International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 08 Issue 09 September 2020, Page no. - 2141-2146
Index Copernicus ICV: 57.55
DOI: 10.47191/ijmcr/v8i9.02

Revised Version of Advanced Vogel's Approximation Method (RAVAM) to Find Better IBFS to Transportation Problems

R. Murugesan[1], T. Esakkiammal[2]

[1]Associate Professor, Department of Mathematics, St. John's College, Palayamkottai, Tirunelveli - 627002
[2]Research Scholar, Department of Mathematics, St. John's College, Palayamkottai,
Affiliated to Manonmaniam Sundaranar University, Tirunelveli – 627012

ARTICLE INFO

Published Online:
29 September 2020

Corresponding Author:
R. Murugesan

ABSTRACT

Reinfeld and Vogel (1958) developed a method known as Vogel's Approximation Method (VAM), which has been the efficient solution procedure for more than sixty years, for obtaining an Initial Basic Feasible Solution (IBFS) for the transportation problems (TPs) as it provides a very good IBFS. The main notion of VAM is to determine penalty cost, which is the difference between the smallest cost and next to the smallest cost in each row and column and make maximum possible allocation at the least cost cell of that row or column which have the highest penalty. While determining the penalty cost the difficulty arises when the smallest and next to the smallest cost have the same identical values. Utpal Kant Das et al. (January 2014) resolved this difficulty and developed a new algorithm named Advanced Vogel's Approximation Method (AVAM) to find an IBFS of TPs and showed that the AVAM gives the lower IBFS than that of by the VAM. During our research, we have identified some limitations of the algorithm of AVAM in selecting a row or column when two or more penalty costs have the same highest magnitude and also in selecting a cell for allocation when the smallest cost cell appears in two or more cells in the selected row or column. In this paper, we propose an effective improvement of AVAM in the solution procedure and named it as Revised version of AVAM (RAVAM) to obtain a better IBFS than AVAM for the TPs. To verify the performance of the proposed method, a comparative study is also carried out. Simulation results authenticate that RAVAM yields better IBFS in 90% of the cases than AVAM.

KEYWORDS: Transportation Problem, IBFS, Optimal Solution, VAM, AVAM and RAVAM

I. INTRODUCTION

Transportation problems have been largely studied in the fields of Operations Research and Computer Science. They occupy a very important role in logistics and supply-chain management for reducing the distribution cost and thereby improving the service. In 1941 Hitchcock [1] developed the basic transportation problem along with the constructive method of solution. In 1951, Dantzig [2] formulated the transportation problem as linear programming problem and also provided the solution method. During 1960s, quite a few methods such as North West Corner (NWC) Method, Least Cost Method (LCM) and Vogel's Approximation Method (VAM) [3] have been established for finding the IBFS of TPs. Among them VAM is the efficient method for finding the IBFS of TPs. The obtained IBFS can be tested, whether it is optimal or not, by applying the MODI method [3]. If not optimal, it can be further improved towards optimal solution by applying the MODI method.

In the recent years several methods have been projected by several researchers to find the best IBFS for TPs. Among them, in January 2014, Utpal Kanti Das et al. [4] proposed a new algorithm, named Advanced Vogel's Approximation Method (AVAM) to find IBFS of TPs which is very close to the optimal solution more than VAM. In 2016, Lakhveer Kaur [5] in his paper established with a counter example that the AVAM algorithm which does not always give better IBFS than the VAM. This motivated us to go further in deep the algorithm of AVAM. While solving various problems by the AVAM algorithm we have identified two drawbacks in the algorithm. By rectifying the two drawbacks, in this paper, we have improved the AVAM algorithm and named it as Revised version of AVAM (RAVAM), which produces better IBFS to

TPs than the existing AVAM algorithm. The performance of the RAVAM algorithm has been tested over the identified 10 benchmark problems and the results are compared and discussed.

The paper is organized as follows: In Section 1, brief introduction is given. In Section 2, the existing algorithm of AVAM is presented. Section 3 points out the limitations observed in the AVAM algorithm. The proposed RAVAM algorithm is presented in Section 4. In Section 5, two benchmark problems from balanced type have been illustrated by the AVAM and RAVAM algorithms. Section 6 lists the identified 10 benchmark transportation problems. Section 7 demonstrates the comparison of the results and discussion on the AVAM and RAVAM algorithms. Lastly, in Section 5 conclusions are drawn.

II. EXISTING ALGORITHM OF AVAM

In the algorithm of AVAM, when the smallest cost appear in two or more times in a row or column then penalty is determined by difference of two minimum cost taken one of them as a minimum and following smallest cost other than equal smallest costs as a next to minimum. As an example, if 5, 12, 5, 9, 11 are the costs of a row or column then select 5 as a smallest cost and select 9 as a next to smallest cost instead of 5 again and penalty will be 4. In that case penalty is not zero and if this penalty has the largest magnitude then probability of the chance of taking larger cost in next iteration will be decreased because of at least one more smallest cost remains. The algorithm of AVAM is given follows:

Step-1: Balance the transportation problem, if not balanced.

Step-2:

a) Identify the smallest and next to smallest cost of each row and column and calculate the difference between them which is called by penalty.
b) If smallest cost appear two or more times in a row or column then select one of them as a smallest and following smaller cost other than the equal smallest costs as the next to smallest cost.
c) If there is no more cost other than equal smallest costs. ie all costs are same then select smallest and next to smallest as same and penalty will be zero.

Step-3:

a) Select the lowest cost cell of that row or column which has largest penalty and allocate maximum possible amount in that cell. If the lowest cost appears in two or more cells in that row or column then choose the extreme left or most top lowest cost cell.
b) If two or more penalty costs have same largest magnitude, then select any one of them (or select most top row or extreme left column).

Step-4: Adjust the supply and demand and cross out the satisfied row or column. If row and column are satisfied simultaneously then crossed out one of them and set zero supply or demand in remaining row or column.

Step-5:

a) If exactly one row or one column with zero supply or demand remains uncrossed out, Stop.
b) If only one row or column with positive supply or demand remains uncrossed out, determine the basic variables in the row or column by the Least-Cost Method.
c) If all uncrossed out rows or column have (remaining) zero supply or demand, determined the zero basic variables by the Least-Cost Method. Stop.
d) Otherwise go to Step-2.

III. LIMITATIONS OBSERVED IN THE AVAM ALGORITHM

While applying the algorithm of AVAM in solving TPs we have identified the following two major limitations: The first one is that the Step-3(b) of the algorithm states that “If two or more penalty costs have the same largest magnitude, then select any one of them (or select most top row or extreme left column)”. It is the first limitation which affects the creation of best IBFS by the AVAM method. To avoid this limitation, we have suggested to select the row or column among the highest penalty costs having the smallest unit transportation cost. The second one is that the Step-3(a) of AVAM algorithm states that “Select the lowest cost of that row or column which has largest penalty and allocate maximum possible amount. If the lowest cost appears in two or more cells in that row or column then choose the extreme left or most top lowest cost cell”. It is the second limitation which affects the yielding of best IBFS by the AVAM method. To avoid this limitation, we have suggested for making the allocation to that least cost cell for which the total sum of all the elements in its corresponding row and column is the *maximum*.

IV. ALGORITHM FOR THE PROPOSED IMPROVED AVAM

By overcoming the above two identified limitations, we have improved the existing AVAM algorithm and proposed the Revised version of the AVAM algorithm, known as RAVAM algorithm. The detailed algorithm is described below:

(1) Checking the Balanced Condition. Construct a transportation table, if the given TP is in statement form. Check whether the problem is balanced or not. If the problem is balanced, go to Step 3; otherwise, go to Step 2.

(2) Conversion to Balanced TP. If the problem is not balanced, then anyone of the following two cases may arise:

a) If total supply exceeds total demand, introduce an additional dummy column to the transportation table to absorb the excess supply. The unit transportation cost for the cells in this dummy column is set to 0. Go to Step 3. (OR)

b) If total demand exceeds total supply, introduce an additional dummy row to the transportation table to satisfy the excess demand. The unit transportation cost for the cells in this dummy row is set to 0. Go to Step 3.

(3) Calculating the Penalty Costs.

a) **Calculating the Row Penalty Costs**.

(i) Identify the smallest cost and next to the smallest cost of each row and compute the difference between them, which is called by row penalty cost. Enter the row penalties to the right of the corresponding rows.

(ii) In a row, if the smallest cost appears two or more times, then select one of them as the smallest cost and following smaller cost (other than the equal smallest cost) as the next to the smallest cost.

(iii) If there is no more cost other than the equal smallest costs, then select the smallest and next to the smallest as the same and hence penalty will be zero.

b) **Calculating the Column Penalty Costs**.

(i) Identify the smallest cost and next to the smallest cost of each column and compute the difference between them, which is called by column penalty cost. Enter the column penalties below the corresponding columns.

(ii) In a column, if the smallest cost appears two or more times, then select one of them as the smallest cost and following smaller cost (other than the equal smallest cost) as the next to the smallest cost.

(iii) If there is no more cost other than the equal smallest costs, then select the smallest and next to the smallest as the same and hence penalty will be zero.

(4) Selecting a Cell for Allocation by applying the Tie Breaking Techniques.

(i) Identify the highest penalty cost among the row and column penalty costs. If this is unique, observe the row or column along which this appears.

(ii) Select the cell (i, j) for allocation, which has the least cost in the observed row/column. If such a cell is unique, make allocation $x_{ij} = \text{Min}(s_i, d_j)$ to the cell (i, j), where s_i is the supply at the i^{th} source and d_j is the demand at the j^{th} destination.

(iii) If tie occurs in case of (i), observe the row or column among the highest penalty costs having the smallest unit cost.

(iv) If tie occurs in case of (iii), then make the allocation to that least cost cell for which the total sum of all the elements in its corresponding row and column is the *maximum*.

(v) Again, if tie occurs in case of (iv), then make the allocation to that cell for which *maximum* allocation value can be made.

(vi) Yet again, if tie occurs in case of (v), then make the allocation to that cell for which the sum of demand and supply in the original transportation table is *maximum*.

(vii) Over again, if tie occurs in case of (v) then make the allocation to that cell for which i < k where (i, j) and (k, j) are the competing cells [or to that cell for which j < l where (i, j) and (i, l) are the competing cells].

(viii) All over again, if tie occurs in case of (vii) then select the cell at random for allocation.

(5) Reducing the TT. After performing Step 4, delete the row or column for further calculation where the supply from a given source is exhausted or the demand for a given destination is satisfied. [Except for the last allocation, if we delete both the row and column where the supply from a given source is exhausted as well as the demand for a given destination is satisfied, then this will generate a degenerate solution. To get a non-degenerate solution, delete either the corresponding row only or the column only (but not both), and adjust the supply (demand) as zero, if column (row) is deleted].

(6) Repeat Steps 3 to 5 until and unless all the demands are satisfied and all the supplies are exhausted.

(7) Writing the allocation values. Write the allocations one by one row-wise.

(8) Computing the Total Transportation Cost. Finally, calculate the total transportation cost, which is the sum of the product of unit transportation cost (from the original TP) and the corresponding allocation value.

V. NUMERICAL ILLUSTRATIONS

Suitable illustrative solution makes the readers to understand the proposed algorithm completely. Bearing in mind, two problems have been illustrated. The first one has been taken from the article by the authors of the AVAM algorithm, which does not produce optimal solution directly. The second one has been taken from the comment article due to Lakhveer Kaur on the AVAM algorithm.

Example-1: Consider the illustrative Example Problem-1 taken from the "AVAM" paper due to Utpal Kanti Das et al [5], which is shown in Table 1.

Table 1: The given TP

Sources	D1	D2	D3	D4	Supply
S1	6	8	10	9	**50**
S2	5	8	11	5	**75**
S3	6	9	12	5	**25**
Demand	**20**	**20**	**50**	**60**	

SOLUTION BY THE AVAM ALGORITHM: First, the given TP is solved using the algorithm of AVAM. The IBFS is obtained as shown in Table 2.

Table 2: IBFS generated due to the AVAM algorithm

Sources	D1	D2	D3	D4	Supply
S1	6	**0**	**50**	9	**50**
S2	**15**	8	11	**60**	**75**
S3	**5**	**20**	12	5	**25**
Demand	**20**	**20**	**50**	**60**	

<u>**Computing the Total Transportation Cost:**</u> Z = (0 × 8) + (50 × 10) + (15 × 5) + (60 × 5) + (5 × 6) + (20 × 9) = 0 + 500 + 75 + 300 + 30 + 180 = $1085.

By checking the condition for optimality by the MODI method, it is found that the generated IBFS by the AVAM algorithm is not an optimal one. By applying the MODI method further, this IBFS has been improved towards optimality with the minimum total transportation cost of $1060 in a ***single*** iteration.

SOLUTION BY THE PROPOSED RAVAM ALGORITHM: Next, the given TP is solved using the proposed RAVAM algorithm. The IBFS is obtained as shown in Table 3.

Table 3: IBFS (Optimal Solution) generated due to the RAVAM algorithm

Sources	D1	D2	D3	D4	Supply
S1	6	**0**	**50**	9	**50**
S2	**20**	**20**	11	**35**	**75**
S3	6	9	12	**25**	**25**
Demand	**20**	**20**	**50**	**60**	

<u>**Computing the Total Transportation Cost:**</u> Z = (0 × 8) + (50 × 10) + (20 × 5) + (20 × 8) + (35 × 5) + (25 × 5) = 0 + 500 + 100 + 160 + 175 + 125 = $1060.

By checking the condition for optimality by MODI method, it is found that the generated IBFS by the proposed RAVAM is an optimal one directly.

Example-2: Consider the illustrated Example Problem taken from the "Implication of AVAM" paper due to Lakhveer Kaur [4], which is shown in Table 4.

Table 4: The given TP

Sources	D1	D2	D3	D4	Supply
S1	3	4	3	5	**5**
S2	7	6	4	6	**6**
S3	3	5	7	8	**2**
S4	5	4	9	5	**6**
Demand	**4**	**6**	**2**	**7**	

SOLUTION BY THE AVAM ALGORITHM: First, the given TP is solved using the AVAM algorithm. The IBFS, thus obtained is shown in Table 5.

Table 5: IBFS generated due to the AVAM algorithm

Sources	D1	D2	D3	D4	Supply
S1	**4**	4	3	**1**	**5**
S2	7	6	**2**	**4**	**6**
S3	3	**2**	7	8	**2**
S4	5	**4**	9	**2**	**6**
Demand	**4**	**6**	**2**	**7**	

<u>**Computing the Total Transportation Cost:**</u> Z = (4 × 3) + (1 × 5) + (2 × 4) + (4 × 6) + (2 × 5) + (4 × 4) + (2 × 5) = 12 + 5 + 8 + 24 + 10 + 16 + 10 = $85.

By checking the condition for optimality by the MODI method, it is found that the generated IBFS by AVAM is not an optimal one. By applying the MODI method further, this IBFS has been improved towards optimality with the minimum total transportation cost of $83 in a ***single*** iteration.

SOLUTION BY THE PROPOSED RAVAM ALGORITHM: Next, the given TP is solved using the proposed RAVAM algorithm. The IBFS, thus obtained is shown in Table 6.

Table 6: IBFS (Optimal Solution) generated due to the RAVAM algorithm

Sources	D1	D2	D3	D4	Supply
S1	**2**	**0**	3	**3**	**5**
S2	7	6	**2**	**4**	**6**
S3	**2**	5	7	8	**2**
S4	5	**6**	9	5	**6**
Demand	**4**	**6**	**2**	**7**	

<u>**Computing the Total Transportation Cost:**</u> Z = (2 × 3) + (0 × 4) + (3 × 5) + (2 × 4) + (4 × 6) + (2 × 3) + (6 × 4) = 6 + 0 + 15 + 8 + 24 + 6 + 24 = $83.

By checking the condition for optimality by MODI method, it is found that the generated IBFS by the proposed RAVAM is an optimal one directly.

VI. NUMERICAL EXAMPLES

To justify the efficiency of the proposed algorithm, we have solved ten numbers of classical benchmark TPs in different sizes, from various literature and book, which are listed in Table 7. Because of space limitations, the author(s) name and year of publication only are given.

Table 7: Some classical benchmark TPs

Example No.,(Author(s), Year,)	**Example No.,(Author(s), Year)**
Example 1 (Opera Jude et al., 2017) [C_{ij}] 3×3= [4 3 5; 6 5 4; 8 10 7] [S_i] 3×1= [90, 80, 100] [D_j] 1×3= [70, 120.80]	**Example 6 (Ahmed M.M. et al., 2014)** [C_{ij}] 4×4= [7 5 9 11; 4 3 8 6; 3 8 10 5; 2 6 7 3] [S_i] 4×1= [30, 25, 20, 15] [D_j] 1×4= [30, 30, 20, 10]
Example 2 (Utpal Kanti Das et al., 2014) [C_{ij}] 3×4= [6 8 10 9;5 8 11 5;6 9 12 5] [S_i] 3×1= [50, 75, 25] [D_j] 1×4= [20, 20, 50, 60]	**Problem 7 (Das et al., 2014)** [C_{ij}] 4×5= [10 8 9 5 13; 7 9 8 10 4; 9 3 7 10 6; 11 4 8 3 9] [S_i] 4×1 = [100, 80, 70, 90] [D_j]1×5= [60, 40, 100, 50, 90]
Example 3 (Aminur R. Khan, 2012) [C_{ij}] 3×4= [6 1 9 3;11 5 2 8;10 12 4 7] [S_i] 3×1= [70, 55, 90] [D_j] 1×4= [85, 35, 50, 45]	**Example 8 (Mhlanga A, 2014)** [C_{ij}] 4×5= [4 9 8 10 12;6 10 3 2 3;3 2 7 10 3; 3 5 5 4 8] [S_i] 4×1= [24, 18, 20, 16] [D_j] 1×5= [10, 20, 10, 18, 20]

Example 4 (Srinivasan et al., 1977) $[C_{ij}]$ 3×4 = [3 6 3 4; 6 5 11 15; 1 3 10 5] $[S_i]$ 3×1 = [80, 90, 55] $[D_j]$ 1×4= [70, 60, 35, 60]	**Example 9 (Das et al., 2014)** $[C_{ij}]$ 5×7 = [12 7 3 8 10 6 6;6 9 7 12 8 12 4;10 12 8 4 99 3; 8 5 11 6 7 9 3;7 6 8 11 9 5 6] $[S_i]$ 5×1 = [60, 80, 70, 100, 90] $[D_j]$15×7 = [20, 30, 40, 70, 60, 80, 100]
Example 5 (Lakhveer Kaur, 2016) $[C_{ij}]$ 4×4= [3 4 3 5; 7 6 4 6; 3 5 7 8; 5 4 9 5] $[S_i]$ 4×1= [5, 6, 2, 6] $[D_j]$ 1×4= [4, 6, 2, 7]	**Example 10 (Khan A.R. et al., 2015)** $[C_{ij}]$ 6×6= [12 4 13 18 9 2; 9 16 10 7 15 11; 4 9 10 8 9 7; 9 3 12 6 4 5;7 11 5 18 2 7; 16 8 4 5 1 10] $[S_i]$ 6×1= [120, 80, 50, 90, 100, 60] $[D_j]$ 1×6= [75, 85, 140, 40, 95, 65]

VII. RESULT ANALYSIS

To measure the effectiveness of the proposed RAVAM algorithm, ten benchmark problems have been tested and the results are compared with the results of the existing AVAM algorithm. The comparison of results is shown in Table 8.

Table 8. Comparison of results obtained by the algorithms

Example-No.	Size	VAM	AVAM	RAVAM	Optimal Solution
1.	3×3	1500	1500	1390	1390
2.	3×4	1100	1085	1060	1060
3.	3×4	1220	1220	1165	1160
4.	3×4	955	955	910	880
5.	4×4	83	85	83	83
6.	4×4	470	470	415	410
7.	4×5	2130	2130	2020	2070
8.	4×5	316	318	332	316
9.	5×7	1930	2000	1900	1900
10.	6×6	2310	2310	2220	2170

From Table 8, we discover that out of 10 benchmark problems tested, for 9 problems the proposed RAVAM algorithm has produced better IBFS than that of by the AVAM algorithm and for 4 problems, the proposed RAVAM has produced optimal solutions directly. However, it is noted that, for Example-8 the proposed RAVAM algorithm has failed to produce better IBFS than the AVAM and VAM algorithms.

VIII. CONCLUSION

In this paper we have identified and eliminated the limitations of the existing Advanced VAM (AVAM) algorithm and proposed a new algorithm named revised version of AVAM (RAVAM). To verify the performance of the proposed algorithm, 10 benchmark TPs from the literature have been tested. Simulation results authenticate that the RAVAM algorithm produces better IBFS in 90% of the cases, than the existing AVAM algorithm.

REFERENCES

1. Dantzig G.B. 1963. *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J.
2. Hitchcock F.L. 2006. *The distribution of a product from several sources to numerous localities*, Journal of Mathematical Physics, 20 (2006), 224-230.
3. Kanti Swarup, Gupta P.K., M. Mohan. 2017. *Operations Research*, 19th Edition, Sultan Chand & Sons, Educational Publishers, New Delhi.
4. Lakhveer Kaur. 2016. *Implication of Advanced Vogel's Approximation Method*, International Journal of Science and Research (IJSR), Vol. 5, Issue 8, 1793-1794.
5. Utpal Kanti Das, Md. Ashraful, Aminur Rahman Khan and Md. Sharif Uddin. January 2014. *Advanced Vogel's Approximation Method (AVAM): A New Approach to Determine Penalty Cost for Better Feasible Solution of Transportation Problem*, International Journal of Engineering Research & Technology (IJERT), Vol. 3, Issue 1, 182-187.