% Maximize (3x1 + 2x2+4x3)

% Maximize (x1 + 5x2+3x3)

% Subject to

% 2x1 +4x2 +x3< = 8,

% 3x1 + 5x2 +4x3>= 15,

% x1>=0, x2>= 0, x3>= 0

format short;

clear all;

clc;

M = 10000;

Variables ={'x\_1','x\_2','x\_3','s\_1','s\_2','A\_3','sol'};

A = [2 4 3 1 0 0 8; 3 5 4 0 1 1 15];

Cost1 = [3 2 1 0 0 -M 0];

Cost2 = [1 5 3 0 0 -M 0];

Cost = 0.5\*Cost1+0.5\*Cost2;

BV = [4 5];

ZjCj=Cost(BV)\*A-Cost

zcj=[Cost;ZjCj;A];

zcj

bigmtable=array2table(zcj);

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

RUN= true;

while RUN

ZC=ZjCj(1:end-1)

if any(ZC<0)

fprintf(' The current BFS is not optimal\n')

[ent\_col,pvt\_col]=min(ZC)

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

sol=A(:,end)

Column=A(:,pvt\_col)

if Column<=0

error('LPP is unbounded');

else

for i=1:size(A,1)

if Column(i)>0

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

else

ratio(i)=inf

end

end

[MinRatio,pvt\_row]=min(ratio)

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

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);

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

RUN=false;

fprintf(' Current BFS is Optimal \n');

end

end

BFS=zeros(1,size(A,2));

BFS(BV)=A(:,end);

BFS(end)=sum(BFS.\*Cost);

Current\_BFS=array2table(BFS);

Current\_BFS.Properties.VariableNames(1:size(Current\_BFS,2))=Variables;