS

x1=X(:,1);

x2=X(:,2);

for i=1:n-2 %n=size(A,1)-2

%for greater than(1) equation we remove A\*X'>B and for less than(-1) we do A(i,:)\*X'<B(i)

if(const(i)>0)

ind=find(A(i,:)\*X'>B(i));

X(ind,:)=[];% indexes that are not satisfying the constraint are being replaced with empty set

else

ind=find(A(i,:)\*X'<B(i));

X(ind,:)=[];

end

end

% EVALUATE THE OBJECTIVE FUNCTION VALUE

if(objective == 1)

obj\_val=c\*X';

[value, ind]=max(obj\_val);

fprintf("The max optimal value is : %f \n",value)

fprintf("The max optimal point is : (%g,%g) \n",X(ind,:))

else

obj\_val=c\*X';

[value, ind]=min(obj\_val);

fprintf("The min optimal value is : %f \n",value)

fprintf("The min optimal point is : (%g,%g) \n",X(ind,:))

end

X(ind,:);

Optimal=[X(ind,:) value];

% Shaded feasible region

x=X(:,1);

y=X(:,2);

scatter(X(:,1),X(:,2),'\*')

hold on

k=convhull(x,y);%the shaded region where a and y is satisfied

fill(x(k),y(k),'m')

% setting the axes

xlim([0 max(x)+1])

ylim([0 max(y)+1])

xlabel('x-axis')

ylabel('y-axis')

title('Feasible region of the linear programming problem')

% legend('2x\_1+x\_2\leq104','x\_1+2x\_2\leq76','x\_1,x\_2\geq0')

%phase 7: Verification

x=0:0.1:max(B);

for z=0:8:value

y=(z-c(1)\*x)/c(2);

plot(x,y)

hold on

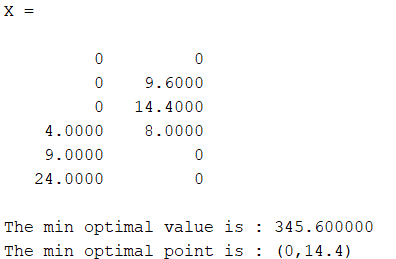
drawnow

pause(0.001)

end

hold on

Output:



Graph:

