Dhaka Univ. J. Sci. 74(2): 331–340, 2026 (July)
ISSN 1022-2502, E-ISSN 2408-8528
DOI: https://doi.org/10.3329/dujs.v74i2.86871

A Proposed Penalty Based Allocation Strategy to Optimize Transportation Problems

Isnat Jahan Owishi, Md. Asadujjaman* and Esrat Jahan Meem

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

(Received: 4 January 2026; Accepted: 8 April 2026)

Abstract

We proposed a penalty-based allocation strategy to solve Transportation Problems. This method is also implemented to address an unbalanced transportation problem. The effectiveness of the proposed method is assessed by comparing it with other existing methods such as Least Cost Method, North West Corner Method, Vogel's Approximation Method and MODI's Optimality Method. We presented the detailed manual calculations and computational approaches for the proposed algorithm to address the problem. We utilized *MATLAB R2016a* to evaluate the solutions of both the proposed and existing methods. The constructed code is generated to effectively solve problems of any size particularly tackling cost matrices and supply-demand vectors of dimensions $m \times n$.

Keywords: Pentagonal Fuzzy Number, Centroid Ranking Technique, Initial Basic Feasible Solution, Optimal Solution, Transportation Costs.

I. Introduction

Linear Programming is a mathematical method used to find out the best possible outcome such as maximizing profit or minimizing cost of a linear objective function, subject to a set of linear equality or inequality constraints. A specific problem which applies the principles of Linear Programming is known as Linear Programming Problem which involves a linear objective function and linear constraints. The Transportation Problems is a special kind of LP problem where the purpose is to figure out the most affordable way to distribute the goods from various sources (suppliers) to various destinations (consumers) so that the constraints of demand and supply are satisfied. Transportation problems have extensive applications in different sectors where the target is to optimize the movement of products, resources or services while reducing costs. In manufacturing, they help in delivering raw materials to factories or finished products to markets effectively. In energy distribution, they confirm the optimal allocation of resources like electricity or fuel to various destinations. Models for route planning and fleet management are essential for airline and transportation scheduling. Likewise, healthcare and emergency services utilize transportation optimization to allocate medical supplies and manage patient or ambulance routing. Applications also spread to agriculture for delivering goods and resources and waste management for organizing efficient collection and disposal routes. In general, transportation problems provide a structured procedure to achieve economical and effective distribution in real-life situations[1,3].

In this paper, we proposed a penalty based algorithm and applied it to a range of real-life transportation problems to demonstrate its effectiveness, efficiency and practical applicability. The algorithm was evaluated in terms of solution quality, computational performance and its ability to minimize transportation costs across diverse problem scenarios.

The paper is organized in the following sequence. In section II, weexplain the concepts of Transportation Problems. In section III, we introduce a new penalty-based algorithm which is designed to efficiently solve large-scale transportation problems by integrating the advantages of both global and local search strategies to optimize transportation costs[13]. We will present a comparative analysis of the proposed penalty-based algorithm against existing solution methods[8]. We develop a computer-oriented solution by using *MATLAB R2016 a* solve the previous numerical examples by using it. The main point will be on evaluating its efficiency, solution quality and scalability across various types of problems. This comparison will help to highlight the strengths, limitations and practical effectiveness of the algorithm in real-world transportation problems. The paper ends with a conclusion highlighting the main outcomes derived from this study and references.

* Author for correspondence. e-mail: asad132@du.ac.bd

II. Preliminaries

In this section, we describe will discuss the concepts of transportation problems and idea of fuzzy set theory in addressing uncertainty within transportation models.

Transportation Problems

Linear Programming Problems help us resolve different types of optimization problems. One very popular and practical type is the Transportation Problem where we figure out the cheapest way to move products from factories to markets. The Transportation Problems (TP) is a specialized version of Linear Programming Problems (LPP) in Operations Research which focuses on ascertaining the most cost-effective way to transport equipment from various sources (like factories) to various destinations (like warehouses or markets) while satisfying supply and demand constraints[2,4].

Standard Mathematical Form

Let, m = number of sources/supply points/distribution centers/production sites

n = number of destinations/customer locations/demand nodes/delivery points

s_i = supply at distribution center i ; where $i = 1, 2, 3, \cdots\cdots\cdots, m$

d_j = Demand at delivery point j ; where $j = 1, 2, 3, \cdots\cdots\cdots, n$

c_{ij} = cost of transporting one unit from distribution center i to delivery point j

x_{ij} = quantity to be delivered from distribution center i to delivery point j

Objective Function

$$\text{Minimize } Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to

Supply Constraints: $\sum_{j=1}^{n} x_{ij} = s_i$; for each source $i = 1, 2, 3, \cdots\cdots\cdots, m$

Demand Constraints: $\sum_{i=1}^{m} x_{ij} = d_j$; for each destination $j = 1, 2, 3, \cdots\cdots\cdots, n$

Note: Each source can only supply up to its available capacity and each destination must receive exactly the amount it requires.

Non-Negativity Restrictions: $x_{ij} \geq 0$; where $i = 1, 2, 3, \cdots\cdots\cdots, m \& j = 1, 2, 3, \cdots\cdots\cdots, n$

Fuzzy Numbers in Transportation Problems

In Transportation Problems, the parameters such as cost, supply and demand are often subject to uncertainty. Fuzzy numbers are used to model uncertainty by assigning a range of values with associated degrees of membership rather than exact numerical values. This approach allows for more realistic decision-making in situations where precise data is unavailable or unreliable.

Fuzzy Sets and Membership Function

Fuzzy numbers are represented as fuzzy sets and each set has a membership function that defines the degree of membership for each possible value. Fuzzy sets are a fundamental concept in fuzzy mathematics which is introduced by Lotfi A. Zadeh in 1965 as an extension of classical set theory.

Various Types of Fuzzy Numbers

There are several types of fuzzy numbers such as:

Trapezoidal Fuzzy Numbers

A trapezoidal fuzzy number is defined by four parameters a, b, c and d; where a = lower bound, b = left peak, c = right peak, d = upper bound and w=membership value.

A fuzzy number $\tilde{A} = (a, b, c, d: w)$ is said to be a generalized trapezoidal fuzzy number if its membership function defined by:

$$\mu_{\tilde{A}} = \begin{cases} w\dfrac{(x-a)}{(b-a)}, & a \leq x \leq b \\ w, & b \leq x \leq c \\ w\dfrac{(x-d)}{(c-d)}, & c \leq x \leq d \\ 0, & Otherwise \end{cases}$$

Pentagonal Fuzzy Numbers

A pentagonal fuzzy number is defined by four parameters a, b, c, d and e; where a = lower bound of the fuzzy number, b = first point of increase (left leg of the pentagon), c = most likely (peak) value, d = First point of decrease (right leg of the pentagon), e = upper bound and w=membership value.

A fuzzy number $\tilde{A} = (a, b, c, d, e: w)$ is said to be a generalized pentagonal fuzzy number if its membership function defined by:

$$\mu_{\tilde{A}} = \begin{cases} 0, & x < a \text{ or } x > e \\ \frac{(x-a)}{(b-a)}, & a \le x < b \\ 1, & b \le x \le c \\ \frac{(e-x)}{(e-d)}, & d < x \le e \\ 0, & Otherwise \end{cases}$$

Standard Techniques to Solve Transportation Problems

There are various procedures available to solve the transportation problems (TP) and each procedure is appropriate for different types of problems. Among them the most frequently used solution techniques areLeast Cost Method, North West Corner Method, Vogel's Approximation Method and MODI's Optimality Method[10,11].

III. Proposed Penalty Based Algorithm for Transportation Problems

In this section, we proposed a penalty-based algorithm for solving Transportation Problems and also demonstrate the applications of the proposed method through some numerical examples[5,6].

The steps of the algorithm are as follows:

Step 1: Verify whether the transportation problem is balanced. In case the problem is not balanced then,

I. Add a dummy row when the overall demand exceeds the total supply or

II. Add a dummy column when the overall demand is lower than the total supply.

Hence, convert it to a balanced transportation problem i.e.

$$\sum_{i=1}^{s} \tilde{a}_i = \sum_{j=1}^{t} \tilde{b}_j$$

Step 2: Foreach row, subtract the smallest value from the largest value and divide the range by the product of the number of rows and columns i.e.

$$\text{Penalty} = \frac{\text{Maximum} - \text{Minimum}}{\text{Number of Rows} \times \text{Number of Columns}}$$

Step 3: Similarly, for each column, subtract the smallest value from the largest value and divide the range by the product of the number of rows and columns i.e.

$$\text{Penalty} = \frac{\text{Maximum} - \text{Minimum}}{\text{Number of Rows} \times \text{Number of Columns}}$$

Step 4: Determine the penalties in an alternating pattern of greatest and lowest values. Locate the cell with the minimum cost entry in the corresponding row or column and assign the minimum of supply and demand to that cell. When two or more rows or columns have the same maximum or minimum penalties or cost entries, any one of them can be chosen arbitrarily.

Step 5: Update supply and demand values to reflect the recent allocation. If the allocation fulfills the origin's supply, then remove the corresponding row. Otherwise eliminate the corresponding column if it satisfies the destination's demand.

Step 6: Proceed with steps two, three, four and five iteratively until the number of allocations reaches $(R + C - 1)$, where R is the number of rows and C is the number of columns and all supply and demand constraints are satisfied.

A Real-life example of Unbalanced transportation problem

Optimizing Inventory Flow to Outlets

A textile company manages four manufacturing companies such as Stock Point, Ware Space, Store Fusion and Cargo Pulse. These repositories deliver their goods to four outlets such as Outlet-1, Outlet-2, Outlet-3 and Outlet-4. Due to real-world uncertainties in production output, transportation management and consumer demand fluctuations, supply capabilities, expenses and demands of outlets are illustrated as Pentagonal Fuzzy Numbers which permit decision-makers to capture more comprehensive analysis of uncertainties through five characteristic points.

Solution: To solve this transportation problem, we will initially convert the generalized Pentagonal Fuzzy Numbers into crisp values by using Centroid Ranking Technique and check whether the transportation problem is balanced. If the problem is unbalanced, then we will make it balanced. After that, we will apply the proposed penalty-based algorithm step by step to find the best solution effectively.

Step 1: By applying the Centroid Ranking Technique, we have to transform the generalized Pentagonal Fuzzy Numbers into crisp values.If $\tilde{A} = (a, b, c, d, e; w)$ is a pentagonal fuzzy number, then

$$R(\widetilde{\alpha_A}) = \left(\frac{2a + 9b + 2c + 9d + 2e}{24}\right)\left(\frac{11w + 2}{24}\right)$$

Thus, the table for crisp values is presented in Table 2.

Table 1. Supply, Demand and Cost Data

Outlet → / Company ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Supply
Stock Point	(35,36, 37,38, 39; 0.95038)	(15,16, 17,18, 19; 0.84493)	(19,20, 21,22, 23; 0.85716)	(36,38, 40,42, 44; 0.6091)	(94,95, 96,97, 98;1)
Ware Space	(29,30, 31,32, 33; 0.87391)	(30,31, 32,33, 34; 0.7047)	(30,31, 32,33, 34; 0.84091)	(20,21, 22,23, 24; 0.7207)	(130, 131, 132, 133, 134; 0.99174)
Store Fusion	(27,28, 29,30, 32; 0.966)	(30,31, 32,33, 34; 0.5)	(23,24, 25,26, 27; 0.7904)	(27,28, 29,30, 31; 0.8113)	(80,81, 82,83, 84; 0.72284)
Cargo Pulse	(40,41, 42,43, 44; 0.85715)	(36,37, 38,39, 40; 0.8517)	(27,28, 29,30, 31; 0.94672)	(35,36, 37,38, 39; 0.9386)	(105, 106, 107, 108, 109; 0.83773)
Demand	(60,61, 62,63, 64; 0.76833)	(70,71, 72,73, 74; 0.8788)	(245,246, 247,248, 249; 0.96651)	(60,61, 62,63, 64: 0.87391)	

Table 2. Crisp Values of Pentagonal Fuzzy Numbers

Outlet → / Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Supply
Stock Point	19.2	8	10	14.5	52
Ware Space	15	13	15	9.1	71
Store Fusion	15.3	10	11.14	13.2	34
Cargo Pulse	20	18	15	19	50
Demand	27	35	107	30	

Step 2: Here, overall demand $= (27 + 35 + 107 + 30) = 199$ and overall supply $= (52 + 71 + 34 + 50) = 207$. Since overall **demand < overall supply**, we must balance the problem by adding a dummy outlet which will absorb the extra supply of $(207 - 199) = 8$ units. The cost of transportation to the dummy demand is always set to zero which helps ensure the balance between overall supply and demand. Therefore, the balanced transportation table is presented in Table 3.

Table 3. Balanced Table for the problem

Outlet → / Company ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply
Stock Point	19.2	8	10	14.5	0	52
Ware Space	15.0	13	15	9.1	0	71
Store Fusion	15.3	10	11.14	13.2	0	34
Cargo Pulse	20.0	18	15	19	0	50
Demand	27	35	107	30	8	

Step 3: Now we will calculate the penalty for each individual row and column by applying the formula. For table 3, row number = 4 and column number = 5.

Formula for the penalty

$$= \frac{\text{Maximum} - \text{Minimum}}{\text{Numberof Rows} \times \text{Number of Columns}}$$

The row and column penalties are shown in Table 4.

Table 4. First Iteration

Outlet → Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply	Penalty
Stock Point	19.2	8	10	14.5	0	52	$\frac{19.2-0}{4\times 5} = 0.96$
Ware Space	15	13	15	9.1	0	71	$\frac{15-0}{4\times 5} = 0.75$
Store Fusion	15.3	10	11.14	13.2	0	34	$\frac{15.3-0}{4\times 5} = 0.76$
Cargo Pulse	20	18	15	19	0	50	$\frac{20-0}{4\times 5} = 1$
Demand	27	35	107	30	8		
Penalty	$\frac{20-15}{4\times 5}$ $= 0.25$	$\frac{18-8}{4\times 5}$ $= 0.5$	$\frac{15-10}{4\times 5}$ $= 0.25$	$\frac{19-9.1}{4\times 5}$ $= 0.49$	$\frac{0-0}{4\times 5} = 0$		

Here, the **maximum** penalty 1 occurs in fourth row and the minimum c_{ij} in this row is c_{45}= 0 which occurs in cell (4, 5). Then the greatest allocation in this cell is minimum (50, 8) = 8 which fulfills the demand of dummy outlet and update the supply of Cargo Pulse from 50 to 42 (50-8 = 42). So, the column corresponding to Dummy Outlet is disregarded after meeting its demand**.** Thus, the revised table is displayed in Table 5.

Table 5. First Allocation

Outlet → Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply	Penalty
Stock Point	19.2	8	10	14.5	0	52	$\frac{19.2-8}{4\times 5} = 0.96$
Ware Space	15	13	15	9.1	0	71	$\frac{15-9.1}{4\times 5} = 0.75$
Store Fusion	15.3	10	11.14	13.2	0	34	$\frac{15.3-10}{4\times 5} = 0.76$
Cargo Pulse	20	18	15	19	0^{8}	$\cancel{50}^{42}$	$\frac{20-15}{4\times 5} = 1$
Demand	27	35	107	30	8^{0}		
Penalty	$\frac{20-15}{4\times 5}$ $= 0.25$	$\frac{18-8}{4\times 5}$ $= 0.5$	$\frac{15-10}{4\times 5}$ $= 0.25$	$\frac{19-9.1}{4\times 5}$ $= 0.49$	$\frac{0-0}{4\times 5} = 0$		

Step 4: In the updated table 5, row number = 4 and column number = 4.The row and column penalties are presented in Table 6.

Table 6. Second Iteration

Outlet → Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply	Penalty
Stock Point	19.2	8	10	14.5	0	52	$\frac{19.2-8}{4\times 4}=0.7$
Ware Space	15	13	15	9.1	0	71	$\frac{15-9.1}{4\times 4}=0.37$
Store Fusion	15.3	10	11.14	13.2	0	34	$\frac{15.3-10}{4\times 4}=0.33$
Cargo Pulse	20	18	15	19	0^8	42	$\frac{20-15}{4\times 4}=0.31$
Demand	27	35	107	30	0		
Penalty	$\frac{20-15}{4\times 4}=0.31$	$\frac{18-8}{4\times 4}=0.62$	$\frac{15-10}{4\times 4}=0.31$	$\frac{19-9.1}{4\times 4}=0.62$	–		

Here, the **minimum** penalty 0.31 occurs in fourth row, first and third column. Among the available choices, we are picking the value of fourth row. The minimum c_{ij} in this row is c_{43}= 15.000 which appears in cell (4,3). Then the greatest allocation in this cell is minimum (42,107) = 42 which satisfies the supply of Cargo Pulse and update the demand of Outlet-3 from 107 to 65 (107-42=65). **Since the row for Cargo Pulse has no remaining supply, it is eliminated from further analysis**.Hence, the modified table is displayed in Table 7.

Table 7. Second Allocation

Outlet → Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply	Penalty
Stock Point	19.2	8	10	14.5	0	52	$\frac{19.2-8}{4\times 4}=0.7$
Ware Space	15	13	15	9.1	0	71	$\frac{15-9.1}{4\times 4}=0.37$
Store Fusion	15.3	10	11.14	13.2	0	34	$\frac{15.3-10}{4\times 4}=0.33$
Cargo Pulse	20	18	15^{42}	19	0^8	$\cancel{42}^0$	$\frac{20-15}{4\times 4}=0.31$
Demand	27	35	107^{65}	30	0		
Penalty	$\frac{20-15}{4\times 4}=0.31$	$\frac{18-8}{4\times 4}=0.62$	$\frac{15-10}{4\times 4}=0.31$	$\frac{19-9.1}{4\times 4}=0.62$	–		

Similarly, we have to continue this procedure by calculating the penalties for the remaining rows and columns and selecting the penalties in an alternating pattern until the entire demand and supply requirements are fulfilled. After the ninth iteration, all the remaining demand and supply requirements will be satisfied. The newly updated table 8 shows that **no further allocations are needed and all requirements are satisfied.**

Table 8: Final Allocation

Outlet → Repository ↓	Outlet-1	Outlet-2	Outlet-3	Outlet-4	Dummy Outlet	Supply	Penalty
Stock Point	19.2	8^{35}	10^{17}	14.5	0	0	–
Ware Space	15^{27}	13	15^{14}	9.1^{30}	0	0	–
Store Fusion	15.3	10	11.14^{34}	13.2	0	0	–
Cargo Pulse	20	18	15^{42}	19	0^8	0	–
Demand	0	0	0	0	0		
Penalty	–	–	–	–	–		

The overall cost for the transportation problem is

$$\begin{aligned} &= 15.000 \times 27 + 8.000 \times 35 + 10.000 \times 17 \\ &\quad + 15.000 \times 14 + 11.140 \times 34 \\ &\quad + 15.000 \times 42 + 9.100 \times 30 + 0 \times 8 \\ &= 2346.8 \end{aligned}$$

IV. Computer Oriented Solution of Transportation Problems

The transportation problems are not restricted to small-scale cases. The constructed code is generated to effectively accommodate and solve problems of any size particularly tackling cost matrices and supply-demand vectors of dimensions $m \times n$. This scalability also verifies that the algorithm remains efficient and realistic for large and complex transportation networks encountered in real-world scenarios. ***MATLAB R2016a*** will allow a fair and accurate comparison of each method's performance in terms of solution quality, computational timeand cost minimization[7,9,12].

MATLAB Code for Proposed Penalty-Based Algorithm

```
function proposed_penalty_based_method_problem_01()
cost = [19.2,    8,   10,     14.5;
        15,     13,   15,      9.1;
        15.3,   10,   11.14,  13.2;
        20,     18,   15,     19];
supply = [52, 71, 34, 50];
demand = [27, 35, 107, 30];
repository = {'Stock_Point', 'Ware_Space', 'Store_Fusion', 'Cargo_Pulse'};
outlet = {'Outlet_1', 'Outlet_2', 'Outlet_3', 'Outlet_4'};

fprintf('Initial Supply:  [%s]\n', num2str(supply));
fprintf('Initial Demand:  [%s]\n', num2str(demand));
total_supply = sum(supply);
total_demand = sum(demand);
dummy_row = false;
dummy_col = false;
if total_supply > total_demand
fprintf('\nBalancing required: Supply (%d) > Demand (%d)\n', total_supply, total_demand);
cost(:, end+1) = 0;
demand(end+1) = total_supply - total_demand;
outlet{end+1} = 'Dummy_Outlet';
dummy_col = true;
fprintf('Dummy destination added with demand = %d\n', demand(end));
elseif total_demand > total_supply
fprintf('\nBalancing required: Demand (%d) > Supply (%d)\n', total_demand, total_supply);
cost(end+1, :) = 0;
supply(end+1) = total_demand - total_supply;
repository{end+1} = 'Dummy_Repository';
dummy_row = true;
fprintf('Dummy source added with supply = %d\n', supply(end));
else

fprintf('\nNo balancing required: Supply and Demand are already equal.\n');
end
[m, n] = size(cost);
extended_cost = [cost, supply(:)];
extended_cost = [extended_cost; [demand, NaN]];
row_labels = repository;
col_labels = outlet;
row_labels{end+1} = 'Total_Demand';
col_labels{end+1} = 'Total_Supply';
fprintf('\nBalanced Cost Matrix (with Supply & Demand):\n');
cost_table = array2table(extended_cost, ...
'VariableNames', col_labels, ...
'RowNames', row_labels);
disp(cost_table);
allocation = zeros(m, n);
active_rows = true(1, m);
active_cols = true(1, n);
total_allocs = m + n - 1;
allocations_done = 0;
pick_high = true;
while allocations_done < total_allocs
[row_p, col_p] = custom_penalty(cost, active_rows, active_cols);
all_penalties = [];
for i = 1:m
if row_p(i) >= 0
all_penalties(end+1, :) = [row_p(i), 1, i];
end
end
for j = 1:n
if col_p(j) >= 0
all_penalties(end+1, :) = [col_p(j), 2, j];
end
end
if isempty(all_penalties)
break;
end
if pick_high
[~, idx] = max(all_penalties(:, 1));
else
[~, idx] = min(all_penalties(:, 1));
end
chosen = all_penalties(idx, :);

pick_high = ~pick_high;
if chosen(2) == 1
i = chosen(3);
available = find(active_cols);
[~, jrel] = min(cost(i, available));
j = available(jrel);
else
j = chosen(3);
available = find(active_rows);
[~, irel] = min(cost(available, j));
i = available(irel);
end
alloc_amt = min(supply(i), demand(j));
allocation(i, j) = alloc_amt;
supply(i) = supply(i) - alloc_amt;
demand(j) = demand(j) - alloc_amt;
if supply(i) == 0
active_rows(i) = false;
end
if demand(j) == 0
active_cols(j) = false;
end
allocations_done = allocations_done + 1;

end
fprintf('\nAllocation Matrix (Proposed Penalty-Based Transportation Method):\n');
allocation_table = array2table(round(allocation, 1), ...
'VariableNames', outlet, ...
'RowNames', repository);
disp(allocation_table);
total_cost = sum(sum(allocation .* cost));
fprintf('\nTotal Transportation Cost: %.1f\n', total_cost);
non_zero_allocs = nnz(allocation);
expected_allocs = m + n - 1;
fprintf('\nNon-Zero Allocations: %d\n', non_zero_allocs);
fprintf('Expected Allocations (m + n - 1): %d\n', expected_allocs);
if non_zero_allocs == expected_allocs
fprintf('The solution is NON-DEGENERATE.\n');
elseif non_zero_allocs < expected_allocs
fprintf('The solution is DEGENERATE.\n');
else
fprintf('Invalid solution: Too many basic variables.\n');
end
end
function [row_p, col_p] = custom_penalty(cost, active_rows, active_cols)

[m, n] = size(cost);
row_p = -ones(1, m);
col_p = -ones(1, n);
for i = 1:m
if active_rows(i)
vals = cost(i, active_cols);
if length(vals) > 1
row_p(i) = (max(vals) - min(vals)) / (m * n);
```

```
else
row_p(i) = 0;
end
end
end
for j = 1:n
if active_cols(j)
vals = cost(active_rows, j);
if length(vals) > 1
col_p(j) = (max(vals) - min(vals)) / (m * n);
else
col_p(j) = 0;
end
end
end
end
```

Output:

```
>> proposed_penalty_based_method_problem_01
Initial Supply:  [52  71  34  50]
Initial Demand:  [27   35  107   30]

Balancing required: Supply (207) > Demand (199)
Dummy destination added with demand = 8

Balanced Cost Matrix (with Supply & Demand):
                    Outlet_1    Outlet_2    Outlet_3    Outlet_4    Dummy_Outlet    Total_Supply
                    ________    ________    ________    ________    ____________    ____________

    Stock_Point     19.2          8           10        14.5        0                52
    Ware_Space        15         13           15         9.1        0                71
    Store_Fusion    15.3         10        11.14        13.2        0                34
    Cargo_Pulse       20         18           15          19        0                50
    Total_Demand      27         35          107          30        8               NaN

Allocation Matrix (Proposed Penalty-Based Transportation Method):
                    Outlet_1    Outlet_2    Outlet_3    Outlet_4    Dummy_Outlet
                    ________    ________    ________    ________    ____________

    Stock_Point      0           35          17           0          0
    Ware_Space      27            0          14          30          0
    Store_Fusion     0            0          34           0          0
    Cargo_Pulse      0            0          42           0          8


Total Transportation Cost: 2346.8

Non-Zero Allocations: 8
Expected Allocations (m + n - 1): 8
The solution is NON-DEGENERATE.
>>
```

V. Comparison with Existing Methods

By using ***MATLAB R2016a*** for the Existing, MODI's Optimality Test and Proposed Penalty-Based Algorithm for Example 1, we obtain the following results which are presented in Table 9.

Table 9. Performance Comparison

Methods	**Transportation Costs (in Dollars)**
Least Cost Method	2386.8
North West Corner Method	2892.2
Vogel's Approximation Method	2346.8
MODI's Optimality Test	2346.8
Proposed Method	**2346.8**

According to the table provided above, it is clearly seen that, our proposed method is giving better results than all of the existing methods so we can express it like this way,

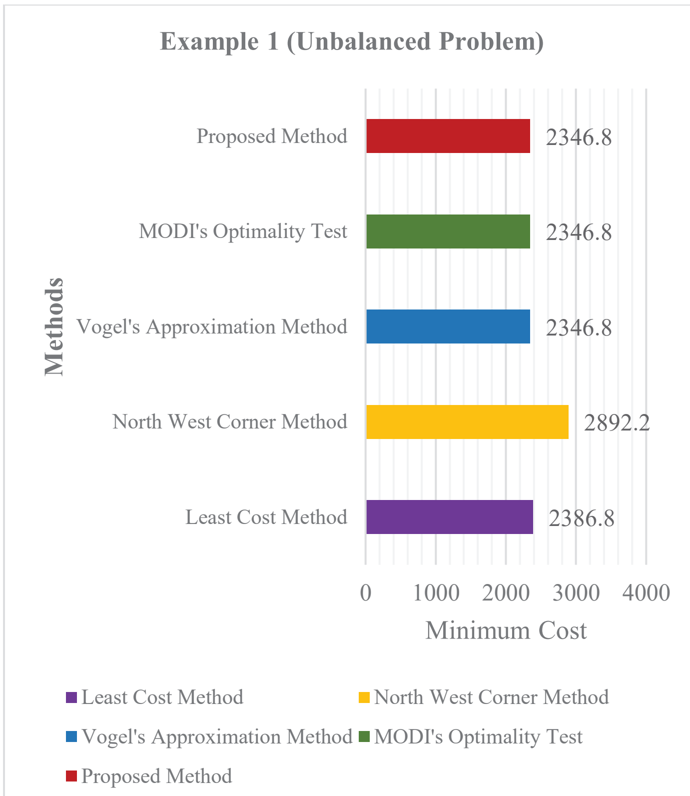
$$\text{North West Corner Method} >> \textit{Least Cost Method} > \textit{Proposed Method}$$

It is worth noting that, the solution of our Proposed Method is exactly equal to the MODI's Optimality Test which implies that our proposed method is giving the optimal solution.

$$\text{Vogel's Approximation Method=MODI's Optimality Test} = \text{Proposed Method}$$

Therefore, our Proposed Penalty-Based Method provides the optimal solution.

Bar Graph Analysis of Algorithms for Example (Unbalanced TP)



As illustrated in the bar graph, the proposed penaltybased algorithm not only yields the same result as Vogel's Approximation Method and MODI's Optimality Test but also demonstrates a significant advantage over the traditional methods such as Least Cost Method and North-West Corner Method.

Similarly, this proposed algorithm is equally applicable to balanced TransportationProblems where the proposed algorithm remains consistent and provides the optimal result.The associated MATLAB code also maintains its flexibility across problems of varying sizes and dimensions.

VI. Conclusion

We have proposed a new penaltybased algorithm and applied this algorithm to a numerical example of transportation problems and compared the results with other existing methods. Our main focus is to emphasize that the proposed algorithm yields better results compared to other existing methods for both transportation problems. The proposed algorithm demonstrates its best applicability in solving transportation problems across various scenarios including both balanced and unbalanced cases.This proposed method is an enhanced version of Vogel's Approximation Method which produces an optimal solution similar to the one produced by MODI's Optimality Test whileavoiding the dependency on an Initial Basic Feasible Solution.Our proposed method provides optimal results for both transportation Problems by reducing solution time and markedly lowering computational effort compared to the existing methods. This improved algorithm is particularly wellsuited for addressing transportation problems across a wide range of fields such as logistics, supply chain optimization, warehouse management and resource distribution. Its versatility enables it to efficiently handle both balanced and unbalanced transportation problems including those with varying cost structures, fluctuating supply and demand and transportation capacity constraints. This also makes it a valuable tool for industries such as manufacturing, retail, energy distribution and humanitarian logistics where optimizing cost, time and resources is critical to achieving operational efficiency and sustainability.

References

1. Ali Ebrahimnejad, 2014. A simplified new approach for solving fuzzy transportation problems with generalized trapezoidal fuzzy numbers, Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Applied Soft Computing February **19**, 171-176.
2. Yong Denga, Yuxin Chenb, Yajuan Zhanga, Sankaran Mahadevanc, 2012. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, Applied Soft Computing, November **12**, 1231-1237.
3. Kenan Cetin, GulfemTuzkaya, OzalpVayvay, 2020. A mathematical model for personnel task assignment problem and an application for banking sector, An International Journal of Optimization and Control: Theories & Applications, April **10 (2)**, 147-158.
4. Amir Mohammad Fathollahi-Fard & Mostafa Hajiaghaei-Keshteli, 2018. Integrated Capacitated Transportation and Production Scheduling Problem in a Fuzzy Environment, June **29 (2)**, 197-211.
5. Amarpreet Kaur, Amit Kumar, 2011. A new approach for solving fuzzy transportation problems using generalized trapezoidal fuzzy numbers, Applied Soft Computing **12**, 1201-1213.
6. Sapan Kumar Das, 2020. Application of Transportation Problem under Pentagonal Neutrosophic Environment, Journal of Fuzzy Extension and Applications, January **1 (1),** 27-40.
7. Stephen Dinagar, D. R. Keerthivasan, 2018. Solving Fuzzy Transportation Problem Using Modified Best Candidate Method, Journal of Computer and Mathematical Sciences, September **9 (9)**, 1179-1186.
8. Apurba Panda, Madhumangal Pal, 2016. A study on pentagonal fuzzy number and its corresponding matrices, Pacific Science Review B: Humanities and Social Sciences, November **1(3),** 131-139.
9. Jhon Paul AntonyR., S. Johnson Savarimuthu, T. Pathinathan, 2014. Method for Solving the Transportation Problem Using Triangular Intuitionistic Fuzzy Number, International Journal of Computing Algorithm, February **3**, 590-605.
10. Anila Gupta,Amit Kumar, 2012. A new method for solving linear multi-objective transportation problems with fuzzy parameters, Applied Mathematical Modelling, April **36 (4),** 1421-1430.
11. Ismail MohideenS. and P. Senthil Kumar, 2010. A Comparative Study on Transportation Problem in Fuzzy Environment, International Journal of Mathematics Research, November **10 (1),** 151-158.
12. MuruganandamDr. S. and R.Srinivasan, 2016. Optimal Solution for Multi-Objective Two Stage Fuzzy Transportation Problem, Asian Journal of Research in Social Sciences and Humanities, May **6 (5),** 744-752.
13. Abouzar Sheikhi and Mohammad Javad Ebadi, 2023. On Solving Linear Fractional Programming Transportation Problems with Fuzzy Numbers, Journal of Fuzzy Extension and Applications, September **4 (4)**, 327-339.