# A Cost Supply Method for Finding an Initial Basic Feasible Solution of the Transportation Problem

**Rihan Farih Bunyamin**
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
6025241060@student.its.ac.id

**Bilqis Amaliah**
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
bamaliah@gmail.com (corresponding author)

**Ahmad Saikhu**
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
ahmad89saikhu@gmail.com

*Received: 24 November 2025 | Revised: 18 December 2025 and 7 January 2026 | Accepted: 17 January 2026*

 *DOI: https://doi.org/10.48084/etasr.16468*

## ABSTRACT

**The Transportation Problem (TP) allocates shipments from multiple supply points to demand points at minimum cost. Solving the TP begins with an Initial Basic Feasible Solution (IBFS), which affects the Total Cost (TC). Widely used IBFS heuristics, such as Vogel's Approximation Method (VAM), Juman–Hoque Method (JHM), Total Opportunity Cost Matrix–Minimal Total (TOCM-MT), Bilqis–Chastine–Erma (BCE), and the Supply Selection Method (SSM), cannot always deliver a low-cost starting solution. This study proposes the Cost-Supply Method (CSM). The key innovation of CSM is the formulation of the Cost-Supply (CS) variable. Unlike earlier approaches that treat cost and supply separately, CSM combines them to identify "high-impact" rows. Across 42 balanced test cases, CSM attained the highest accuracy (78.57%), defined as instances in which the IBFS matches the known optimal, and the lowest mean percentage deviation (0.87%) from the optimal cost. Compared with VAM, accuracy improved from 52.38% to 78.57%, and deviation dropped from 4.21% to 0.87%, indicating that CSM yields lower-cost starting solutions more consistently.**

***Keywords-initial basic feasible solution; Vogel approximation method; optimal solution; transportation problem***

## I. INTRODUCTION

TP concerns the cost-minimizing allocation of shipments from multiple sources to multiple destinations under supply and demand constraints [1, 2]. First formally described in [3], TP continues to appear in logistics planning and other operational decisions where transportation costs are a significant component of expenditure [3-5]. Research also utilizes TP-style formulations for issues like scheduling, sustainable routing, congestion pricing, and cash-flow management [6].

Optimal solutions are commonly obtained through improvement procedures such as the Modified Distribution (MODI) and Stepping Stone methods [7, 8]. In practice, the speed of these procedures heavily relies on the initial solution. When IBFS is near optimal, the improvement phase generally needs fewer pivots, leading to shorter runtime and lower overall computation time.

Existing IBFS heuristics can be classified into two main categories: penalty-driven rules and capacity-aware rules. Penalty methods, such as VAM and its variants [9-12], prioritize allocations based on the potential cost penalty. While this strategy avoids costly early choices, it can overlook scale. A row may exhibit a small penalty but carry a large supply, where subsequent feasible allocations may induce significant flow through higher-cost routes, thereby increasing the final TC. Opportunity-cost approaches, such as the Total Opportunity Cost Matrix (TOCM) [13-15, 18], shift the emphasis from absolute to relative costs. However, they do not directly incorporate the magnitude of supply into the decision rule. To address this challenge, researchers have proposed weighted or hybrid heuristics [16]. For instance, the method in [8] creates weighted cost matrix ratios to guide allocations.

Authors in [17] further hybridized this idea by combining KSAM with the Total Differences Method (TDM). These approaches are valuable as they reshape the cost landscape to reveal "relatively attractive" cells. However, their weighting is primarily embedded in matrix transformation, and they do not explicitly represent the exposure of leaving large remaining supplies in systematically expensive rows until late-stage allocation.

Capacity-based heuristics take the opposite approach. Methods, such as SSM [18], prioritize sources with larger supplies so that major constraints are satisfied earlier. Building on JHM [19] and BCE [18], SSM reallocates surplus by focusing on supply quantities. The limitation is that supply volume becomes the dominant factor, meaning large-supply rows are treated similarly, regardless of whether their unit costs are generally low or consistently high. As a result, a method can appear "efficient" in satisfying quantities while still pushing large allocations through costly routes simply because the corresponding supply is large.

This study proposes CSM to address the specific gap. Its key innovation is the CS variable. Unlike BCE, TOCM-MT, and earlier methods that treat cost and supply separately, CSM combines them to identify rows with high cost and large supply. Focusing on these rows helps prevent allocating large amounts to costly cells, a common issue in simpler heuristics. CSM is not guaranteed to be optimal. Reaching optimality requires a follow-up step, such as the stepping-stone procedure. CSM sits between cost-matrix weighting approaches and purely capacity-based rules. Instead of transforming the cost matrix, CSM introduces a new variable (CS = TC × *Si*) and embeds it into SSM-style rebalancing with explicit tie-breaking, targeting cases where large supplies coincide with systematically high row costs.

This research evaluates CSM on 42 test cases, including benchmark instances from literature, synthetic datasets, and real-world data from XYZ, a logistics company located in East Java. The scope of this study is strictly limited to Balanced TPs. The performance of CSM is benchmarked against five established alternatives: VAM, JHM, TOCM-MT, BCE, and SSM, with a focus on both the accuracy of hitting the optimal solution and the minimization of deviation from optimality.

## II. MATHEMATICAL MODEL

The conventional matrix form of the TP is outlined in Figure 1 and Table I.

TABLE I. TP MATRIX

| | **Demand$_1$** | | **Demand$_2$** | | **…** | | **Demand$_n$** | | **Supply** |
|---|---|---|---|---|---|---|---|---|---|
| **Supply$_1$** | $C_{11}$ | $X_{11}$ | $C_{12}$ | $X_{12}$ | … | … | $C_{1n}$ | $X_{1n}$ | $S_1$ |
| **Supply$_2$** | $C_{21}$ | $X_{21}$ | $C_{22}$ | $X_{22}$ | … | … | $C_{2n}$ | $X_{2n}$ | $S_2$ |
| **…** | … | … | … | … | … | … | … | … | … |
| **Supply$_m$** | $C_{m1}$ | $X_{m1}$ | $C_{m2}$ | $X_{m2}$ | … | … | $C_{mn}$ | $X_{mn}$ | $S_m$ |
| **Demand** | $D_1$ | | $D_2$ | | … | | $D_n$ | | |

Rows index the $m$ sources $S_i$ and columns index the $n$ demands $D_j$. Each cell $(i,j)$ records the unit transportation cost $C_{ij}$ and the decision variable $X_{ij}$, the quantity shipped from $S_i$ to $D_j$. The complete mathematical formulation is stated in (1). The aim is to determine $X_{ij}$ and minimize the total TP cost $(Z)$.

$$\left.\begin{array}{r} min\, Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}X_{ij} \\ \sum_{j=1}^{n} X_{ij} = S_i \quad for\, i = 1,2,..,m \\ \text{subject to } \sum_{j=1}^{n} X_{ij} = D_j \quad for\, j = 1,2,..,n \\ X_{ij} > 0 \ \forall i,j \end{array}\right\} \quad (1)$$

where $m$ denotes the total number of sources, $n$ denotes the total number of destinations; $S_i$ is the supply $i$, $D_j$ is the demand $j$; $C_{ij}$ is the transportation cost per unit from $S_i$ to $D_j$; $X_{ij}$ is the number of units shipped from $S_i$ to $D_j$ and $X_{ij}$ cannot be negative.

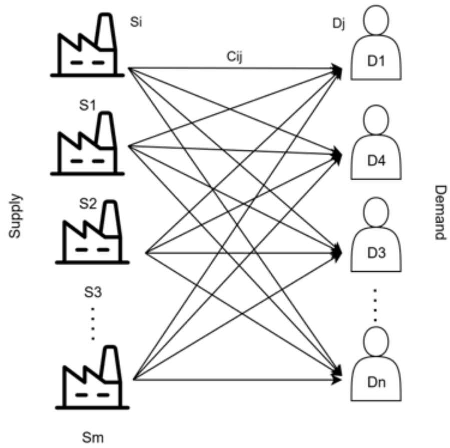


Fig. 1. TP network.

## III. COST SUPPLY METHOD

The CSM method modifies the SSM method to obtain a near-optimal IBFS value. SSM transfers the Excess Row (ER) from the First Least Cost (FLC) row to the Second Least Cost (SLC) row by comparing supply only. The contribution of CSM is to add the TC variable, multiply it by the supply, and then compare the results. The first step is to create an $m$ x $n$ matrix and then add a column to calculate the CS, which is the product of the TC and the row's supply. Priority is given to the row with the largest CS. Figure 2 shows the flowchart of the CSM method. The followed steps are:

1. Step 1: Create an $m$ x $n$ matrix for the TP with cost Cij, supply *Si*, and demand *Dj*. If the problem is unbalanced, change the TP by adding a dummy row (if Demand > Supply) or a dummy column (if Supply > Demand) with zero unit costs. Then proceed to Step 2.
2. Step 2: Create two new columns: the first column for TC and the second for CS. TC is the sum of all costs in each row, while CS is the product of TC and that row's supply. Calculate TC as shown in (2) and CS as shown in (3)/TC and CS as shown in:

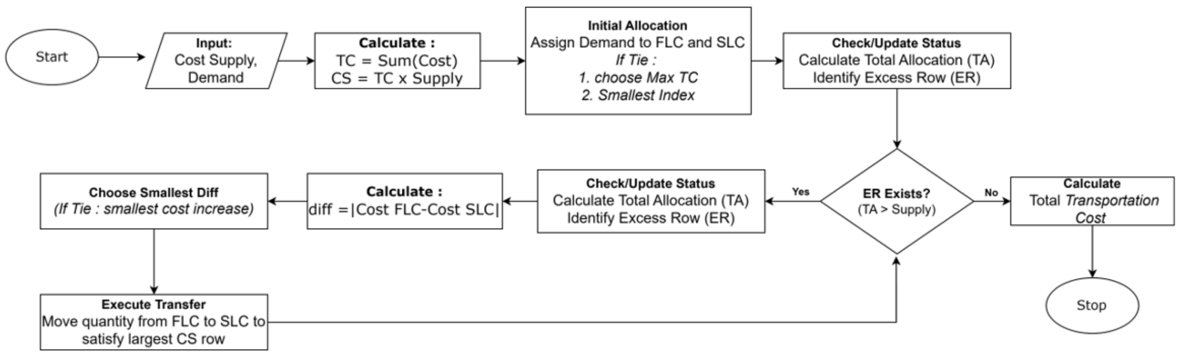


Fig. 2. CSM flowchart.

$$TC_i = \sum_{j=1}^{n} C_{ij} \quad (2)$$

$$Cs = TC_i \times S_i \quad (3)$$

3. Step 3: Find the FLC for each column and the allocation demand in FLC. FLC is the smallest cost in the *Dj* column (column-minima) and can be calculated as:

$$FLC_j = min(C_1, C_2, \ldots C_{nj}) \quad (4)$$

4. Step 4: Calculate the Total Allocation (TA) for each row *i*. Create a new column next to the CS column, labeled 'status'. There are three types of statuses: Satisfy (S), Not Satisfy (NS), and ER. If the TA is equal to the supply, then the status is S; if the TA is smaller than the supply, then the NS state/the status is NS; if the TA is greater than the Supply, then the status of the row is changed to ER. If there is an ER status, proceed to step 5; otherwise, proceed to step 9. TA can be calculated by:

$$TA = \sum_{j=1}^{m} X_{ij} \quad (5)$$

5. Step 5: For each cell in the ER row, obtain the difference (Diff) value by subtracting FLC from SLC. SLC refers to the second lowest cost value in a column after the smallest FLC. SLC cannot be derived from an ER row. If there is more than one SLC, then select the SLC with the largest TC. Then select the Smallest Difference (SD) from the existing diffs. Diff and SD are calculated using:

$$\text{diff} = |\text{FLC} - \text{SLC}| \quad (6)$$

$$\text{SD} = \min(\text{diff}) \quad (7)$$

6. Step 6: Check the number of SD candidates:

- If a unique SD exists, proceed directly to Step 7.
- If multiple candidates share the same SD value (a tie), calculate the potential cost increase for each candidate by multiplying the quantity to be transferred by the unit cost of the destination cell. Then choose the smallest increase and proceed to step 7.
- If it remains the same, choose the option with the smallest unit cost ($C_{ij}$) and proceed to Step 7.

7. Step 7: Check if the FLC supply matches the SLC. If so, satisfy the FLC; if not, satisfy the row with the largest CS.
8. Step 8: Calculate the TA for each row *i*. Check the status of each row. If there is still an ER, go back to step 5. Otherwise, proceed to step 9.
9. Step 9: Calculate the TTC using (8) by multiplying the cost of $C_{ij}$ and the allocation of units $X_{ij}$:

$$TTC = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \quad (8)$$

Derived from SSM, CSM modifies the transfer procedure in the following ways:

- Ties among FLC/SLC candidates: If two (or more) cells tie as FLC/SLC in a column, select the one whose row has the highest $TC_i$ (i.e., higher aggregate unit costs).
- Ties in smallest differences (SDiff): If there are two SDiff, choose the option that minimizes; if still tied, choose the option with the lower unit cost.
- Row satisfaction (FLC versus SLC): If the two rows have equal supplies, ensure that the FLC row is satisfied. Otherwise, satisfy the row with the largest $CS_i$ (larger $TC_i \times s_i$).

Table II presents the modifications and the step-by-step process, while emphasizing the differences between CSM and SSM.

SSM transfers decisions solely on supply comparisons, ignoring the row's cost [18]. With heterogeneous unit costs, this can direct flow toward rows that seem feasible but increase TC. CSM addresses this by introducing a straightforward cost-aware transfer rule that utilizes a CS variable. This makes the rule sensitive to both supply and TC, thereby aligning with TP's primary goal of minimizing cost. Additionally, SSM leaves several tie situations underdefined (e.g., equal Diff, equal supplies, or multiple FLC/SLC candidates), creating ambiguity during implementation. CSM provides explicit tiebreakers for these cases. The detailed pseudocode of the proposed CSM, including the initialization and rebalancing logic, is presented in Figure 3.

TABLE II. COMPARISON BETWEEN SSM AND CSM

| Difference | SSM | CSM |
|---|---|---|
| TP initial matrix | Standard TP matrix (with no additions) | Addition of the new TC and CS columns |
| Searching for FLC and/or SLC (if there are two values of the same value) | None | Choose the FLCs and/or SLCs that have a greater TC value. If still the same, choose the smallest index |
| If there is more than one SD, select the SD. | Select the option that has the greatest allocation | Determine the number of units to be moved, then calculate the cost increase, and select the move with the smallest cost increase |
| ER moving rules | If there is only one ER, then satisfy the row with the smallest supply; otherwise, satisfy the row with the largest supply | Selecting the largest CS (result of the multiplication of the TC by the supply) |

```
Input: Matrix Cost[m][n], Supply[m], Demand[n]
Output: Matrix Allocation[m][n], TotalCost

// Phase 1: Initialization
    FOR each row i in Supply:
        TC[i] = Sum of all costs in row i
        CS[i] = TC[i] * Supply[i]

// Phase 2: Initial Allocation
    FOR each column j in Demand:
        Find row 'best_row' with minimum Cost[i][j]
        IF tie in Cost (FLC/SLC):
            Select row with higher TC[i]
                IF still tie :
                Select row with least index[i]
        Allocate max possible units to Allocation[best_row][j]
        Update remaining Supply and Demand
        Determine Row Status: "Excess Row" (Supply left) or "Need Supply" (Deficit).

// Phase 3: Rebalancing Loop (Steps 5-7)
WHILE (exists any "Excess Row"):
    BestCandidate = NULL
    MinDiff = Infinity

    // A. Search for the Best Move
    FOR each potential transfer from 'FLC' (Excess) to 'Receiver' (Need):
        CurrentDiff = |Cost_FLC - Cost_Receiver|

        // SELECTION CRITERIA
        Priority 1: Smallest Difference (SD)
        IF (CurrentDiff < MinDiff):
            MinDiff = CurrentDiff
            BestCandidate = {FLC, Receiver}
        Priority 2: Smallest Cost Increase (if SD ties)
        Priority 3: Higher CS Value (if Cost Increase ties)

            // CRITERIA 2: Tie-Breaker (Cost-Supply Priority)
        ELSE IF (CurrentDiff == MinDiff):
            // Compare CS values to decide priority
            Priority_Current = MAX(CS[FLC], CS[Receiver])
            Priority_Best    = MAX(CS[BestCandidate.FLC], CS[BestCandidate.Receiver])

            IF (Priority_Current > Priority_Best):
                BestCandidate = {FLC, SLC}
            END IF
        END IF
    END FOR

    // B. Execute Move (Step 7)
    IF (BestCandidate is found):
        Identify which row has higher CS (FLC or SLC).
        Calculate Quantity Q to satisfy that high-priority row.
        Transfer Q units from FLC to SLC.
        Update Row Statuses.
    ELSE:
        BREAK Loop.
END WHILE
Calculate Final Total Cost.
```

Fig. 3. CSM pseudocode.

The following is an example of the application of the CSM algorithm to example N32 from [20].

*1) Steps 1-4: Initialization and Initial Allocation*

Create a TP matrix, as shown in Table III. Calculate the total supply and total demand. Since the total supply and demand of the problem are balanced (each 340), no dummy rows/columns are required. The algorithm calculates the TC and CS for each row.

TABLE III. TP MATRIX FROM (N32)

| | D1 | D2 | D3 | D4 | D5 | S |
|---|---|---|---|---|---|---|
| **S1** | 4 | 4 | 9 | 10 | 13 | 100 |
| **S2** | 7 | 9 | 8 | 10 | 4 | 90 |
| **S3** | 9 | 3 | 7 | 10 | 6 | 80 |
| **S4** | 11 | 4 | 10 | 6 | 9 | 70 |
| **D** | 60 | 40 | 90 | 70 | 80 | |

- Row S1: TC=40, CS=4000
- Row S2: TC=38, CS=3420
- Row S3: TC=35, CS=2800
- Row S4: TC=40, CS=2800

An initial allocation based on the FLC rule yields a preliminary TC of 1820. However, the status check reveals a discrepancy: Row S3 has an ER, while rows S1 and S2 are NS, as portrayed on Table IV.

TABLE IV. STATUS OF EACH ROW AFTER ALLOCATION

| | D1 | D2 | D3 | D4 | D5 | S | TC | CS | Status |
|---|---|---|---|---|---|---|---|---|---|
| **S1** | 4 (60) | 4 | 9 | 10 | 13 | 100 | 40 | 4000 | NS |
| **S2** | 7 | 9 | 8 | 10 | 4 (80) | 90 | 38 | 3420 | NS |
| **S3** | 9 | 3 (40) | 7 (90) | 10 | 6 | 80 | 35 | 2800 | ER |
| **S4** | 11 | 4 | 10 | 6 (70) | 9 | 70 | 40 | 2800 | S |
| **D** | 60 | 40 | 90 | 70 | 80 | | | | |

*2) Steps 5-7: Rebalancing (Transfer)*

The algorithm enters the rebalancing loop to resolve the excess in row S3.

- Iteration 1 (S3-S2): The algorithm checks possible transfers from the ER (S3) to the deficit rows (S1 or S2). This shows a tie in the cost-difference values for columns D2 and D3. To resolve it, a multiplication tie-breaker is used, which selects the option with the smaller cost increase. Moving along D2 would increase the cost by 160 (40 x 4), while moving along D3 would increase it by 80 (10 x 8). Since (80 < 160), D3 is chosen as the SD column. Based on this decision, the algorithm transfers 10 units from S3 to S2. As shown in Table V, this adjustment is necessary because it assigns flow to a lower-cost cell earlier, thereby reducing the likelihood that later allocations are forced into higher-cost cells under the remaining constraints.

TABLE V. CONDITION AFTER MOVING S3D3 TO S2D3

| | D1 | D2 | D3 | D4 | D5 | S | TC | CS | Status |
|---|---|---|---|---|---|---|---|---|---|
| **S1** | 4 (60) | 4 | 9 | 10 | 13 | 100 | 40 | 4000 | NS |
| **S2** | 7 | 9 | 8 (10) | 10 | 4 (80) | 90 | 38 | 3420 | S |
| **S3** | 9 | 3 (40) | 7 (80) | 10 | 6 | 80 | 35 | 2800 | ER |
| **S4** | 11 | 4 | 10 | 6 (70) | 9 | 70 | 40 | 2800 | S |
| **D** | 60 | 40 | 90 | 70 | 80 | | | | |

- Iteration 2 (S2-S1): With S3 still having surplus capacity (ER), the algorithm identifies S1 as the remaining deficit row (NS). Consequently, 40 units are transferred from S3 to S1. As presented in Table VI, this action aligns with the rules, as S1 has a larger CS value (4000).

In this numerical example, the IBFS costs produced by SSM and CSM are 1820 and 1780, respectively. CSM attains the optimal solution.

TABLE VI. CONDITION AFTER MOVING S2D2 TO S1D2

| | D1 | D2 | D3 | D4 | D5 | S | TC | CS | Status |
|---|---|---|---|---|---|---|---|---|---|
| **S1** | 4 (**60**) | 4 (**40**) | 9 | 10 | 13 | 100 | 40 | 4000 | S |
| **S2** | 7 | 9 | 8 (**10**) | 10 | 4 (**80**) | 90 | 38 | 3420 | S |
| **S3** | 9 | 3 | 7 (**80**) | 10 | 6 | 80 | 35 | 2800 | S |
| **S4** | 11 | 4 | 10 | 6 (**70**) | 9 | 70 | 40 | 2800 | S |
| **D** | 60 | 40 | 90 | 70 | 80 | | | | |

*3) Steps 8-9: Result*

After these transfers, all supply and demand constraints are met (Status: Satisfied). The final TTC is calculated as: (4 x 60) + (4 x 40) + (8 x 10) + (7 x 80) + (6 x 70) = 1780.

## IV. EXPERIMENT RESULTS

CSM was evaluated using solution-quality measures commonly reported for IBFS heuristics, particularly accuracy and deviation from optimal cost, to determine the performance of the proposed logic across different instances.

A head-to-head comparison was conducted among VAM, TOCM-MT, JHM, BCE, SSM, and CSM. The evaluation used 42 datasets: 32 benchmark cases from prior studies, 5 synthetic instances designed to test larger problem sizes, and 5 anonymized real-world cases from a logistics company. Table VII summarizes each dataset. For instance, N32 is a 4 supply and 5 demand (4×5 TP matrix), with an optimal cost of 1780. Optimal solutions were computed using TORA and Gurobi. The complete datasets are available in [21].

TABLE VII. SUMMARY OF DATASETS

| Dataset group | Note | Count |
|---|---|---|
| Published study | Benchmark cases from prior studies | 32 |
| Synthetic | Randomly generated instances on larger sizes | 5 |
| Real world case (company XYZ) | Anonymized real-world cases from a logistics company | 5 |

Figure 4 reports the Accuracy Percentage (Ap) across all 42 datasets for: VAM, JHM, TOCM-MT, BCE, SSM, and CSM. The Ap metric, defined by (9), measures how often an IBFS matches the optimal cost. Based on the data in Table VII, the number of optimal IBFS solutions is 22 (VAM), 27 (JHM), 26 (TOCM-MT), 29 (BCE), 31 (SSM), and 33 (CSM), equivalent to 52.38%, 64.29%, 61.90%, 69.05%, 73.81%, and 78.57%. CSM ranks first, achieving optimality in 33 of 42 cases.

$$Ap = \frac{Sum\ IBFS\ Achieve\ Optimal\ Solution}{Total\ Dataset} x100\% \quad (9)$$

Table VIII provides a comparative summary of the Improvement Percentage (Ip) outcomes. The Ip, calculated using (10), can be positive, negative, or zero. Table VIII also presents the average Ip, comparing CSM against VAM, TOCM-MT, JHM, BCE, and SSM. Positive Ip indicates that CSM achieves a lower TC, negative Ip indicates a higher cost, and zero denotes a tie. For example, in N32, CSM shows an improvement over SSM with an Ip of 2.20%. While SSM yields a cost of 1820, CSM reaches the optimal cost of 1780.

$$Ip = \frac{IBFS - CSM}{IBFS} \times 100\% \quad (10)$$

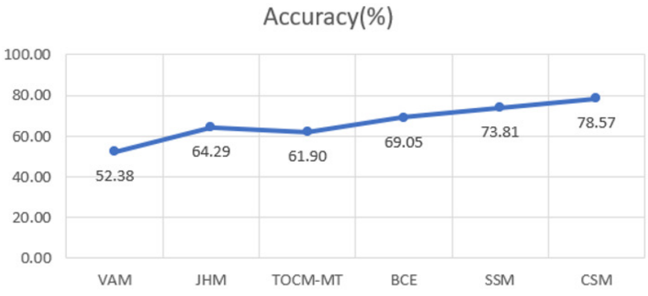


Fig. 4. Ap among the methods.

Although ties are common across the benchmark, Table VIII shows a meaningful number of cases where CSM yields a lower cost. In particular, CSM outperforms VAM in 15 out of 42 datasets (36%).

TABLE VIII. IP BETWEEN CSM AND OTHER METHODS

| NO | Method | Total of | | | Average Ip (%) |
|---|---|---|---|---|---|
| | | Positive | Negative | Zero | |
| 1 | VAM | 15 | 5 | 22 | 3.00 |
| 2 | JHM | 9 | 5 | 29 | 0.84 |
| 3 | TOCM-MT | 13 | 6 | 24 | 0.16 |
| 4 | BCE | 9 | 4 | 30 | 0.38 |
| 5 | SSM | 8 | 6 | 29 | 0.14 |

Figure 5 presents the Deviation Percentage (Dp) for each method relative to the optimal cost, calculated using (11). A Dp value of zero indicates an optimal IBFS, while positive values indicate greater deviation from optimality. For N32, CSM attains a Dp of 0.00, confirming its optimality for that case. The average deviations are 4.21% (VAM), 1.68% (JHM), 0.95% (TOCM-MT), 1.15% (BCE), 1.38% (SSM), and 0.87% (CSM). CSM has the lowest mean deviation, indicating that, on average, it yields solutions closer to the optimal.

$$Dp = \frac{(\text{IBFS} - \text{Optimal Solution})}{\text{Optimal Solution}} \text{x}100\% \quad (11)$$

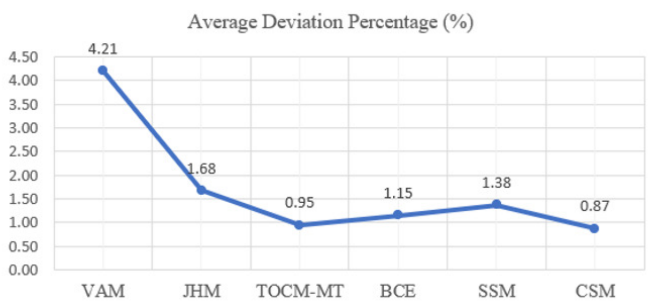


Fig. 5. Average Dp among the methods.

Table IX reports the average runtime of each method together with its stated time complexity. CSM tends to run longer than SSM because it introduces additional decision rules during ER row reallocation, including the computation of TC and CS, and it applies explicit tie-breaking criteria when SSM is ambiguous. As a result, CSM performs more checks per iteration and requires more computational effort than SSM.

TABLE IX. AVERAGE RUNTIME AND TIME COMPLEXITY FOR EACH METHOD

| Methods | Average runtime (ms) | Time complexity |
|---|---|---|
| VAM | 150.965 | O (mn(m+n)) |
| JHM | 4.365 | O $(mn^2)$ |
| TOCM-MT | 12.548 | O (mn) |
| BCE | 3.317 | O $(mn^2)$ |
| SSM | 4.254 | O (mn) |
| CSM | 6.287 | O $(mn^2)$ |

A paired-samples t-test ($N = 42$) was performed to evaluate solution robustness by comparing the Percentage Deviation from Optimality of CSM against each benchmark heuristic (Table X).

TABLE X. P-VALUE FOR EACH METHOD COMPARED TO CSM

| Methods | P-value (one-tailed) |
|---|---|
| VAM | 0.015 |
| JHM | 0.070 |
| TOCM-MT | 0.435 |
| BCE | 0.230 |
| SSM | 0.228 |

CSM achieves a statistically significant improvement over VAM ($p = 0.015 < 0.05$), reducing the mean deviation from 4.21% to 0.87%. For JHM, TOCM-MT, BCE, and SSM, the differences are not statistically significant at the 0.05 level. Nevertheless, CSM shows the lowest average deviation on this benchmark (compared to 1.38% for SSM), suggesting that its decision rules often lead to solutions that are very close to the optimal solution.

CSM emphasizes "high-impact rows" to prevent costly, forced allocations common in penalty-based methods, but it introduces overhead. In Table IX, CSM's average runtime is 6.287 ms, whereas SSM's is 4.254 ms. Its asymptotic cost is O $(mn^2)$ due to repeated TC and CS calculations and tie-breaking during reallocation. For the small-to-medium matrices tested, this overhead is modest, but for very large instances, it may be significant. CSM is suited for scenarios where (i) supply is skewed (one or two sources dominate total supply), (ii) unit costs are heterogeneous, especially when high-supply rows are not the cheapest ones, and (iii) multiple ties occur in rebalancing, where CSM's tie-breaking favors the smallest marginal cost increase.

## V. CONCLUSIONS

This study introduces the Cost–Supply Method (CSM), a supply-weighted SSM for creating an Initial Basic Feasible Solution (IBFS) in Transportation Problems (TPs). Implemented in C, it was tested on 42 instances, with optimal costs verified via TORA and Gurobi. CSM matched the optimal in 33 cases (78.57%), outperforming VAM (52.38%), JHM (64.29%), TOCM-MT (61.90%), BCE (69.05%), and SSM (73.81%). It achieved the smallest mean deviation from the optimum (0.87%), indicating that its solutions are closer to the best.

While CSM produces a feasible solution, its detailed selection and rebalancing steps introduce additional overhead, particularly for larger matrices. Like other deterministic IBFS heuristics, CSM is not guaranteed to be optimal and offers only a limited advantage when costs are nearly uniform or when multiple allocations have similar objective values. CSM often ties with SSM, and the additional computation may not be justified.

With these constraints in mind, CSM is best viewed as a practical option for settings where obtaining a strong starting point is useful, particularly when a better IBFS can reduce steps in logistics planning. Future work will focus on reducing computational overhead (e.g., hybrid tie-breaking or localized search) and extending the approach to broader variants of transportation models, including unbalanced and more complex distribution networks.

## ACKNOWLEDGMENT

The authors acknowledge the financial support from the Institut Teknologi Sepuluh Nopember for this work, under the project scheme of the Publication Writing and IPR Incentive Program (PPHKI) 2025.

## REFERENCES

[1] H. Garg and R. M. Rizk-Allah, "A novel approach for solving rough multi-objective transportation problem: development and prospects," *Computational and Applied Mathematics*, vol. 40, no. 4, May 2021, Art. no. 149, https://doi.org/10.1007/s40314-021-01507-5.

[2] L. Sahoo, "A new score function based Fermatean fuzzy transportation problem," *Results in Control and Optimization*, vol. 4, Sept. 2021, Art. no. 100040, https://doi.org/10.1016/j.rico.2021.100040.

[3] F. L. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," *Journal of Mathematics and Physics*, vol. 20, no. 1–4, pp. 224–230, 1941, https://doi.org/10.1002/sapm1941201224.

[4] C. A. B. N. Bitundu, J. F. Wounba, C. Bwemba, and S. K. Fondzenyuy, "Designing of An Integrated Port Development Planning Model: Application to The Three Main Cameroonian Ports," *Vokasi Unesa Bulletin of Engineering, Technology and Applied Science*, vol. 2, no. 1, pp. 12–35, Mar. 2025, https://doi.org/10.26740/vubeta.v2i1.35876.

[5] M. Sam'an, Y. Dasril, N. B. Bujang, and Farikhin, "Improved total difference method (ITDM): a new approach to solving transportation problem based on modifications of total difference method 1 and integration of total ratio cost matrix," *International Journal of Computing Science and Mathematics*, vol. 16, no. 1, 2022, https://doi.org/10.1504/IJCSM.2022.126767.

[6] J. A. Obari, S. T. A, I. M. A, and A. B. H, "An emission and weight of vehicles-based road traffic congestion pricing system and control with consideration of investment worthiness," *Vokasi Unesa Bulletin of Engineering, Technology and Applied Science*, vol. 2, no. 3, pp. 401–411, Aug. 2025, https://doi.org/10.26740/vubeta.v2i3.38528.

[7] E. Hosseini, "Three new methods to find initial basic feasible solution of transportation problems," *Applied Mathematical Sciences*, vol. 11, no. 37, pp. 1803–1814, 2017, https://doi.org/10.12988/ams.2017.75178.

[8] K. Karagul and Y. Sahin, "A novel approximation method to obtain initial basic feasible solution of transportation problem," *Journal of King Saud University - Engineering Sciences*, vol. 32, no. 3, pp. 211–218, Mar. 2020, https://doi.org/10.1016/j.jksues.2019.03.003.

[9] M. A. Babu, M. A. Helal, M. S. Hasan, and U. K. Das, "Implied cost method (ICM): an alternative approach to find the feasible solution of

transportation problem," *Global Journal of Science Frontier Research-F: Mathematics and Decision Sciences*, vol. 14, no. 1, pp. 5–13, 2014.

[10] H. A. Hussein and M. A. K. Shiker, "A Modification to Vogel's Approximation Method to Solve Transportation Problems," *Journal of Physics: Conference Series*, vol. 1591, no. 1, 2020, Art. no. 012029, https://doi.org/10.1088/1742-6596/1591/1/012029.

[11] M. Mathirajan, S. Reddy, and M. V. Rani, "An experimental study of newly proposed initial basic feasible solution methods for a transportation problem," *OPSEARCH*, vol. 59, no. 1, pp. 102–145, 2022, https://doi.org/10.1007/s12597-021-00533-5.

[12] F. A. Wireko, I. D. K. Mensah, E. N. A. Aborhey, S. A. Appiah, C. Sebil, and J. Ackora-Prah, "The maximum range method for finding initial basic feasible solution for transportation problems," *Results in Control and Optimization*, vol. 19, June 2025, Art. no. 100551, https://doi.org/10.1016/j.rico.2025.100551.

[13] Ö. Kirca and A. Şatir, "A Heuristic for Obtaining and Initial Solution for the Transportation Problem," *Journal of the Operational Research Society*, vol. 41, no. 9, pp. 865–871, 1990, https://doi.org/10.1057/jors.1990.124.

[14] A. A. S. Bilkour, Vishal Vincent Henry, and Satakshi, "Initial Basic Feasible Solution for Transportation Problem using TOCM with Zero Point Minimum Method," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 21s, pp. 3332–3343, 2024.

[15] R. Murugesan and T. Esakkiammal, "TOCM-VAM method versus asm method in transportation problems," *Advances in Mathematics: Scientific Journal*, vol. 9, no. 6, pp. 3549–3566, 2020, https://doi.org/10.37418/amsj.9.6.34.

[16] M. Amreen and V. B, "A New Way For Solving Transportation Issues Based on the Exponential Distribution and the Contraharmonic Mean," *Journal of applied mathematics & informatics*, vol. 42, no. 3, pp. 647–661, 2024, https://doi.org/10.14317/jami.2024.647.

[17] M. Sam'an and Y. N. Ifriza, "A combination of TDM and KSAM to determine initial feasible solution of transportation problems," *Journal of Soft Computing Exploration*, vol. 2, no. 1, pp. 17–24, Mar. 2021, https://doi.org/10.52465/joscex.v2i1.16.

[18] B. Amaliah, C. Fatichah, and E. Suryani, "A Supply Selection Method for better Feasible Solution of balanced transportation problem," *Expert Systems with Applications*, vol. 203, Oct. 2022, Art. no. 117399, https://doi.org/10.1016/j.eswa.2022.117399.

[19] Z. A. M. S. Juman and M. A. Hoque, "An efficient heuristic to obtain a better initial feasible solution to the transportation problem," *Applied Soft Computing*, vol. 34, pp. 813–826, Sept. 2015, https://doi.org/10.1016/j.asoc.2015.05.009.

[20] Md. A. Babu, M. A. Hoque, and Md. S. Uddin, "A heuristic for obtaining better initial feasible solution to the transportation problem," *OPSEARCH*, vol. 57, no. 1, pp. 221–245, 2020, https://doi.org/10.1007/s12597-019-00429-5.

[21] R. F. Bunyamin, "CSM Dataset and Detailed Results," CSM Dataset and Detailed Results, https://rihanfarih.blogspot.com/2025/11/csm-dataset-and-detailed-results.html.