BIG-M METHOD

Contents

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);
```

```
Simplex Table to solve:
     x1       x2       s2     s3    a1    a2     Sol
    _____    _____    ____    __    __    __    _____

    -6998    -3999    1000    0     0     0     -9000
        3        1       0    0     1     0         3
        4        3      -1    0     0     1         6
        1        2       0    1     0     0         3

Current BFS is NOT OPTIMAL
Entering Column = 1
Leaving Row = 1
```

```
Table after iteration:
    x1      x2        s2     s3       a1       a2     Sol
    __    _______    ____    __    ________    __    _____

    0     -1666.3    1000    0       2332.7    0     -2002
    1     0.33333       0    0      0.33333    0        1
    0      1.6667      -1    0      -1.3333    1        2
    0      1.6667       0    1     -0.33333    0        2

Current BFS is NOT OPTIMAL
Entering Column = 2
Leaving Row = 2
Table after iteration:
    x1        x2          s2     s3     a1       a2      Sol
    __    ___________    ____    __    _____    _____    ____

    0              0     0.2     0     999.6    999.8    -2.4
    1     -5.5511e-18    0.2     0       0.6     -0.2     0.6
    0              1    -0.6     0      -0.8      0.6     1.2
    0              0       1     1         1       -1       0

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

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);
```

```
The simplex table:
     x1      x2     s1    s2    a3     Sol
    _____    ___    __    __    __    _____

    -1003    998    0     0     0     -1000
       1       1    1     0     0        2
       1       3    0     1     0        3
       1      -1    0     0     1        1

Current BFS is NOT OPTIMAL
Entering Column = 1
Leaving Row = 3
Table after iteration:
    x1    x2    s1    s2     a3     Sol
    __    __    __    __    ____    ___

    0     -5    0     0     1003     3
    0      2    1     0       -1     1
    0      4    0     1       -1     2
    1     -1    0     0        1     1

Current BFS is NOT OPTIMAL
Entering Column = 2
Leaving Row = 1
Table after iteration:
    x1    x2    s1     s2      a3      Sol
    __    __    ___    __    ______    ___

    0     0     2.5    0     1000.5    5.5
    0     1     0.5    0       -0.5    0.5
    0     0      -2    1          1      0
    1     0     0.5    0        0.5    1.5

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

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,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);
```

```
Simplex Table to solve:
      x1       x2       s1      s2      s3     a1    a2    a3     Sol
    _____    _____    ____    ____    ____    __    __    __    _____

    -11988    -9990    1000    1000    1000    0     0     0     -48000
         5        1      -1       0       0    1     0     0         10
         6        5       0      -1       0    0     1     0         30
         1        4       0       0      -1    0     0     1          8

Current BFS is NOT OPTIMAL
Entering Column = 1
Leaving Row = 1
Table after iteration:
        x1           x2         s1        s2      s3       a1      a2    a3     Sol
    ___________    _______    _____    ____    ____    _____    __    __    _____

     8.8818e-13    -7592.4    -1397.6    1000    1000    2397.6    0     0     -24024
              1        0.2       -0.2       0       0       0.2    0     0          2
    -8.8818e-16        3.8        1.2      -1       0      -1.2    1     0         18
              0        3.8        0.2       0      -1      -0.2    0     1          6

Current BFS is NOT OPTIMAL
Entering Column = 2
Leaving Row = 3
Table after iteration:
    x1        x2            s1        s2        s3          a1        a2       a3         Sol
    __    __________    ________    ____    ________    _________    __    ________    _____
```

```
    0     -4.1922e-13        -998    1000        -998         1998    0         1998    -12036
    1      3.1554e-17    -0.21053       0    0.052632      0.21053    0    -0.052632    1.6842
    0      4.2188e-16           1      -1           1           -1    1           -1         12
    0               1    0.052632       0    -0.26316    -0.052632    0      0.26316    1.5789

Current BFS is NOT OPTIMAL
Entering Column = 3
Leaving Row = 2
Table after iteration:
    x1        x2             s1            s2          s3          a1          a2          a3          Sol
    __    ___________    ___________    ________    ________    _________    _________    ________    _______

    0              0     9.7936e-17           2           0          1000          998        1000         -60
    1     -1.1288e-32    -6.3353e-18    -0.21053     0.26316    6.3353e-18      0.21053    -0.26316      4.2105
    0     -5.1769e-32              1          -1           1           -1            1          -1           12
    0               1    -2.1912e-18    0.052632    -0.31579    2.1912e-18    -0.052632     0.31579     0.94737

CURRENT BFS IS OPTIMAL
Final Optimal Table:
      x1        x2       s1    s2    s3    a1    a2    a3    Sol
    _____    _____    ___    ___    ___    ___    ___    ___    ____

    4.2105    0.94737    12    0     0     0     0     0     -60

Optimal Value of Z: 60
```

Published with MATLAB® R2023b