**CASE STUDY ON VOGEL’S APPROXIMATION METHOD (VAM)**

**INTRODUCTION:**

Vogel’s Approximation Method (VAM) is one of the methods used to calculate the initial basic feasible solution to a transportation problem. However, VAM 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.

**PROBLEM:**

A mobile phone manufacturing company has three branches located in three different regions, Jaipur, Udaipur and Mumbai. The company has to transport mobile phones to three destinations, say Kanpur, Pune and Delhi. The availability from Jaipur, Udaipur and Mumbai is 40, 60 and 70 units respectively. The demand at Kanpur, Pune and Delhi are 70, 40 and 60 respectively. Use the Vogel’s Approximation Method to find a basic feasible solution (BFS).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Kanpur** | **Pune** | **Delhi** | **Supply** |
| **Jaipur** | 4 | 5 | 1 | 40 |
| **Udaipur** | 3 | 4 | 3 | 60 |
| **Mumbai** | 6 | 2 | 8 | 70 |
| **Demand** | 70 | 40 | 60 |  |

**STEPS:**

Step 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.

Step 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.

Step 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.

Step 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, i.e., from step 1.

**SOLUTION:**

**Table-1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi |  | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1 |  | 40 | 3=4-1 |
| Udaipur | 3 | 4 | 3 |  | 60 | 0=3-3 |
| Mumbai | 6 | 2 | 8 |  | 70 | 4=6-2 |
|  | | | | | |
| Demand | 70 | 40 | 60 |  |  |  |
| Column Penalty | 1=4-3 | 2=4-2 | 2=3-1 |  |  |  |

The maximum penalty, 4, occurs in row Mumbai.

The minimum cij in this row is c32=2.

The maximum allocation in this cell is min(70,40) = 40.

It satisfy demand of Pune and adjust the supply of Mumbai from 70 to 30 (70 - 40=30).

**Table-2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1 | 40 | 3=4-1 |
| Udaipur | 3 | 4 | 3 | 60 | 0=3-3 |
| Mumbai | 6 | 2**(40)** | 8 | 30 | 2=8-6 |
| Demand | 70 | 0 | 60 |  |  |
| Column Penalty | 1=4-3 | -- | 2=3-1 |  |  |

The maximum penalty, 3, occurs in row Jaipur.

The minimum cij in this row is c13=1.

The maximum allocation in this cell is min(40,60) = 40.

It satisfy supply of Jaipur and adjust the demand of Delhi from 60 to 20 (60 - 40=20).

**Table-3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1**(40)** | 0 | -- |
| Udaipur | 3 | 4 | 3 | 60 | 0=3-3 |
| Mumbai | 6 | 2**(40)** | 8 | 30 | 2=8-6 |
| Demand | 70 | 0 | 20 |  |  |
| Column Penalty | 3=6-3 | -- | 5=8-3 |  |  |

The maximum penalty, 5, occurs in column Delhi.

The minimum cij in this column is c23=3.

The maximum allocation in this cell is min(60,20) = 20.

It satisfy demand of Delhi and adjust the supply of Udaipur from 60 to 40 (60 - 20=40).

**Table-4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1**(40)** | 0 | -- |
| Udaipur | 3 | 4 | 3**(20)** | 40 | 3 |
| Mumbai | 6 | 2**(40)** | 8 | 30 | 6 |
| Demand | 70 | 0 | 0 |  |  |
| Column Penalty | 3=6-3 | -- | -- |  |  |

The maximum penalty, 6, occurs in row Mumbai.

The minimum cij in this row is c31=6.

The maximum allocation in this cell is min(30,70) = 30.

It satisfy supply of Mumbai and adjust the demand of Kanpur from 70 to 40 (70 - 30=40).

**Table-5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1**(40)** | 0 | -- |
| Udaipur | 3 | 4 | 3**(20)** | 40 | 3 |
| Mumbai | 6**(30)** | 2**(40)** | 8 | 0 | -- |
| Demand | 40 | 0 | 0 |  |  |
| Column Penalty | 3 | -- | -- |  |  |

The maximum penalty, 3, occurs in row Udaipur.

The minimum cij in this row is c21=3.

The maximum allocation in this cell is min(40,40) = 40.

It satisfy supply of Udaipur and demand of Kanpur.

**Final Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kanpur | Pune | Delhi | Supply | Row Penalty |
| Jaipur | 4 | 5 | 1**(40)** | 40 | 3 |  3 | -- | -- | -- | |
| Udaipur | 3**(40)** | 4 | 3**(20)** | 60 | 0 |  0 |  0 |  3 |  3 | |
| Mumbai | 6**(30)** | 2**(40)** | 8 | 70 | 4 |  2 |  2 |  6 | -- | |
| Demand | 70 | 40 | 60 |  |  |
| Column Penalty | 1 1 3 3 3 | 2 -- -- -- -- | 2 2 5 -- -- |  |  |

Transportation cost = 1×40 + 3×40 + 3×20 + 6×30 + 2×40 = 480

**CODE:**

library(lpSolve)

costs <- matrix(c(4,5,1,

3,4,3,

6,2,8), nrow = 3, byrow = TRUE)

colnames(costs) <- c("Kanpur","Pune","Delhi")

rownames(costs) <- c("Jaipur","Udaipur","Mumbai")

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

row.rhs <- c(40,60,70)

col.signs <- rep(">=",3)

col.rhs <- c(70,40,60)

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:**

