

LEAST COST METHOD - SHREEYA CHATTERJI(102103447)

Contents

- [QUESTION 1](#)
- [QUESTION 2](#)
- [QUESTION 3](#)

QUESTION 1

```
clc
clear all
format short
%Input the paramters
Cost = [5 2 4 3; 6 4 9 5; 2 3 8 1];
S = [30 40 55]; %supply is atmost
D= [15 20 40 50]; %demand is atleast

% To check if UBTP and fix it if UBTP
if(sum(S)==sum(D))
    fprintf("This is a Balanced Transportation Problem\n")
else
    fprintf("This is an Unbalanced Transportation Problem\n")
    if(sum(S)<sum(D))
        % Add a dummy column with size matching the number of origins (in Demand)
        Cost(:,end+1) = zeros(1,size(D,2));
        % Adding an element in S
        S(end+1)=sum(D)-sum(S);
    else
        % Add a dummy row with size matching the number of destinations (in Supply)
        Cost(:,end+1) = zeros(1,size(S,2));
        %Adding an element in D
        D(end+1)=sum(S)-sum(D);
    end
end

% START LCEM to find MINIMUM COST and MAXIMUM ALLOCATION
ICost=Cost;%make a copy of Cost to make edits during the process
X= zeros(size(Cost));%initial allocation to all places = 0
[m,n]=size(Cost);% to find the number of rows and columns to calcumate m+n-1
nBFS=m+n-1; %total number of BFS cells
% Now find cells with min cost and max allocation
for i=1:size(Cost,1)
    for j=1:size(Cost,2)
        hh=min(Cost(:)); %to find the min cost value in the entire matrix
        [row_ind,col_ind]=find(hh==Cost); %to find the index where hh is equal to the value in cost(pos of minimum cell)
        x11=min(S(row_ind),D(col_ind));%finding the minimum values for allocation for the selected col/row index
        [val,ind]=max(x11);%Choose the max among the min values for maximum allocation and also find it's index
        ii = row_ind(ind); %to identify the row position in the Cost matrix
        jj = col_ind(ind); %to identify the column poisiton in the Cost matrix
        y11=min(S(ii),D(jj)); %to find the value to be inserted into X
        X(ii,jj)=y11; %allocating the value at X at (ii,jj)
        Cost(ii,jj)=Inf; %Allocate a maximum value at the cell where we just found the BV so that it is not considered in further iterations
        %updating the S and D values
        S(ii)=S(ii)-y11;
        D(jj)=D(jj)-y11;
    end
end

% To print the initial BFS
fprintf("Initial BFS:\n")
IBFS=array2table(X);
IBFS.Properties.VariableNames(1:size(IBFS,2))={'D1', 'D2', 'D3', 'D4'};
disp(IBFS)

%%TO CHECK DEGENERATE
FinalBFS=length(nonzeros(X));
if(FinalBFS==nBFS)
    fprintf('The initial BFS is NOT DEGENERATE\n');
else
    fprintf('The initial BFS is DEGENERATE\n');
end
```

```

%%TO COMPUTE IBFS TP Cost
TPCost=sum(sum(ICost.*X));
fprintf('The initial Transportation Cost is = %d\n', TPCost)

```

This is a Balanced Transportation Problem

Initial BFS:

D1	D2	D3	D4
—	—	—	—
0	20	10	0
10	0	30	0
5	0	0	50

The initial BFS is NOT DEGENERATE

The initial Transportation Cost is = 470

QUESTION 2

```

clc
clear all
format short
% Input the paramters
Cost = [2 7 4;3 3 1;5 5 4;1 6 2];
S = [5 8 7 14]; %supply is atmost
D= [7 9 18]; %demand is atleast

% To check if UBTP and fix it if UBTP
if(sum(S)==sum(D))
    fprintf("This is a Balanced Transportation Problem\n")
else
    fprintf("This is an Unbalanced Transportation Problem\n")
    if(sum(S)<sum(D))
        % Add a dummy column with size matching the number of origins (in Demand)
        Cost(:,end+1) = zeros(1,size(D,2));
        % Adding an element in S
        S(end+1)=sum(D)-sum(S);
    else
        % Add a dummy row with size matching the number of destinations (in Supply)
        Cost(:,end+1) = zeros(1,size(S,2));
        %Adding an element in D
        D(end+1)=sum(S)-sum(D);
    end
end

% START LCEM to find MINIMUM COST and MAXIMUM ALLOCATION
ICost=Cost;%make a copy of Cost to make edits during the process
X= zeros(size(Cost));%initial allocation to all places = 0
[m,n]=size(Cost);% to find the number of rows and columns to calcumate m+n-1
nBFS=m+n-1; %total number of BFS cells
% Now find cells with min cost and max allocation
for i=1:size(Cost,1)
    for j=1:size(Cost,2)
        hh=min(Cost(:)); %to find the min cost value in the entire matrix
        [row_ind,col_ind]=find(hh==Cost); %to find the index where hh is equal to the value in cost(pos of minimum cell)
        x11=min(S(row_ind),D(col_ind));%finding the minimum values for allocation for the selected col/row index
        [val,ind]=max(x11);%Choose the max among the min values for maximum allocation and also find it's index
        ii = row_ind(ind); %to identify the row position in the Cost matrix
        jj = col_ind(ind); %to identify the column poisiton in the Cost matrix
        y11=min(S(ii),D(jj)); %to find the value to be inserted into X
        X(ii,jj)=y11; %allocating the value at X at (ii,jj)
        Cost(ii,jj)=Inf; %Allocate a maximum value at the cell where we just found the BV so that it is not considered in further iterations
        %updating the S and D values
        S(ii)=S(ii)-y11;
        D(jj)=D(jj)-y11;
    end
end

% To print the initial BFS
fprintf("Initial BFS:\n")
IBFS=array2table(X);
IBFS.Properties.VariableNames(1:size(IBFS,2))={'W1','W2','W3'};

```

```

disp(IBFS)

%%TO CHECK DEGENERATE
FinalBFS=length(nonzeros(X));
if(FinalBFS==nBFS)
    fprintf('The initial BFS is NOT DEGENERATE\n');
else
    fprintf('The initial BFS is DEGENERATE\n');
end

%%TO COMPUTE IBFS TP Cost
TPCost=sum(sum(ICost.*X));
fprintf('The initial Transportation Cost is = %d\n', TPCost)

```

This is a Balanced Transportation Problem

Initial BFS:

W1	W2	W3
—	—	—
0	2	3
0	0	8
0	7	0
7	0	7

The initial BFS is NOT DEGENERATE

The initial Transportation Cost is = 90

QUESTION 3

```

clc
clear all
format short
% Input the paramters
Cost = [11 20 7 8;21 16 10 12;8 12 18 19];%Cost
S = [50 40 70]; %supply
D= [30 25 35 40]; %demand

% To check if UBTP and fix it if UBTP
if(sum(S)==sum(D))
    fprintf("This is a Balanced Transportation Problem\n")
else
    fprintf("This is an Unbalanced Transportation Problem\n")
    if(sum(S)<sum(D))
        % Add a dummy column with size matching the number of origins (in Demand)
        Cost(:,end+1) = zeros(1,size(D,2));
        % Adding an element in S
        S(end+1)=sum(D)-sum(S);
    else
        % Add a dummy row with size matching the number of destinations (in Supply)
        Cost(:,end+1) = zeros(1,size(S,2));
        %Adding an element in D
        D(end+1)=sum(S)-sum(D);
    end
end

% START LCEM to find MINIMUM COST and MAXIMUM ALLOCATION
ICost=Cost;%make a copy of Cost to make edits during the process
X= zeros(size(Cost));%initial allocation to all places = 0
[m,n]=size(Cost);% to find the number of rows and columns to calcumate m+n-1
nBFS=m+n-1; %total number of BFS cells
% Now find cells with min cost and max allocation
for i=1:size(Cost,1)
    for j=1:size(Cost,2)
        hh=min(Cost(:)); %to find the min cost value in the entire matrix
        [row_ind,col_ind]=find(hh==Cost); %to find the index where hh is equal to the value in cost(pos of minimum cell)
        x11=min(S(row_ind),D(col_ind));%finding the minimum values for allocation for the selected col/row index
        [val,ind]=max(x11);%Choose the max among the min values for maximum allocation and also find it's index
        ii = row_ind(ind); %to identify the row position in the Cost matrix
        jj = col_ind(ind); %to identify the column poisiton in the Cost matrix
        y11=min(S(ii),D(jj)); %to find the value to be inserted into X
        X(ii,jj)=y11; %allocating the value at X at (ii,jj)
        Cost(ii,jj)=Inf; %Allocate a maximum value at the cell where we just found the BV so that it is not considered in further iterations
    end
end

```

```

        %updating the S and D values
        S(ii)=S(ii)-y11;
        D(jj)=D(jj)-y11;
    end
end

% To print the initial BFS
fprintf("Initial BFS:\n")
IBFS=array2table(X);
IBFS.Properties.VariableNames(1:size(IBFS,2))={'W1','W2','W3','W4','W5'};
disp(IBFS)

%%TO CHECK DEGENERATE
FinalBFS=length(nonzeros(X));
if(FinalBFS==nBFS)
    fprintf('The initial BFS is NOT DEGENERATE\n');
else
    fprintf('The initial BFS is DEGENERATE\n');
end

%%TO COMPUTE IBFS TP Cost
TPCost=sum(sum(ICost.*X));
fprintf('The initial Transportation Cost is = %d\n', TPCost)

```

This is an Unbalanced Transportation Problem

Initial BFS:

W1	W2	W3	W4	W5
—	—	—	—	—
0	0	20	0	30
0	0	15	25	0
30	25	0	15	0

The initial BFS is NOT DEGENERATE

The initial Transportation Cost is = 1415