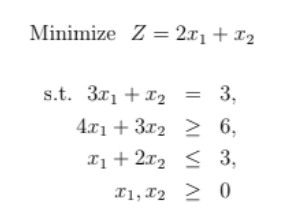
**OT LAB ASSIGNMENT 6**

**Question 1:**



**Code:**

%% QUESTION 1

% Solve LPP using simplex using Simplex Algorithm with Big-M method

% Minimize Z = 2x1+x2

% s.t. 3x1 + x2 = 3

% 4x1 + 3x2 >= 6

% x1 + 2x2 <= 3

% xi >= 0 i=1-3

% Maximize Z = -2x1-x2-Ma1-Ma2

% s.t. 3x1 + x2 +a1 = 3

% 4x1 + 3x2 -s2 + a2 = 6

% x1 + 2x2 +s3 = 3

% xi >= 0 i=1-3

clc

clear all

format short

% Input Phase

Variables = {'x1','x2','s2','s3','a1','a2','Sol'};

M=1000;

Cost = [-2,-1,0,0,-M,-M,0];

a=[3,1,0,0,1,0; 4,3,-1,0,0,1;1,2,0,1,0,0];

b=[3;6;3];

A=[a b];

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

%FINDING STARTING BFS

BV=[];

for j=1:size(s,2)

for i=1:size(A,2)

if A(:,i)==s(:,j)

BV=[BV i];

end

end

end

% COMPUTE VALUE OF TABLE

B= A(:,BV);

A= inv(B)\*A;

ZjCj= Cost(BV)\*A-Cost;

% TO PRINT THE TABLE

fprintf('Simplex Table to solve: \n')

ZCj = [ZjCj;A];

SimpTable = array2table(ZCj);

SimpTable.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(SimpTable);

% SIMPLEX METHOD START

RUN =true;

while RUN

ZC = ZjCj(:,1:end-1);

if any(ZC<0)

fprintf('Current BFS is NOT OPTIMAL\n');

[Entval,pvt\_col]=min(ZC);

fprintf('Entering Column = %d \n',pvt\_col);

%finding leaving var

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)=sol(i)./Column(i);

else

ratio(i)=inf;

end

end

[minR, pvt\_row]=min(ratio);

fprintf('Leaving Row = %d\n',pvt\_row);

% UPDATE THE BV & TABLE

BV(pvt\_row)=pvt\_col;

B=A(:,BV);

A= inv(B)\*A;

ZjCj = Cost(BV)\*A-Cost;

%to print intermediate table

fprintf('Table after iteration: \n')

ZCj = [ZjCj;A];

TABLE = array2table(ZCj);

TABLE.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(TABLE);

end

else

RUN = false;

fprintf('CURRENT BFS IS OPTIMAL \n');

end

end

%FINAL OPTIMAL SOLUTION PRINT:

% TO PRINT THE TABLE

FINAL\_BFS= zeros(1,size(A,2));

FINAL\_BFS(BV) = A(:,end);

FINAL\_BFS(end) = sum(FINAL\_BFS.\*Cost);

% TO PRINT THE TABLE

OptimalBFS = array2table(FINAL\_BFS);

OptimalBFS.Properties.VariableNames(1:size(OptimalBFS,2))=Variables;

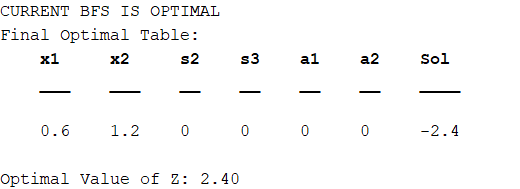
fprintf("Final Optimal Table:\n");

disp(OptimalBFS);

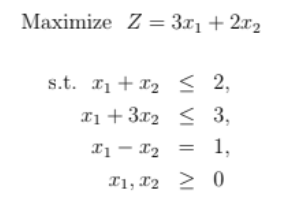
val=-(FINAL\_BFS(end));

fprintf("Optimal Value of Z: %0.2f\n",val);

**Output:**

****

**Question 2:**



**Code:**

%% QUESTION 2

% Solve LPP using simplex using Simplex Algorithm with Big-M method

% Maximize Z = 3x1+2x2+0s1+0s2-Ma3

% s.t. x1 + x2 + s1 = 2

% x1 + 3x2 + s2 = 3

% x1 - x2 + a3 = 1

% xi >= 0 i=1-3

clc

clear all

format short

% Input Phase

Variables = {'x1','x2','s1','s2','a3','Sol'};

M=1000;

Cost = [3,2,0,0,-M,0];

a=[1,1,1,0,0; 1,3,0,1,0;1,-1,0,0,1];

b=[2;3;1];

A=[a b];

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

%FINDING STARTING BFS

BV=[];

for j=1:size(s,2)

for i=1:size(A,2)

if A(:,i)==s(:,j)

BV=[BV i];

end

end

end

% COMPUTE VALUE OF TABLE

B= A(:,BV);

A= inv(B)\*A;

ZjCj= Cost(BV)\*A-Cost;

% TO PRINT THE TABLE

fprintf('The simplex table: \n')

ZCj = [ZjCj;A];

SimpTable = array2table(ZCj);

SimpTable.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(SimpTable)

% SIMPLEX METHOD START

RUN =true;

while RUN

ZC = ZjCj(:,1:end-1);

if any(ZC<0)

fprintf('Current BFS is NOT OPTIMAL\n');

[Entval,pvt\_col]=min(ZC);

fprintf('Entering Column = %d \n',pvt\_col);

%finding leaving var

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)=sol(i)./Column(i);

else

ratio(i)=inf;

end

end

[minR, pvt\_row]=min(ratio);

fprintf('Leaving Row = %d\n',pvt\_row);

% UPDATE THE BV & TABLE

BV(pvt\_row)=pvt\_col;

B=A(:,BV);

A= inv(B)\*A;

ZjCj = Cost(BV)\*A-Cost;

%to print intermediate table

ZCj = [ZjCj;A];

fprintf('Table after iteration: \n')

TABLE = array2table(ZCj);

TABLE.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(TABLE)

end

else

RUN = false;

fprintf('CURRENT BFS IS OPTIMAL \n');

end

end

%FINAL OPTIMAL SOLUTION PRINT:

% TO PRINT THE TABLE

FINAL\_BFS= zeros(1,size(A,2));

FINAL\_BFS(BV) = A(:,end);

FINAL\_BFS(end) = sum(FINAL\_BFS.\*Cost);

% TO PRINT THE TABLE

OptimalBFS = array2table(FINAL\_BFS);

OptimalBFS.Properties.VariableNames(1:size(OptimalBFS,2))=Variables;

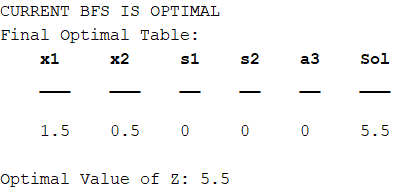
fprintf("Final Optimal Table:\n");

disp(OptimalBFS);

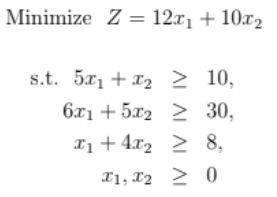
val=(FINAL\_BFS(end));

fprintf("Optimal Value of Z: %0.1f\n",val);

**Output:**



**Question 3:**



**Code:**

%% QUESTION 3

% Solve LPP using simplex using Simplex Algorithm with Big-M method

% Minimize Z = 12x1+10x2

% s.t. 5x1 + x2 >= 10

% 6x1 + 5x2 >= 30

% x1 + 4x2 >= 8

% xi >= 0 i=1-3

% Maximize Z = -12x1-10x2-Ma1-Ma2-Ma3

% s.t. 5x1 + x2 - s1 + a1 = 10

% 6x1 + 5x2 - s2 + a2 = 30

% x1 + 4x2 - s3 + a3 = 8

% xi >= 0 i=1-3

clc

clear all

format short

% Input Phase

Variables = {'x1','x2','s1','s2','s3','a1','a2','a3','Sol'};

M=1000;

Cost = [-12,-10,0,0,0,-M,-M,-M,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];

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

%FINDING STARTING BFS

BV=[];

for j=1:size(s,2)

for i=1:size(A,2)

if A(:,i)==s(:,j)

BV=[BV i];

end

end

end

% COMPUTE VALUE OF TABLE

B= A(:,BV);

A= inv(B)\*A;

ZjCj= Cost(BV)\*A-Cost;

% TO PRINT THE TABLE

fprintf('Simplex Table to solve: \n')

ZCj = [ZjCj;A];

SimpTable = array2table(ZCj);

SimpTable.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(SimpTable);

% SIMPLEX METHOD START

RUN =true;

while RUN

ZC = ZjCj(:,1:end-1);

if any(ZC<0)

fprintf('Current BFS is NOT OPTIMAL\n');

[Entval,pvt\_col]=min(ZC);

fprintf('Entering Column = %d \n',pvt\_col);

%finding leaving var

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)=sol(i)./Column(i);

else

ratio(i)=inf;

end

end

[minR, pvt\_row]=min(ratio);

fprintf('Leaving Row = %d\n',pvt\_row);

% UPDATE THE BV & TABLE

BV(pvt\_row)=pvt\_col;

B=A(:,BV);

A= inv(B)\*A;

ZjCj = Cost(BV)\*A-Cost;

%to print intermediate table

fprintf('Table after iteration: \n')

ZCj = [ZjCj;A];

TABLE = array2table(ZCj);

TABLE.Properties.VariableNames(1:size(ZCj,2))=Variables;

disp(TABLE);

end

else

RUN = false;

fprintf('CURRENT BFS IS OPTIMAL \n');

end

end

%FINAL OPTIMAL SOLUTION PRINT:

% TO PRINT THE TABLE

FINAL\_BFS= zeros(1,size(A,2));

FINAL\_BFS(BV) = A(:,end);

FINAL\_BFS(end) = sum(FINAL\_BFS.\*Cost);

% TO PRINT THE TABLE

OptimalBFS = array2table(FINAL\_BFS);

OptimalBFS.Properties.VariableNames(1:size(OptimalBFS,2))=Variables;

fprintf("Final Optimal Table:\n");

disp(OptimalBFS);

val=-(FINAL\_BFS(end));

fprintf("Optimal Value of Z: %0.0f\n",val);

**Output:**

