**Operation Research Mini Project**

**Aim:**

Solve the following given Transportation problem.

A supplier supplies catch fishes each of one kg from different sources to different destinations. A survey is conducted in Nagapattinam and Thiruvarur district, Tamilnadu, India and the data has been collected by conducting questionnaire from whole sellers. There are different 3 suppliers (S1,S2, S3) and 3 destinations (D1, D2, D3).Transportation cost (by road) for each fish per kilometre is ₹.0.0015 Find the minimum distance and transportation cost for total fishes.

|  |
| --- |
|  |
| Origin | D1 | D2 | D3 | Supply  (in lakhs) |
| S1 | 53 | 76 | 75 | 7 |
| S2 | 42 | 62 | 65 | 5 |
| S3 | 85 | 100 | 78 | 4 |
| Demand (in lakhs) | 3 | 4 | 6 |  |  |

Data for distance

# **Theory:**

**Vogel’s Approximation Method (VAM)** is one of the methods used to calculate the initial basic feasible solution to a transportation problem. However, it is an iterative procedure such that in each step, we should find the penalties for each available row and column by taking the least cost and second least cost.

Steps:

1. Identify the two lowest costs in each row and column of the given cost matrix and then write the absolute row and column difference. These differences are called penalties.
2. Identify the row or column with the maximum penalty and assign the corresponding cell’s min(supply, demand). If two or more columns or rows have the same maximum penalty, then we can choose one among them as per our convenience.
3. If the assignment in the previous satisfies the supply at the origin, delete the corresponding row. If it satisfies the demand at that destination, delete the corresponding column.
4. Stop the procedure if supply at each origin is 0, i.e., every supply is exhausted, and demand at each destination is 0, i.e., every demand is satisfying. If not, repeat the above steps.

**MODIFIED DISTRIBUTION METHOD**

**(MODI Method):**

The modified distribution method also known as MODI or u-v method provides minimum cost solution to the transportation problem.

In the stepping stone method we have to draw as many closed paths as equal to the unoccupied cells for their evaluation.

To the contrary, in MODI method, only closed path with unoccupied cell with highest opportunity cost is drawn.

Steps:

1. Determine an initial basic feasible solution using any of the three methods given below

NWCR

LCM

VAM

Solution should have m+n-1 allocations in independent positions.

1. Determine the values of the dual variables ui and vj using ui+vj=cij for occupied cells.
2. Compute opportunity cost using dij= Cij – (ui + vj) for unoccupied cells.
3. Check the sign of each opportunity cost.
   1. If opportunity cost of all unoccupied cells are either positive or zero, then given solution is optimal solution.
   2. If one or more unoccupied cells have negative opportunity cost, the given solution is not optimum solution and further saving in transportation cost are possible.
4. Select the unoccupied cell with the smallest negative opportunity cost as the cell to be included in next solution.
5. Draw a closed path or loop for the unoccupied cell selected in previous step.Please note that the right angle turn in this path is permitted only at occupied cells and at the original unoccupied cell.
6. Assign alternate plus and minus sign at the unoccupied cells on the corner points of the closed path with a plus sign at the cell being evaluated.
7. Determine maximum number of units that should be shipped to this unoccupied cell.The smallest value with negative position on closed path indicates the number of units that can be shipped to the entering cell.Now add this quantity to all the cells on the corner points of the closed path marked with plus sign and subtract it from those cells marked with minus sign.In this way , an unoccupied cell becomes occupied cell
8. Repeat the whole procedure until an optimum solution is obtained.

**Solution:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Origin | D1 | D2 | D3 | Supply  (in lakhs) |
| S1 | 53 | 76 | 75 | 7 |
| S2 | 42 | 62 | 65 | 5 |
| S3 | 85 | 100 | 78 | 4 |
| Demand (in lakhs) | 3 | 4 | 6 |  |  |

The proposed transportation model is not a balanced one. In order to make the problem balanced, we introduce a dummy variable say D4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Origin | D1 | D2 | D3 | D4 | Supply  (in lakhs) |
| S1 | 53 | 76 | 75 | 0 | 7 |
| S2 | 42 | 62 | 65 | 0 | 5 |
| S3 | 85 | 100 | 78 | 0 | 4 |
| Demand (in lakhs) | 3 | 4 | 6 | 3 | 16 |  |

Distance Matrix Balanced

VAM method:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | Row Penalty |
| *S*1 | 53 | 76 | 75 | 0 | 7 | 53=53-0 |
| *S*2 | 42 | 62 | 65 | 0 | 5 | 42=42-0 |
| *S*3 | 85 | 100 | 78 | 0 | 4 | 78=78-0 |
| Demand | 3 | 4 | 6 | 3 |  |
| Column Penalty | 11=53-42 | 14=76-62 | 10=75-65 | 0=0-0 |

The maximum penalty, 78, occurs in row *S*3.  
The minimum *cij* in this row is *c*34=0.  
The maximum allocation in this cell is min(4,3) = **3**.  
It satisfy demand of *D*4 and adjust the supply of *S*3 from 4 to 1 (4 - 3=1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | Row Penalty |
| *S*1 | 53 | 76 | 75 | 0 | 7 | 22=75-53 |
| *S*2 | 42 | 62 | 65 | 0 | 5 | 20=62-42 |
| *S*3 | 85 | 100 | 78 | 0**(3)** | 1 | 7=85-78 |
| Demand | 3 | 4 | 6 | 0 |  |  |
| Column Penalty | 11=53-42 | 14=76-62 | 10=75-65 | -- |  |  |

The maximum penalty, 22, occurs in row *S*1.  
The minimum *cij* in this row is *c*11=53.  
The maximum allocation in this cell is min(7,3) = **3**.  
It satisfy demand of *D*1 and adjust the supply of *S*1 from 7 to 4 (7 - 3=4).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | Row Penalty |
| *S*1 | 53**(3)** | 76 | 75 | 0 | 4 | 1=76-75 |
| *S*2 | 42 | 62 | 65 | 0 | 5 | 3=65-62 |
| *S*3 | 85 | 100 | 78 | 0**(3)** | 1 | 22=100-78 |
| Demand | 0 | 4 | 6 | 0 |  |  |
| Column Penalty | -- | 14=76-62 | 10=75-65 | -- |  |  |

The maximum penalty, 22, occurs in row *S*3.  
The minimum *cij* in this row is *c*33=78.  
The maximum allocation in this cell is min(1,6) = **1**.  
It satisfy supply of *S*3 and adjust the demand of *D*3 from 6 to 5 (6 - 1=5).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | | Row Penalty | |
| *S*1 | 53**(3)** | 76 | 75 | 0 | 4 | | 1=76-75 | |
| *S*2 | 42 | 62 | 65 | 0 | 5 | | 3=65-62 | |
| *S*3 | 85 | 100 | 78**(1)** | 0**(3)** | 0 | | -- | |
| Demand | 0 | 4 | 5 | 0 |  | |  | |
| Column Penalty | -- | 14=76-62 | 10=75-65 | -- |  |  | |  | |

The maximum penalty, 14, occurs in column *D*2.  
The minimum *cij* in this column is *c*22=62.  
The maximum allocation in this cell is min(5,4) = **4**.  
It satisfy demand of *D*2 and adjust the supply of *S*2 from 5 to 1 (5 - 4=1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | Row Penalty |
| *S*1 | 53**(3)** | 76 | 75 | 0 | 4 | 75 |
| *S*2 | 42 | 62**(4)** | 65 | 0 | 1 | 65 |
| *S*3 | 85 | 100 | 78**(1)** | 0**(3)** | 0 | -- |
| Demand | 0 | 0 | 5 | 0 |  |  |
| Column Penalty | -- | -- | 10=75-65 | -- |  |  |

The maximum penalty, 75, occurs in row *S*1.  
The minimum *cij* in this row is *c*13=75.  
The maximum allocation in this cell is min(4,5) = **4**.  
It satisfy supply of *S*1 and adjust the demand of *D*3 from 5 to 1 (5 - 4=1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | Row Penalty |
| *S*1 | 53**(3)** | 76 | 75**(4)** | 0 | 0 | -- |
| *S*2 | 42 | 62**(4)** | 65 | 0 | 1 | 65 |
| *S*3 | 85 | 100 | 78**(1)** | 0**(3)** | 0 | -- |
| Demand | 0 | 0 | 1 | 0 |  |  |
| Column Penalty | -- | -- | 65 | -- |  |  |

The maximum penalty, 65, occurs in row *S*2.  
The minimum *cij* in this row is *c*23=65.  
The maximum allocation in this cell is min(1,1) = **1**.  
It satisfy supply of *S*2 and demand of *D*3.  
  
  
Finally we get,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply |
| *S*1 | 53**(3)** | 76 | 75**(4)** | 0 | 7 |
| *S*2 | 42 | 62**(4)** | 65**(1)** | 0 | 5 |
| *S*3 | 85 | 100 | 78**(1)** | 0**(3)** | 4 |
| Demand | 3 | 4 | 6 | 3 |  |

The minimum total transportation cost =53×3+75×4+62×4+65×1+78×1+0×3=850 km

Optimality test using MODI method:

Allocation Table is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply |
| *S*1 | 53 **(3)** | 76 | 75 **(4)** | 0 | 7 |
| *S*2 | 42 | 62 **(4)** | 65 **(1)** | 0 | 5 |
| *S*3 | 85 | 100 | 78 **(1)** | 0 **(3)** | 4 |
| Demand | 3 | 4 | 6 | 3 |  |

Iteration-1 of optimality test  
**1.** Find *ui* and *vj* for all occupied cells(i,j), where *cij*=*ui*+*vj*  
  
1. Substituting, *v*3=0, we get  
  
2.*c*13=*u*1+*v*3⇒*u*1=*c*13-*v*3⇒*u*1=75-0⇒*u*1=75  
  
3.*c*11=*u*1+*v*1⇒*v*1=*c*11-*u*1⇒*v*1=53-75⇒*v*1=-22  
  
4.*c*23=*u*2+*v*3⇒*u*2=*c*23-*v*3⇒*u*2=65-0⇒*u*2=65  
  
5.*c*22=*u*2+*v*2⇒*v*2=*c*22-*u*2⇒*v*2=62-65⇒*v*2=-3  
  
6.*c*33=*u*3+*v*3⇒*u*3=*c*33-*v*3⇒*u*3=78-0⇒*u*3=78  
  
7.*c*34=*u*3+*v*4⇒*v*4=*c*34-*u*3⇒*v*4=0-78⇒*v*4=-78

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | *ui* |
| *S*1 | 53 **(3)** | 76 | 75 **(4)** | 0 | 7 | *u*1=75 |
| *S*2 | 42 | 62 **(4)** | 65 **(1)** | 0 | 5 | *u*2=65 |
| *S*3 | 85 | 100 | 78 **(1)** | 0 **(3)** | 4 | *u*3=78 |
| Demand | 3 | 4 | 6 | 3 |  |  |
| *vj* | *v*1=-22 | *v*2=-3 | *v*3=0 | *v*4=-78 |  |  |

**2.** Find *dij* for all unoccupied cells(i,j), where *dij*=*cij*-(*ui*+*vj*)  
  
1.*d*12=*c*12-(*u*1+*v*2)=76-(75-3)=4  
  
2.*d*14=*c*14-(*u*1+*v*4)=0-(75-78)=3  
  
3.*d*21=*c*21-(*u*2+*v*1)=42-(65-22)=-1  
  
4.*d*24=*c*24-(*u*2+*v*4)=0-(65-78)=13  
  
5.*d*31=*c*31-(*u*3+*v*1)=85-(78-22)=29  
  
6.*d*32=*c*32-(*u*3+*v*2)=100-(78-3)=25

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | *ui* |
| *S*1 | 53 **(3)** | 76 [4] | 75 **(4)** | 0 [3] | 7 | *u*1=75 |
| *S*2 | 42 [-1] | 62 **(4)** | 65 **(1)** | 0 [13] | 5 | *u*2=65 |
| *S*3 | 85 [29] | 100 [25] | 78 **(1)** | 0 **(3)** | 4 | *u*3=78 |
| Demand | 3 | 4 | 6 | 3 |  |  |
| *vj* | *v*1=-22 | *v*2=-3 | *v*3=0 | *v*4=-78 |  |  |

**3.** Now choose the minimum negative value from all *dij* (opportunity cost) = *d*21 = [-1]  
  
and draw a closed path from *S*2*D*1.  
  
Closed path is *S*2*D*1→*S*2*D*3→*S*1*D*3→*S*1*D*1  
  
Closed path and plus/minus sign allocation...

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | *ui* |
| *S*1 | 53 **(3)** **(-)** | 76 [4] | 75 **(4)** **(+)** | 0 [3] | 7 | *u*1=75 |
| *S*2 | 42 [-1] **(+)** | 62 **(4)** | 65 **(1)** **(-)** | 0 [13] | 5 | *u*2=65 |
| *S*3 | 85 [29] | 100 [25] | 78 **(1)** | 0 **(3)** | 4 | *u*3=78 |
| Demand | 3 | 4 | 6 | 3 |  |  |
| *vj* | *v*1=-22 | *v*2=-3 | *v*3=0 | *v*4=-78 |  |  |

**4.** Minimum allocated value among all negative position **(-)** on closed path = 1  
Substract 1 from all **(-)** and Add it to all **(+)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply |
| *S*1 | 53 **(2)** | 76 | 75 **(5)** | 0 | 7 |
| *S*2 | 42 **(1)** | 62 **(4)** | 65 | 0 | 5 |
| *S*3 | 85 | 100 | 78 **(1)** | 0 **(3)** | 4 |
| Demand | 3 | 4 | 6 | 3 |  |

**5.** Repeat the step 1 to 4, until an optimal solution is obtained.

Iteration-2 of optimality test  
 **1.** Find *ui* and *vj* for all occupied cells(i,j), where *cij*=*ui*+*vj*  
  
1. Substituting, *u*1=0, we get  
  
2.*c*11=*u*1+*v*1⇒*v*1=*c*11-*u*1⇒*v*1=53-0⇒*v*1=53  
  
3.*c*21=*u*2+*v*1⇒*u*2=*c*21-*v*1⇒*u*2=42-53⇒*u*2=-11  
  
4.*c*22=*u*2+*v*2⇒*v*2=*c*22-*u*2⇒*v*2=62+11⇒*v*2=73  
  
5.*c*13=*u*1+*v*3⇒*v*3=*c*13-*u*1⇒*v*3=75-0⇒*v*3=75  
  
6.*c*33=*u*3+*v*3⇒*u*3=*c*33-*v*3⇒*u*3=78-75⇒*u*3=3  
  
7.*c*34=*u*3+*v*4⇒*v*4=*c*34-*u*3⇒*v*4=0-3⇒*v*4=-3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | *ui* |
| *S*1 | 53 **(2)** | 76 | 75 **(5)** | 0 | 7 | *u*1=0 |
| *S*2 | 42 **(1)** | 62 **(4)** | 65 | 0 | 5 | *u*2=-11 |
| *S*3 | 85 | 100 | 78 **(1)** | 0 **(3)** | 4 | *u*3=3 |
| Demand | 3 | 4 | 6 | 3 |  |  |
| *vj* | *v*1=53 | *v*2=73 | *v*3=75 | *v*4=-3 |  |  |

**2.** Find *dij* for all unoccupied cells(i,j), where *dij*=*cij*-(*ui*+*vj*)  
  
1.*d*12=*c*12-(*u*1+*v*2)=76-(0+73)=3  
  
2.*d*14=*c*14-(*u*1+*v*4)=0-(0-3)=3  
  
3.*d*23=*c*23-(*u*2+*v*3)=65-(-11+75)=1  
  
4.*d*24=*c*24-(*u*2+*v*4)=0-(-11-3)=14  
  
5.*d*31=*c*31-(*u*3+*v*1)=85-(3+53)=29  
  
6.*d*32=*c*32-(*u*3+*v*2)=100-(3+73)=24

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply | *ui* |
| *S*1 | 53 **(2)** | 76 [3] | 75 **(5)** | 0 [3] | 7 | *u*1=0 |
| *S*2 | 42 **(1)** | 62 **(4)** | 65 [1] | 0 [14] | 5 | *u*2=-11 |
| *S*3 | 85 [29] | 100 [24] | 78 **(1)** | 0 **(3)** | 4 | *u*3=3 |
| Demand | 3 | 4 | 6 | 3 |  |  |
| *vj* | *v*1=53 | *v*2=73 | *v*3=75 | *v*4=-3 |  |  |

Since all *dij*≥0.  
  
So final optimal solution is arrived.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *D*1 | *D*2 | *D*3 | *D*4 | Supply |
| *S*1 | 53 **(2)** | 76 | 75 **(5)** | 0 | 7 |
| *S*2 | 42 **(1)** | 62 **(4)** | 65 | 0 | 5 |
| *S*3 | 85 | 100 | 78 **(1)** | 0 **(3)** | 4 |
| Demand | 3 | 4 | 6 | 3 |  |

The minimum total transportation cost =53×2+75×5+42×1+62×4+78×1+0×3=849 km

**Code:**

library(lpSolve)

costs <- matrix(c(53, 76, 75, 0,

42, 62, 65, 0,

85, 100 , 78, 0), nrow = 3, byrow = TRUE)

colnames(costs) <- c("Destination 1", "Destination 2", "Destination 3", "Destination 4")

rownames(costs) <- c("Supplier 1", "Supplier 2", "Supplier 3")

row.signs <- rep("<=", 3)

row.rhs <- c(7, 5, 4)

col.signs <- rep(">=", 4)

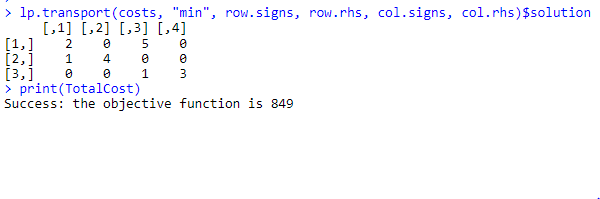
col.rhs <- c(3, 4, 6, 3)

TotalCost <- lp.transport(costs, "min", row.signs, row.rhs, col.signs, col.rhs)

lp.transport(costs, "min", row.signs, row.rhs, col.signs, col.rhs)$solution

print(TotalCost)

**Output:**

****

**Conclusion:**

The minimum Distance is 849 km.

Transportation cost (by road) for each fish per kilometre is ₹.0.0015

Transportation cost per fish is ₹.0.0015 x 849 km = ₹.1.2735

Transportation cost for total quantity = 13,00,000 (minimum requirement) x ₹.1.2735 = ₹.16,55,550

Therefore, Transportation cost for total quantity is

₹.16,55,550