SIMPLEX METHOD

Contents

QUESTION-1

Solve LPP using simplex using Simplex Algo

```
%MAx Z= x1+2x2
% s.t.  -x1+x2<=1
%        x1+x2<=2
%       xi>=0  i=1-2
% Phase-T: Input the Parameter
clc
clear all
Noofvariables=2;
variables={'x1','x2','s1','s2','sol'};
c=[1 2]; % cost of objective function
Abar=[-1 1;1 1];% const coeff
B=[1;2]; %RHS of constraints
s=eye(size(Abar,1));
A=[Abar s B];
Cost=zeros(1,size(A,2));
Cost(1:Noofvariables)=c;
% Contraints BV
BV=Noofvariables+1:1:size(A,2)-1;
% To calculate Zj-Cj
ZjCj=Cost(BV)*A-Cost;
% For printing 1st simplex table
ZCj=[ZjCj;A];
simplextable=array2table(ZCj);
simplextable.Properties.VariableNames(1:size(ZCj,2))=variables;
% Start simplex Algorithm
Run=true;
while Run
    if any(ZjCj<0) % to check if any negative value there
        fprintf('The current BFS is not optimal\n')
        fprintf('Next iteration required \n')
        disp('Old basic variable (BV)=')
        disp(BV)
        % For finding entering variable
        Zc=ZjCj(1:end-1);
        [Ent_col pvt_col]=min(Zc);
        fprintf('The most negative value in Zj-Cj row is %d and coresponding to column %d \n',Ent_col,pvt_col)
        fprintf('Entering variable is %d \n',pvt_col)
        %For finding the leaving variable
        sol=A(:,end);
        column=A(:,pvt_col);
        if all(column<=0)
            error('The LPP has unbounded solution \n since all enteries are <=0 in %d \n',pvt_col)
        else
            for i=1:size(column,1)
                if column(i)>0
```

```
                    ratio(i)=sol(i)./column(i)
                else
                    ratio(i)=inf
                end
            end
            % To finding minimmum ratio
            [minratio pvt_row]=min(ratio);
            fprintf('The minimum ratio corresponding to pivot row %d \n ',pvt_row)
            fprintf('leaving variable is %d \n ',BV(pvt_row))
            BV(pvt_row)=pvt_col;
            disp('New basic variable(BV)==')
            disp(BV)
            pvt_key=A(pvt_row,pvt_col)
            % To update table for next iteration
            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
                ZjCj=ZjCj-ZjCj(pvt_col).*A(pvt_row,:);
            end

                 end
    else
        Run= false;
        ZCj=[ZjCj;A]
        FinalTable=array2table(ZCj);
        FinalTable.Properties.VariableNames(1:size(ZCj,2))=variables
        FinalTable.Properties.RowNames(1:size(ZCj,1))={'Zj-Cj','x1','x2'}
        BFS=zeros(1,size(A,2));
        BFS(BV)=A(:,end)
        BFS(end)=sum(BFS.*Cost);
        currentBFS=array2table(BFS);
        currentBFS.Properties.VariableNames(1:size(currentBFS,2))={'x1','x2','s1','s2','Opt.Val of Z'}
        disp('Optimal sol is reached')
    end
end
```

```
The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     3     4

The most negative value in Zj-Cj row is -2 and coresponding to column 2
Entering variable is 2

ratio =

     1



ratio =

     1     2

The minimum ratio corresponding to pivot row 1
 leaving variable is 3
 New basic variable(BV)==
     2     4
```

```
pvt_key =

     1


A =

    -1     1     1     0     1
     1     1     0     1     2

The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     2     4

The most negative value in Zj-Cj row is -3 and coresponding to column 1
Entering variable is 1

ratio =

   Inf     2


ratio =

       Inf    0.5000

The minimum ratio corresponding to pivot row 2
 leaving variable is 4
 New basic variable(BV)==
     2     1


pvt_key =

     2


A =

   -1.0000    1.0000    1.0000         0    1.0000
    1.0000         0   -0.5000    0.5000    0.5000


ZCj =

         0         0    0.5000    1.5000    3.5000
         0    1.0000    0.5000    0.5000    1.5000
    1.0000         0   -0.5000    0.5000    0.5000


FinalTable =

  3×5 table

    x1    x2     s1     s2     sol
    __    __    ____    ___    ___
```

```
    0     0      0.5    1.5    3.5
    0     1      0.5    0.5    1.5
    1     0     -0.5    0.5    0.5


FinalTable =

  3×5 table

             x1    x2     s1      s2     sol
             __    __    ____    ___    ___

    Zj-Cj    0     0      0.5    1.5    3.5
    x1       0     1      0.5    0.5    1.5
    x2       1     0     -0.5    0.5    0.5


BFS =

    0.5000    1.5000         0         0         0


currentBFS =

  1×5 table

    x1     x2     s1    s2    Opt.Val of Z
    ___    ___    __    __    ____________

    0.5    1.5    0     0         3.5

Optimal sol is reached
```

QUESTION-2

```
%Solve LPP using simplex using Simplex Algo
% Min Z= x1-3x2+2x3
% s.t.   3x1-x2+2x3<=7
%        -2x1+4x2<=12
%        -4x1+3x2+8x3<=10
%        xi>=0  i=1-3
% Phase-T: Input the Parameter
clc
clear all
Noofvariables=3;
variables={'x1','x2','x3','s1','s2','s3','sol'};
c=[-1 3 -2]; % cost of objective func
Abar=[3 -1 2;-2 4 0;-4 3 8];% const coeff
B=[7;12;10]; %RHS of constraints
s=eye(size(Abar,1));
A=[Abar s B];
Cost=zeros(1,size(A,2));
Cost(1:Noofvariables)=c;
% Contraints BV
BV=Noofvariables+1:1:size(A,2)-1;
% To calculate Zj-Cj
ZjCj=Cost(BV)*A-Cost;
% For printing 1st simplex table
ZCj=[ZjCj;A];
```

```
simplextable=array2table(ZCj);
simplextable.Properties.VariableNames(1:size(ZCj,2))=variables;
% Start simplex Algorithm
Run=true;
while Run
    if any(ZjCj<0) % to check if any negative value there
        fprintf('The current BFS is not optimal\n')
        fprintf('Next iteration required \n')
        disp('Old basic variable (BV)=')
        disp(BV)
        % For finding entering variable
        Zc=ZjCj(1:end-1);
        [Ent_col pvt_col]=min(Zc);
        fprintf('The most negative value in Zj-Cj row is %d and coresponding to column %d \n',Ent_col,pvt_col)
        fprintf('Entering variable is %d \n',pvt_col)
        %For finding the leaving variable
        sol=A(:,end);
        column=A(:,pvt_col);
        if all(column<=0)
            error('The LPP has unbounded solution \n since all enteries are <=0 in %d \n',pvt_col)
        else
            for i=1:size(column,1)
                if column(i)>0
                    ratio(i)=sol(i)./column(i)
                else
                    ratio(i)=inf
                end
            end
            % To finding minimmum ratio
            [minratio pvt_row]=min(ratio);
            fprintf('The minimum ratio corresponding to pivot row %d \n ',pvt_row)
            fprintf('leaving variable is %d \n ',BV(pvt_row))
            BV(pvt_row)=pvt_col;
            disp('New basic variable(BV)==')
            disp(BV)
            pvt_key=A(pvt_row,pvt_col)
            % To update table for next iteration
            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
                ZjCj=ZjCj-ZjCj(pvt_col).*A(pvt_row,:);
            end

             end
    else
        Run= false;
        ZCj=[ZjCj;A]
        FinalTable=array2table(ZCj);
        FinalTable.Properties.VariableNames(1:size(ZCj,2))=variables
        FinalTable.Properties.RowNames(1:size(ZCj,1))={'Zj-Cj','x1','s2','x3'}
        BFS=zeros(1,size(A,2));
        BFS(BV)=A(:,end)
        BFS(end)=0-sum(BFS.*Cost);
        currentBFS=array2table(BFS);
        currentBFS.Properties.VariableNames(1:size(currentBFS,2))={'x1','x2','x3','s1','s2','s3','Opt.Val of Z'}
        disp('Optimal sol is reached')
    end
end
```

```
The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     4     5     6

The most negative value in Zj-Cj row is -3 and coresponding to column 2
Entering variable is 2

ratio =

   Inf


ratio =

   Inf     3


ratio =

       Inf    3.0000    3.3333

The minimum ratio corresponding to pivot row 2
 leaving variable is 5
 New basic variable(BV)==
     4     2     6


pvt_key =

     4


A =

    3.0000   -1.0000    2.0000    1.0000         0         0    7.0000
   -0.5000    1.0000         0         0    0.2500         0    3.0000
   -4.0000    3.0000    8.0000         0         0    1.0000   10.0000

The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     4     2     6

The most negative value in Zj-Cj row is -5.000000e-01 and coresponding to column 1
Entering variable is 1

ratio =

    4.0000    3.0000    3.3333


ratio =

    4.0000       Inf    3.3333


ratio =

     4   Inf   Inf
```

```
The minimum ratio corresponding to pivot row 1
 leaving variable is 4
 New basic variable(BV)==
     1     2     6


pvt_key =

    2.5000


A =

    1.0000         0    0.8000    0.4000    0.1000         0    4.0000
   -0.5000    1.0000         0         0    0.2500         0    3.0000
   -2.5000         0    8.0000         0   -0.7500    1.0000    1.0000


ZCj =

         0         0    2.4000    0.2000    0.8000         0   11.0000
    1.0000         0    0.8000    0.4000    0.1000         0    4.0000
         0    1.0000    0.4000    0.2000    0.3000         0    5.0000
         0         0   10.0000    1.0000   -0.5000    1.0000   11.0000


FinalTable =

  4×7 table

    x1    x2    x3     s1      s2     s3    sol
    __    __    ___    ___    ____    __    ___

    0     0     2.4    0.2     0.8    0     11
    1     0     0.8    0.4     0.1    0      4
    0     1     0.4    0.2     0.3    0      5
    0     0      10      1    -0.5    1     11


FinalTable =

  4×7 table

             x1    x2    x3     s1      s2     s3    sol
             __    __    ___    ___    ____    __    ___

    Zj-Cj    0     0     2.4    0.2     0.8    0     11
    x1       1     0     0.8    0.4     0.1    0      4
    s2       0     1     0.4    0.2     0.3    0      5
    x3       0     0      10      1    -0.5    1     11


BFS =

     4     5     0     0     0    11     0


currentBFS =
```

```
  1×7 table

    x1    x2    x3    s1    s2    s3    Opt.Val of Z
    __    __    __    __    __    __    ____________

    4     5     0     0     0     11        -11

Optimal sol is reached
```

QUESTION-3

Solve LPP using simplex using Simplex Algo Max Z= 5x1+3x2 s.t. 3x1+5x2<=15 5x1+2x2<=10 xi>=0 i=1-2 Phase-T: Input the Parameter

```
clc
clear all
Noofvariables=2;
variables={'x1','x2','s1','s2','sol'};
c=[5 3]; % cost of objective func
Abar=[3 5;5 2];% const coeff
B=[15;10]; %RHS of constraints
s=eye(size(Abar,1));
A=[Abar s B];
Cost=zeros(1,size(A,2));
Cost(1:Noofvariables)=c;
% Contraints BV
BV=Noofvariables+1:1:size(A,2)-1;
% To calculate Zj-Cj
ZjCj=Cost(BV)*A-Cost;
% For printing 1st simplex table
ZCj=[ZjCj;A];
simplextable=array2table(ZCj);
simplextable.Properties.VariableNames(1:size(ZCj,2))=variables;
% Start simplex Algorithm
Run=true;
while Run
    if any(ZjCj<0) % to check if any negative value there
        fprintf('The current BFS is not optimal\n')
        fprintf('Next iteration required \n')
        disp('Old basic variable (BV)=')
        disp(BV)
        % For finding entering variable
        Zc=ZjCj(1:end-1);
        [Ent_col pvt_col]=min(Zc);
        fprintf('The most negative value in Zj-Cj row is %d and coresponding to column %d \n',Ent_col,pvt_col)
        fprintf('Entering variable is %d \n',pvt_col)
        %For finding the leaving variable
        sol=A(:,end);
        column=A(:,pvt_col);
        if all(column<=0)
            error('The LPP has unbounded solution \n since all enteries are <=0 in %d \n',pvt_col)
        else
            for i=1:size(column,1)
                if column(i)>0
                    ratio(i)=sol(i)./column(i)
                else
                    ratio(i)=inf
                end
            end
            % To finding minimum ratio
```

```
            [minratio pvt_row]=min(ratio);
            fprintf('The minimum ratio corresponding to pivot row %d \n ',pvt_row)
            fprintf('leaving variable is %d \n ',BV(pvt_row))
            BV(pvt_row)=pvt_col;
            disp('New basic variable(BV)==')
            disp(BV)
            pvt_key=A(pvt_row,pvt_col)
            % To update table for next iteration
            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
                ZjCj=ZjCj-ZjCj(pvt_col).*A(pvt_row,:);
            end

                 end
    else
        Run= false;
        ZCj=[ZjCj;A]
        FinalTable=array2table(ZCj);
        FinalTable.Properties.VariableNames(1:size(ZCj,2))=variables
        FinalTable.Properties.RowNames(1:size(ZCj,1))={'Zj-Cj','x1','x2'}
        BFS=zeros(1,size(A,2));
        BFS(BV)=A(:,end)
        BFS(end)=sum(BFS.*Cost);
        currentBFS=array2table(BFS);
        currentBFS.Properties.VariableNames(1:size(currentBFS,2))={'x1','x2','s1','s2','Opt.Val of Z'}
        disp('Optimal sol is reached')
    end
end
```

```
The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     3     4

The most negative value in Zj-Cj row is -5 and coresponding to column 1
Entering variable is 1

ratio =

     5


ratio =

     5     2

The minimum ratio corresponding to pivot row 2
 leaving variable is 4
 New basic variable(BV)==
     3     1


pvt_key =

     5
```

```
A =

    3.0000    5.0000    1.0000         0   15.0000
    1.0000    0.4000         0    0.2000    2.0000

The current BFS is not optimal
Next iteration required
Old basic variable (BV)=
     3     1

The most negative value in Zj-Cj row is -1 and coresponding to column 2
Entering variable is 2

ratio =

    2.3684    2.0000


ratio =

    2.3684    5.0000

The minimum ratio corresponding to pivot row 1
 leaving variable is 3
 New basic variable(BV)==
     2     1


pvt_key =

    3.8000


A =

         0    1.0000    0.2632   -0.1579    2.3684
    1.0000    0.4000         0    0.2000    2.0000


ZCj =

         0         0    0.2632    0.8421   12.3684
         0    1.0000    0.2632   -0.1579    2.3684
    1.0000         0   -0.1053    0.2632    1.0526


FinalTable =

  3×5 table

    x1    x2       s1          s2        sol  
    __    __    ________    ________    ______

    0     0      0.26316     0.84211    12.368
    0     1      0.26316    -0.15789    2.3684
    1     0     -0.10526     0.26316    1.0526


FinalTable =
```

```
  3×5 table

             x1    x2      s1          s2        sol
             __    __    ________    ________    ______

    Zj-Cj    0     0      0.26316     0.84211    12.368
    x1       0     1      0.26316    -0.15789    2.3684
    x2       1     0     -0.10526     0.26316    1.0526


BFS =

    1.0526    2.3684         0         0         0


currentBFS =

  1×5 table

      x1        x2      s1    s2    Opt.Val of Z
    ______    ______    __    __    ____________

    1.0526    2.3684    0     0        12.368

Optimal sol is reached
```

Published with MATLAB® R2023b