BFS

Max. z = x1 + 2x2, subject to − x1 + x2 ≤ 1, x1 + x2 ≤ 2, x1, x2 ≥ 0.

A1)

%To find the basic feasible solutions

clc

clear all

format short

A=[-1 1 1 0; 1 1 0 1] %Coefficient Matrix

b=[1; 2] %For Right Hand Side

C=[1 2 0 0] %Cost Coefficients

n=size(A,2) %Number of variables

m=size(A,1) %Number of constraints

if(n>m)

nCm=nchoosek(n,m) %Total number of Basic Solutions

pair=nchoosek(1:n,m)%Pair of basic variables

sol=[] %Default sol is zero

for i=1:nCm

y=zeros(n,1)

x=A(:,pair(i,:))\b

%To check the feasibility condition

if all(x>=0 & x~=inf & x~=-inf)

y(pair(i,:))= x

sol=[sol,y]

end

end

else

error('nCm does not exist.')

end

% To find the objective function value

Z=C\*sol

%To find the optimal solution

[Zmax, Zindex]=max(Z)

BFS=sol(:,Zindex)

%Phase VII - To print all the solutions

optimal\_value=[BFS' Zmax]

optimal\_BFS=array2table(optimal\_value)

optimal\_BFS.Properties.VariableNames(1:size(optimal\_BFS,2))={'x1','x2','s1','s2','Z'}

GRAPhical

q1)

Maximize/Minimize (3x1 + 2x2) Subject to 2x1 +4x2< = 8, 3x1 + 5x2>= 15, x1>=0, x2>= 0.

Ans)

%% to solve the graphical method for a lpp

clearvars;

clc;

A = [2,4;3,5;1,0 ; 0,1]

B=[8;15;0;0]

C=[3,2]'

x =0:max(B)

%now find y from all the constraint

y1 = (B(1)-A(1,1)\*x)/A(1,2)

y2 = (B(2)-A(2,1)\*x)/A(2,2)

y3 = (B(3)-A(3,1)\*x)/A(3,2)

y1= max(0,y1)

y2= max(0,y2)

y3= max(0,y3)

plot(x,y1,'r--',x,y2,'b--',x,y3,'g--')

title('GRAPHICAL METHOD')

xlabel('x axis')

ylabel('y axis')

%% find the intersection of every two lines

sol=[]

[m,n] = size(A)

for i=1:m

A1 = A(i,:)

B1 = B(i,:)

for j=i+1:m

A2=A(j,:)

B2=B(j,:)

A3 = [A1;A2]

B3 = [B1;B2]

X = inv(A3)\*B3

sol = [sol,X]

end

end

finalsol=[sol]'

%% now find the feasible solution

x = finalsol(:,1)

y = finalsol(:,2)

%now we will find those rows where the values doesn't satisfy the 1st constraint

H1= find(2\*x+(4\*(y))-8>0)

%delete those rows

finalsol(H1,:)=[]

%again getting the possible set of values

x = finalsol(:,1)

y = finalsol(:,2)

%now we will find those rows where the values doesn't satisfy the 2nd constraint

H2= find(3\*x+5\*y-15<0)

%delete those rows

finalsol(H2,:)=[]

%again getting the possible set of values

x = finalsol(:,1)

y = finalsol(:,2)

%now we will find those rows where the values doesn't satisfy the 4th constraints

H3= find(x<0)

%delete those rows

finalsol(H3,:)=[]

%again getting the possible set of values

x = finalsol(:,1)

y = finalsol(:,2)%now we will find those rows where the values doesn't satisfy the constraints

H4= find(y<0)

%delete those rows

finalsol(H4,:)=[]

%again getting the possible set of values

x = finalsol(:,1);

y = finalsol(:,2)

%% To find the optimal solution

for k= 1:size(finalsol,1)

Obj(k,:)=sum(finalsol(k,:).\*C')

end

%find the maximum and minimum of objective function

A = max(Obj)

B = min(Obj)

%FIND THAT ROW CORROSPONDING TO WHICH OBJ IS MAX OR MIN

P = find(Obj==A)

Q = find(Obj==B)

%FIND THE VALUES OF THE ROW THAT REPRESENTS THE OPTIMAL SOLUTION

OS1 = finalsol(P,:)

OS2 = finalsol(Q,:)

**Simplex**

Simplex Method %max z=2x1+5X2 %x1+4x2<=24 %3x1+1x2<=21 %x1+x2<=9 clc clear all format short Noofvariables = 2; C = [2 5]; a = [1 4; 3 1; 1 1]; b = [24; 21; 9]; 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', 's\_3', 'sol'} RUN = true; while RUN if any(zjcj < 0); %check for (most) negative value 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', p vt\_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', 's\_3', 'sol'} else RUN = false; fprintf('The current BFS is optimal \n') end end

**BIG M**

1. M in. z = 3x1 + 5x2, S.T. x1 + 3x2 ≥ 3, x1 + x2 ≥ 2, x1, x2 ≥ 0.

clc

clear

M=1000;

art\_v=[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]; % extra zero for 'b'

a=[A b];

%================================================

array2table(a,'VariableNames',{'x1','x2','s1','s2','A1','A2','b'});

bvar=[5 6];

z=c(bvar)\*a-c;

simp\_table=[z;a];

Var={'x1','x2','s1','s2','A1','A2','b'};

array2table(simp\_table,'VariableNames',Var)

%=====================================================

for j=1:15

if all(z(1:end-1)>=0)

if any(bvar==art\_v(1))

fprintf('Infeasible solution');

break;

end

fprintf('Current table is optimal\n');

opt\_value=z(end);

fprintf('The optimal value of the current lpp is %f',opt\_value);

break;

else

fprintf('Current table is not optimal');

[entering\_variable\_value, pvt\_col]=min(z(1:end-1));

if all(a(:,pvt\_col)<=0)

error('The given lpp is unbounded');

else

solution=a(:,end);

column=a(:,pvt\_col);

for i=1:size(a,1)

if column(i)>0

ratio(i)=solution(i)/column(i);

else

ratio(i)=inf;

end

end

[leaving\_variable\_value,pvt\_row]=min(ratio);

bvar(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 % ~ not equal to

a(i,:)=a(i,:)-a(i,pvt\_col)\*a(pvt\_row,:);

end

end

z=c(bvar)\*a-c;

simp\_table=[z;a];

array2table(simp\_table,'VariableNames',Var)

end

end

end

**2phase**

Min. z = 12x1 + 10x2, S.T. 5x1 + x2 ≥ 10, 6x1 + 5x2 ≥ 30, x1 + 4x2 ≥ 8, x1, x2 ≥ 0

clc

clear all

format short

Vars={'x1','x2','s1','s2','s3','A1','A2','A3','sol'};

OVars={'x1','x2','s1','s2','s3','sol'}; % original variables

OriginalC=[-12 -10 0 0 0 -1 -1 -1 0];

a=[5 1 -1 0 0 1 0 0; 6 5 0 -1 0 0 1 0; 1 4 0 0 -1 0 0 1];

b=[10;30;8];

A=[a b];

% ===========PHASE-1==========================

fprintf('================== PHASE-1 =============\n')

cost=[0 0 0 0 0 -1 -1 -1 0]

Art\_var=[6 7 8]

bvars=[6 7 8];

zjcj=cost(bvars)\*A-cost;

simplex\_table=[zjcj;A];

array2table(simplex\_table,'VariableNames',Vars)

RUN=true;

while RUN

if any(zjcj(1:end-1)<0) %check for negative value

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, pivot\_row]=min(ratio);

end

bvars(pivot\_row)=pvt\_col;

pvt\_key=A(pivot\_row, pvt\_col);

A(pivot\_row,:)=A(pivot\_row,:)./pvt\_key;

for i=1:size(A,1)

if i~=pivot\_row

A(i,:)=A(i,:)-A(i, pvt\_col).\*A(pivot\_row,:);

end

end

zjcj=cost(bvars)\*A-cost;

zcj=[zjcj;A];

table=array2table(zcj,'VariableNames',Vars)

else

RUN=false;

if any(bvars==Art\_var(1)) || any(bvars==Art\_var(2))

error('Infeasible solution');

else

fprintf('optimal table of phase-1 is achieved \n');

end

end

end

%PHASE-2

fprintf('\*\*\*\* PHASE-2 \*\*\*\* \n')

A(:,Art\_var)=[]; %Removing Artificial var by giving them empty value

OriginalC(:,Art\_var)=[]; %Removing Artificial var cost by giving them empty value

cost=OriginalC;

zjcj=cost(bvars)\*A-cost;

simplex\_table=[zjcj;A];

array2table(simplex\_table,'VariableNames',OVars)

RUN=true;

while RUN

if any(zjcj(1:end-1)<0) %check for negative value

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, pivot\_row]=min(ratio);

end

bvars(pivot\_row)=pvt\_col;

pvt\_key=A(pivot\_row, pvt\_col);

A(pivot\_row,:)=A(pivot\_row,:)./pvt\_key;

for i=1:size(A,1)

if i~=pivot\_row

A(i,:)=A(i,:)-A(i, pvt\_col).\*A(pivot\_row,:);

end

end

zjcj=cost(bvars)\*A-cost;

zcj=[zjcj;A];

table=array2table(zcj,'VariableNames',OVars)

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