

OT LAB ASSIGNMENT 6

Question 1:

$$\text{Minimize } Z = 2x_1 + x_2$$

$$\text{s.t. } 3x_1 + x_2 = 3,$$

$$4x_1 + 3x_2 \geq 6,$$

$$x_1 + 2x_2 \leq 3,$$

$$x_1, x_2 \geq 0$$

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: %.2f\n",val);
```

Output:

```
CURRENT BFS IS OPTIMAL
```

```
Final Optimal Table:
```

x1	x2	s2	s3	a1	a2	Sol
0.6	1.2	0	0	0	0	-2.4

```
Optimal Value of Z: 2.40
```

Question 2:

Maximize $Z = 3x_1 + 2x_2$

s.t. $x_1 + x_2 \leq 2,$

$x_1 + 3x_2 \leq 3,$

$x_1 - x_2 = 1,$

$x_1, x_2 \geq 0$

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:

```
CURRENT BFS IS OPTIMAL
Final Optimal Table:

    x1    x2    s1    s2    a3    Sol
    ---    ---    ---    ---    ---    ---
    1.5    0.5    0     0     0     5.5

Optimal Value of Z: 5.5
```

Question 3:

```
Minimize  $Z = 12x_1 + 10x_2$ 

s.t.  $5x_1 + x_2 \geq 10,$ 
 $6x_1 + 5x_2 \geq 30,$ 
 $x_1 + 4x_2 \geq 8,$ 
 $x_1, x_2 \geq 0$ 
```

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;
        end
    end
end
```

```

        %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:

```
CURRENT BFS IS OPTIMAL
```

```
Final Optimal Table:
```

x1	x2	s1	s2	a3	Sol
—	—	—	—	—	—
1.5	0.5	0	0	0	5.5

```
Optimal Value of Z: 5.5
```