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 Company has 3 production facilities S1, S2 and S3 with production capacity of 7, 9 and 18 units (in 100's) per week of a product, respectively. These units are tobe shipped to 4 warehouses D1, D2, D3 and D4 with requirement of 5,6,7 and 14 units (in 100's) per week, respectively. The transportation costs (in rupees) per unit between factories to warehouses are given in the table below.

	D1	**D2**	**D3**	**D4**	**Capacity**
S1	19	30	50	10	7
S2	70	30	40	60	9
S3	40	8	70	20	18
Demand	5	8	7	14	34

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

	D1	D2	D3	D4	Capacity	
S1	19	30	50	10	7	9=19-10
S2	70	30	40	60	9	10=40-30
S3	40	8	70	20	18	12=20-8
Demand	5	8	7	14		
	21=40-19	22=30-8	10=50-40	10=20-10		

The maximum penalty, 22, occurs in column D2.

The minimum cij in this column is c32 = 8.

The maximum allocation in this cell is min(18,8) = 8.

It satisfy demand of D2 and adjust the supply of S3 from 18 to 10 (18 - 8 = 10).

Table-2

	D1	D2	D3	D4	Capacity	
S1	19	30	50	10	7	9=19-10
S2	70	30	40	60	9	20=60-30
S3	40	8(8)	70	20	18	20=40-20
Demand	5	0	7	14		
	21=40-19	-	10=50-40	10=20-10		

The maximum penalty, 21, occurs in column D1.

The minimum cij in this column is c11 = 19.

The maximum allocation in this cell is min(7,5) = 5.

It satisfy demand of D1 and adjust the supply of S1 from 7 to 2 (7 - 5 = 2).

Table-3

	D1	D2	D3	D4	Capacity	
S1	**19(5)**	**30**	**50**	**10**	**2**	**40=50-10**
S2	**70**	**30**	**40**	**60**	**9**	**20=60-40**
S3	**40**	**8(8)**	**70**	**20**	**10**	**50=70-20**
Demand	**0**	**0**	**7**	**14**		
	-	-	**10=50-40**	**10=20-10**		

The maximum penalty, 50, occurs in row S3.

The minimum cij in this row is c34 = 20.

The maximum allocation in this cell is min(10,14) = 10.

It satisfy supply of S3 and adjust the demand of D4 from 14 to 4 (14 - 10 = 4).

Table-4

	D1	D2	D3	D4	Capacity	
S1	19(5)	30	50	10	2	40=50-10
S2	70	30	40	60	9	20=60-40
S3	40	8(8)	70	20(10)	0	-
Demand	0	0	7	4		
	-	-	10=50-40	50=60-10		

The maximum penalty, 50, occurs in column *D*4.
The minimum *cij* in this column is *c*14 = 10.
The maximum allocation in this cell is min(2,4) = **2**.
It satisfy supply of *S*1 and adjust the demand of *D*4 from 4 to 2 (4 - 2 = 2).

Table-5

	D1	D2	D3	D4	Capacity	Column Penalty
S1	19(5)	30	50	10(2)	0	-
S2	70	30	40	60	9	20=60-40
S3	40	8(8)	70	20(10)	0	-
Demand	0	0	7	2		
Column Penalty	-	-	40	60		

The maximum penalty, 60, occurs in column D4.

The minimum cij in this column is c24 = 60.

The maximum allocation in this cell is min(9,2) = 2.

It satisfy demand of D4 and adjust the supply of S2 from 9 to 7 (9 - 2 = 7).

Table 6

	D1	D2	D3	D4	Capacity	Column Penalty
S1	19(5)	30	50	10(2)	0	-
S2	70	30	40	60(2)	7	40
S3	40	8(8)	70	20(10)	0	-
Demand	0	0	7	0		
Column Penalty	-	-	40	-		

The maximum penalty, 40, occurs in row S2.

The minimum cij in this row is c23 = 40.

The maximum allocation in this cell is min(7,7) = 7.

It satisfy supply of S2 and demand of D3.

Final Table

	D1	D2	D3	D4	Capacity	Column Penalty
S1	19(5)	30	50	10(2)	7	9 | 9 | 40 | 40 | -- | -- |
S2	70	30	40(7)	60(2)	9	10 | 20 | 20 | 20 | 20 | 40 |
S3	40	8(8)	70	20(10)	18	12 | 20 | 50 | -- | -- | -- |
Demand	5	8	7	14		
Column Penalty	21 21 -- -- -- --	22 -- -- -- -- --	10 10 10 10 40 40	10 10 10 50 60 --		

The minimum total transportation cost
=19×5+10×2+40×7+60×2+8×8+20×10=743

Here, the number of allocated cells = 6 is equal to m + n - 1 = 3 + 4 - 1 = 6

∴ This solution is non-degenerate

CODE:

```
library(lpSolve)
costs <- matrix(c(19,30,50,10,
                  70,30,40,60,
                  40,8,70,20), nrow = 3, byrow = TRUE)
colnames(costs) <- c("D1","D2","D3","D4")
rownames(costs) <- c("S1","S2","S3")
row.signs <- rep("<=",3)
row.rhs <- c(7,9,18)
col.signs <- rep(">=",4)
col.rhs <- c(5,8,7,14)
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:

```
library(lpSolve)
costs <- matrix(c(19,30,50,10,
                  70,30,40,60,
                  40,8,70,20), nrow = 3, byrow = TRUE)
colnames(costs) <- c("D1","D2","D3","D4")
rownames(costs) <- c("S1","S2","S3")
row.signs <- rep("<=",3)
row.rhs <- c(7,9,18)
col.signs <- rep(">=",4)
col.rhs <- c(5,8,7,14)
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)
```

Run (Ctrl-Enter)

Any scripts or data that you put into this service are public.

```
     [,1] [,2] [,3] [,4]
[1,]    5    0    0    2
[2,]    0    2    7    0
[3,]    0    6    0   12
Success: the objective function is 743
```