https://mathco.journals.pnu.ac.ir

Payame Noor University

Control and Optimization in Applied Mathematics
Volume 11, Issue 2, Summer-Autumn 2026, Pages 65–82
DOI:10.30473/coam.2026.76242.1350

OPEN ACCESS

Research Article

Inverse Balanced Facility Location Problem in the Plane

Sina Nemati **Jafar Fathali** ✉ **Abolfazl Poureidi**

Faculty of Mathematical Sciences, Shahrood University of Technology, University Blvd., Shahrood, Iran

Article History

Received: October 26, 2025
Accepted: March 9, 2026
Available online: May 21, 2026

Abstract. Classical inverse location models aim to modify problem parameters such that pre-specified facility locations become optimal with respect to a given objective. This paper addresses a fundamentally different variant: the *inverse balanced facility location problem in the Euclidean plane*, in which parameters are adjusted so as to achieve an equitable distribution of client demand between two given facilities. Specifically, given a set of n weighted points in the plane and two predetermined facility locations, the objective is to minimally modify either the weights or the coordinates of the client points such that the absolute difference in total demand assigned to each facility—referred to as the *unbalancing number*—is minimized. For the weight-modification case, we establish that the planar problem is structurally equivalent to its network counterpart and is therefore solvable in $O(n \log n)$ time under any L_p norm, via an existing linear programming formulation. For the coordinate-modification case under the Euclidean norm, we exploit the isometric property of orthogonal rotations to prove that thetwo-dimensional problem reduces, without loss of generality, to a one-dimensional problem along the perpendicular bisector of the segment joining the two facilities. Leveraging this reduction, we design three novel greedy algorithms—IFLP1, IFLP2, and IFLP3—that prioritize minimization of the unbalancing number, minimization of the total transfer cost, and a hybrid criterion balancing both objectives, respectively. Under uniform weights and identical modification costs, all three algorithms are proven to yield optimal solutions and operate within $O(n^2)$ time complexity. Extensive computational experiments on standard benchmark datasets and randomly generated instances demonstrate that IFLP1 achieves the lowest CPU time and smallest unbalancing number, while IFLP3 yields superior performance in termsof total transfer cost and is recommended for practical applications.

Keywords. Facility location, Inverse facility location, Balanced allocation, Greedy algorithm, Coordinate modification, Euclidean plane.

MSC. 90B90; 90B06.

How to Cite
Nemati, S., Fathali, J., Poureidi, A. (2026). "Inverse balanced facility location problem in the plane". *Control and Optimization in Applied Mathematics*, 11(2), 65-82. `https://doi.org/10.30473/coam.2026.76242.1350`

✉Corresponding author. **E-mail:** fathali@shahroodut.ac.ir

1 Introduction

In recent years, there has been a growing interest in equity location models. These models tackle facility location problems with the goal of maximizing equality in serving demand points. Numerous authors have explored this subject. For instance, Gavalec and Hudec [12] focused on the balancing function model, which aims to maximize the difference in distance between a demand point and its farthest and nearest facility. Berman et al. [3] studied the problem of locating p facilities to minimize the maximum weight assigned to any single facility. Marin [20] later introduced the balanced location problem, optimizing the weight disparity between the most and least loaded facilities. Fathali and Zaferanieh [10] further advanced this line of work by proposing polynomial-time algorithms for balanced location models on tree networks. Lejeune and Prasad [19] examined the trade-off between effectiveness and equity by presenting a bi-criteria model for this problem. Landete and Marin [18] focused on minimizing the differences among the weights allocated to the facilities. Additionally, Barbati and Piccolo [5] presented properties describing the behavior of equality measures in facility location models. For further reading on equity measurement in location theory, refer to the literature reviews by Marsh and Schilling [21] and Eiselt and Laporte [8].

The classical location models focus on finding the optimal locations for facilities. However, in certain scenarios, the facilities may already be in place, and the objective becomes improving these existing locations by modifying some parameters. When the goal is to change the parameters with minimum cost and at the same time optimize the given facility locations, this problem is called the inverse location problem.

There is extensive research on the network version of the inverse facility location problem. Among them Burkard et al. [6] investigated the inverse median problem and presented an $O(n \log n)$ algorithm for this problem on a tree network. The median problem asks to find the location of a facility such that the sum of weighted distances from clients to the closest facility is minimized. Galavii [11] reduced the time complexity of the inverse median problem on trees to linear time. Burkard et al. [7] proposed an $O(n^2)$ algorithm for solving the inverse median problem on cycles. Later work by Guan and Zhang [17] and Wu et al. [28] extended this framework to tree networks, considering variable vertex weights and edge lengths using Chebyshev and Hamming distances, respectively. Omidi et al. [25] and Nazari and Fathali [22] investigated the inverse balanced 2-facility location problems on a tree, aiming to modify edge lengths and vertex weights, respectively, such that the difference in total weights assigned to the facilities is minimized. They proposed $O(n \log n)$-time algorithms to solve these problems. In another study, Omidi and Fathali [24] examined the inverse single-facility location problem on a tree, where the goal is to balance the distances from the server to clients by modifying edge lengths. For this problem, they developed $O(n \log n)$ and $O(n^2)$-time algorithms for the unbounded and bounded edge-length cases, respectively. Recently, Sayar et al. [26] studied a special case of the inverse location problem where weights are transferred between vertices (without reduction or augmentation), thus preserving the total weight of the system. For this scenario, they formulated a linear programming model to solve the inverse median problem with weight transfers.

Authors have given less attention to the continuous version of the inverse facility location problem compared to the network version. Most inverse continuous facility location problems are devoted to the inverse minisum and minimax facility location models. The goals of minisum and minimax facility location problems are minimizing the sum and maximum weighted distances between clients and facilities, respectively. Burkard et al. [6] addressed the inverse minisum single facility location problem with variable weights. They presented an $O(n \log n)$-time algorithm for solving this problem under the rectilinear norm. For cases where distances are measured using the Euclidean norm, Burkard et al. [4] formulated a linear programming model. They demonstrated that the unit-cost model can be solved in $O(n \log n)$ time. Baroughi-Bonab et al. [1] investigated the inverse minisum single facility location problem with variable coordinates of the given points. They proved that the problems with rectilinear and Chebyshev norms are NP-hard. Nguyen et al. [23] considered the inverse minimax single facility location problem with the Chebyshev norm. The general case of inverse continuous facility location problems with variable weights and coordinates have been developed by Fathali [9] and Tour-Savadkoohi and Fathali [27], respectively.

They proposed efficient recursive algorithms which solve mathematical models in each iteration to modify the weights or coordinates of points. Gholami and Fathali [13, 15] and [14], respectively have explored two additional continuous inverse location models: the inverse minimax and minisum circle location problems. In these problems, the objective is to modify the weights assigned to specific points at minimal cost, with the aim of minimizing the maximum and sum of weighted distances from the circumference of a given circle, respectively. Recently, Golpayegani and Fathali [16] proposed some greedy algorithms for the inverse minimax line location problems with variable weights and coordinates.

The inverse balanced facility location problem in the plane has not been previously investigated. To address this gap, in this paper we study this problem assuming that the locations of two facilities are given, and that by modifying certain parameters of the given points, a balance is achieved among the clients assigned to the facilities. In the case that the weights of points are changed, we observe that the problem is equivalent to the network version which Nazari and Fathali [22] show can be solved in $O(n \log n)$ time. On the other hand, the problem with modifying the coordinates of points is more challenging than the problem with adjusting their weights. By leveraging properties specific to the Euclidean norm, we propose three greedy algorithms for this case and compare their experimental results.

Note that the inverse balanced facility location problem differs from the inverse minisum and minimax location problems, in which the parameters are modified so that the sum or maximum weighted distance between the given points and a facility is minimized.

This paper is organized as follows. In Section 2 the inverse balanced facility location problem is defined. The problem with variable weights of points is considered in Section 3. In Section 4, we investigate the inverse balanced facility location problem with variable coordinates and propose three greedy algorithms. In Section 5 we report the results of computational experiments.

2 Problem Description

Let a set of n points $\Omega = \{A_i | \ A_i = (a_i, b_i), \ i = 1, ..., n\}$ be given in the plane. Suppose these points are the location of clients and for $i = 1, ..., n$ the demand of client on A_i is represented by w_i. For any $N \subseteq \Omega$, let $W(N) = \sum_{A_i \in N} w_i$ be the total weight of all points in N. For any pair of points, A_i and A_j in Ω, let $d_{ij} = d(A_i, A_j)$ be the distance between points A_i and A_j. Let m_1 and m_2 be the locations of two given facilities in the plane. Let

$$N_1 = \{A_i \in \Omega | d(A_i, m_1) < d(A_i, m_2)\},$$

and $N_2 = \Omega \setminus N_1$, be the sets of clients assigned to facilities m_1 and m_2, respectively. In the inverse balanced facility location model, we want to change the weights or coordinates of the given points with minimum cost such that the difference of the total weights of vertices in N_1 and N_2 is minimized. Indeed, we consider the following objective functions:

$$\min f(C), \tag{1}$$

$$\min |W(N_1) - W(N_2)|, \tag{2}$$

where $f(C)$ is the total cost of changing weights or coordinates of given points. This type of inverse location problem has not been previously studied in the plane. It has many applications in the real world, such as balancing delivery workloads between two warehouses or distribution centers to reduce operational costs, allocating users to two server nodes in a 5G network to prevent congestion, etc. In these examples, a planner intervenes with minimal cost to make the two facilities' workloads as equal as possible, promoting fairness and efficiency. In the weight-modification case, a demand-side intervention occurs and we change how much work originates from each client

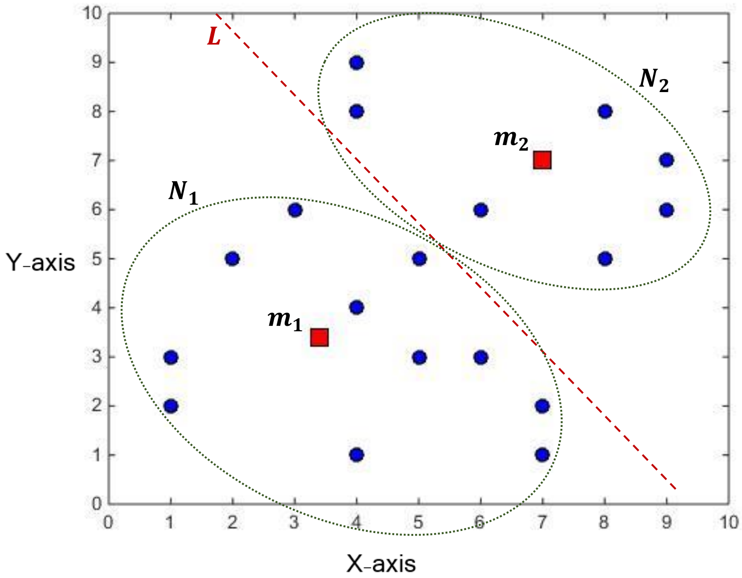


Figure 1: The given facilities m_1 and m_2, the set of N_1 and N_2, and the line L, the perpendicular bisector of the line connecting m_1 and m_2.

point. In the coordinate-modification case a supply-side or geographic intervention is considered. We effectively change the assignment of clients to facilities by altering their positions in the network. This can involve physical changes or redefining service territories.

In Section 3, we discuss the case of modifying the weights of given points and show that the problem can be solved in $O(n \log n)$ time. In Section 4, by considering line L, the perpendicular bisector of the segment connecting m_1 and m_2, and rotating coordinates, we reduce the problem with variable coordinates to the one-dimensional case. We then propose three greedy algorithms to solve this problem. Figure 1 illustrates the sets N_1 and N_2, along with line L.

3 Inverse Problem with Variable Weights

Consider the case in which the weights of points should be changed to achieve the balancing. For any point $A_i \in \Omega$, let c_i^+ and c_i^- be the increasing and decreasing cost per unit of w_i, respectively. Let q_i^+ and q_i^- be the values by which the weight w_i is increased and decreased, respectively. We suppose that q_i^+ has an upper bound u_i. In the ideal case, the weights of points would be modified such that $W(N_1) = W(N_2)$. This would ensure a balanced distribution of weights between the two facilities. Thus, we would consider the following problem.

$$\min f_1(C) = \sum_{i=1}^{n} (c_i^+ q_i^+ + c_i^- q_i^-) \tag{3}$$

$$s.t.$$

$$|W(\hat{N}_1) - W(\hat{N}_2)| = 0, \tag{4}$$

$$0 \le q_i^+ \le u_i, \quad i = 1, ..., n, \tag{5}$$

$$0 \le q_i^- \le w_i, \quad i = 1, ..., n. \tag{6}$$

where $\hat{N}_1$ and $\hat{N}_2$ are the modified sets N_1 and N_2, respectively, after changing the weights of points.

Indeed, it can be seen that the inverse balanced problem with weight modifying in the plane is equivalent to the inverse balanced facility location problem on a network, as studied by Nazari and Fathali [22]. This means that their solution approaches and methodologies can be applied to address the inverse balanced problem in the plane. They obtained the following linear programming model and proposed an $O(n \log n)$ algorithm to solve the problem.

$$\min f_1(C) = \sum_{i=1}^{n} (c_i^+ q_i^+ + c_i^- q_i^-) \tag{7}$$

$$s.t.$$

$$\sum_{A_i \in N_1} (q_i^+ - q_i^-) - \sum_{A_i \in N_2} (q_i^+ - q_i^-) = W(N_2) - W(N_1), \tag{8}$$

$$0 \le q_i^+ \le u_i, \quad i = 1, ..., n. \tag{9}$$

$$0 \le q_i^- \le w_i, \quad i = 1, ..., n. \tag{10}$$

Note that since in this case distances are not changed, thus this linear programming model and the method in [22] can be applied for any arbitrary norm.

4 Inverse Problem with Variable Coordinates

In this section, for balancing facilities, we consider the case where the coordinates of the given points are changed and the distances are measured by Euclidean norm. Note that if $W(N_1) = W(N_2)$ then the servers are balanced and the coordinates of points remain unchanged. Otherwise, without loss of generality, let $W(N_1) > W(N_2)$. Let L be the perpendicular bisector of the line connecting m_1 and m_2. Then the following property holds.

Lemma 1. To find the optimal solution, some points in N_1 should be moved to N_2 with minimum cost.

Proof. Since $W(N_1) > W(N_2)$, to achieve balance, some points in N_1 must be moved to N_2. For an optimal solution, the points whose total transfer cost is minimum should be selected. □

Before presenting the solution method, we show that the problem can be reduced to one dimension.

Lemma 2. Let T be an orthogonal matrix corresponding to a rotation in R^2, then in the case that distances are measured by L_2 norm, this rotation is an isometric transformation, i.e. the distances between any two points before and after rotation are the same.

Proof. Let p and q be two points in R^2, and $p' = Tp$ and $q' = Tq$ be the new points after rotation, respectively. Then

$$d(p', q') = ||Tp - Tq||_2 = ||T(p-q)||_2 = \sqrt{T(p-q)^t T(p-q)} = \sqrt{(p-q)T^t T(p-q)}.$$

Since T is orthogonal, then $T^t T = I$, so

$$d(p', q') = \sqrt{(p-q)(p-q)} = d(p, q).$$

□

Note that the property in Lemma 2 does not hold for other L_p norms. For example, consider the L_1 distance. Let $p = (0, 0)$ and $q = (1, 0)$, then $d_1(p, q) = ||p - q||_1 = 1$. When rotating by $\frac{\pi}{4}$, the rotation matrix is

$$T(\frac{\pi}{4}) = \begin{bmatrix} \cos\frac{\pi}{4} & \sin\frac{\pi}{4} \\ -\sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Then $p' = (0, 0)$ and $q' = \frac{1}{\sqrt{2}}(1, 1) \approxeq (0.7071, 0.7071)$. So $d_1(p', q') \approxeq 1.4142$.

Theorem 1. The inverse balanced facility location problem in the plane with Euclidean norm can be reduced to a one-dimensional problem.

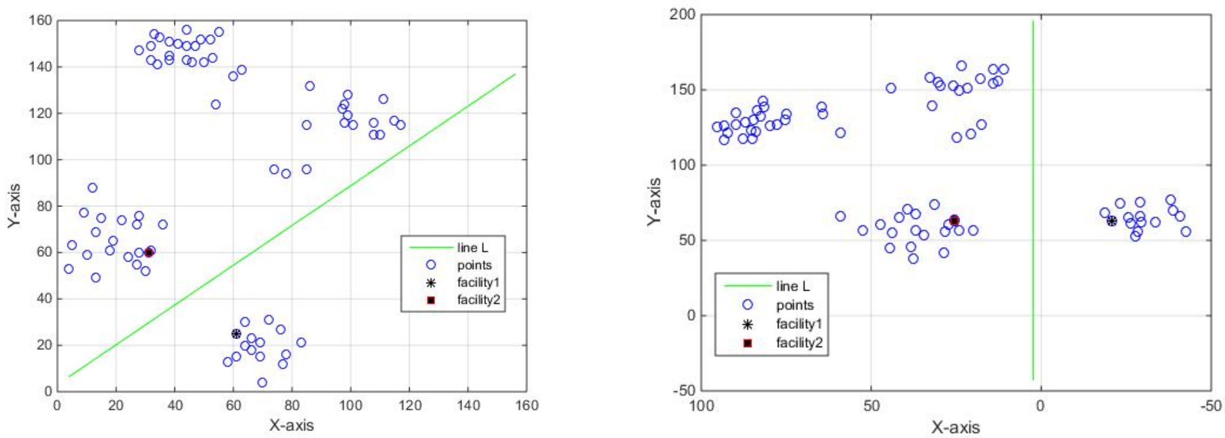
Proof. We move the origin to point O_1 and rotate the y-axis to coincide with the line L. Where O_1 is the midpoint of the line connecting m_1 and m_2. Let θ be the anti-clockwise angle between the line L and the positive direction of the y-axis. The rotation matrix is as follows

$$T(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}.$$

Moreover, by transferring the origin to the point $O_1 = (o_1, o_2)$ the new coordinate of any point $A = (a, b)$ can be computed as follows

$$A_{new} = (a - o_1, b - o_2).$$

After these operations, the origin and y-axis coincide with O_1 and line L, respectively. By Lemma 2 the distances between any two points before and after rotation are the same. Moreover, the points in N_1 remain in the same side of y-axis and the points in N_2 lie on the other side. Thus, after rotation, to move a point $A = (a, b)$ from N_1 to N_2, we only need to change the value of a, see Figure 2. On the other hand, using Lemma 1, to find the optimal balancing we should move the points from N_1 to N_2 with minimum cost. So to solve the problem, we just need to transfer points along the x-axis. □



(a) (b)

Figure 2: (a) The initial points and line L. (b) The points after rotating.

Subroutine *Initial* computes the new coordinates of all points after rotating the y-axis and translating the origin.

Suppose the line L coincide with y-axis and the points that lie on the line L are assigned to m_2. For any point $A_i \in \Omega$, let c_i be the cost of modifying per unit of a_i. To reach balancing, we should move some points in N_1 to L with minimum cost. The weighted distance between a_i and L is $w_i|a_i|$. Thus, moving a_i to N_2 costs $c_i w_i |a_i|$.

Let $n_1 = |N_1|$ and $n_2 = |N_2|$. For $i = 1, ..., n_1$, let $d_i = c_i|a_i|$ and $K = |W(N_1) - W(N_2)|$, we call K the unbalancing number.

The goal is minimizing the following objective functions.

1. The transfer cost.
2. The unbalancing number.

Algorithm 3 Initial(m_1, m_2, Ω)

1. **Input:** The set of n given points, $\Omega = \{A_1, \ldots, A_n\}$; two points m_1 and m_2 as facility locations.
2. **Output:** Rotated coordinates of points such that the midpoint of m_1 and m_2 coincides with the origin and the perpendicular bisector of the segment m_1m_2 coincides with the y-axis; the partitioned sets $\hat{N}_1$ and $\hat{N}_2$.
3. $N_1 \leftarrow \{A_i \in \Omega \mid d(A_i, m_1) < d(A_i, m_2)\}$.
4. $N_2 \leftarrow \Omega \setminus N_1$.
5. **If** $W(N_1) = W(N_2)$, **return** "Current coordinates are optimal."
6. **Else if** $W(N_1) > W(N_2)$, **then** $\hat{N}_1 \leftarrow N_1; \quad \hat{N}_2 \leftarrow N_2$.

   Else $\hat{N}_1 \leftarrow N_2; \quad \hat{N}_2 \leftarrow N_1$.
7. **For** each point $A_i \in \hat{N}_1$ such that $d(A_i, m_1) = d(A_i, m_2)$, **set:**

   $\hat{N}_1 \leftarrow \hat{N}_1 \setminus \{A_i\}, \quad \hat{N}_2 \leftarrow \hat{N}_2 \cup \{A_i\}$.
8. Update $W(\hat{N}_1)$ and $W(\hat{N}_2)$.
9. $L \leftarrow$ the perpendicular bisector of segment m_1m_2.
10. $\theta \leftarrow$ the counter-clockwise angle from line L to the positive y-axis.
11. $O_1 = (o_1, o_2) \leftarrow$ the midpoint of segment m_1m_2.
12. **For** $i \leftarrow 1$ **to** n **do:**
    - $\begin{bmatrix} \hat{a}_i \\ \hat{b}_i \end{bmatrix} \leftarrow \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_i \\ b_i \end{bmatrix}$,
    - $\hat{a}_i \leftarrow \hat{a}_i - o_1$,
    - $\hat{b}_i \leftarrow \hat{b}_i - o_2$.
13. **Return** $\hat{N}_1$, $\hat{N}_2$, and the new coordinates of all points.

These two objective functions are in contrast. To solve the considered inverse location problem, we propose the following three greedy strategies:

1. Transferring the points that reduce the unbalancing number as much as possible.
2. Moving some points from N_1 to N_2 in which the cost of their weighted distances to L are less than others.
3. Moving the points based on hybrid of minimizing transfer cost and unbalancing number.

Consider the first strategy. In each iteration of this strategy, we move a point from N_1 to N_2, which causes the largest reduction in the unbalancing number. For this aim, let $K = W(N_1) - W(N_2)$ and for $A_i \in N_1$, $k_i = |K - 2w_i|$. Then the point with minimum k_i should be moved. If this minimum happened for two or more points, the points with minimum transfer cost $r_i = |a_i|c_i w_i$ is moved. Note that, in some cases moving a point from N_1 to N_2, may causes a worse balancing than previous iteration. Thus we should check this case before moving the point. The algorithm terminates if the unbalancing number becomes zero or becomes larger than the unbalancing number in the previous iteration. These ideas lead us Algorithm 4.

Note that the Iteration step terminates at most in $(n-2)$ iterations. The worst case is when $N_2 = \{m_2\}$ and m_1 has a large demand weight. Then all points in N_1, except m_1, should be transported to N_2. In this case, steps 5 and 6 of the iteration step require $O(n)$ time, thus the time complexity of the algorithm is $O(n^2)$.

The second strategy is similar to the first, except that it minimizes the transfer cost. In each iteration of this strategy, the point with minimum cost of weighted distance is chosen to move to N_2. We call the algorithm of this strategy IFLP2. IFLP2 is the same as IFLP1, except that the set M and index t in the initial and iteration steps should be calculated as follows:

$$M = \{s | r_s = min_{\hat{A}_i \in \hat{N}_1} r_i\},$$

$$t = argmin\{k_i^{(j)} | i \in M\}.$$

The first and second strategies move the points based on minimizing unbalancing number and transfer cost, respectively. However, reducing both simultaneously may yield better solutions. Thus, as the third strategy, we consider both transfer cost and unbalancing number. To reduce both of them simultaneously, we move the point with minimum $k_i r_i$. Thus, in algorithm IFLP3, the set M and index t should be computed as follows:

$$M = \{s | r_s k_s^{(j)} = min_{\hat{A}_i \in \hat{N}_1} r_i k_i^{(j)}\},$$

$$t = argmin\{k_i^{(j)} | i \in M\}.$$

A comparison summarizing the differences between IFLP1-3 algorithms is given in Table 1.

Table 1: Comparison of the three proposed algorithms.

Objective Function Criterion	**IFLP1**	**IFLP2**	**IFLP3**
Transfer cost	No	Yes	Yes
Unbalancing number	Yes	No	Yes

Note that the time complexity of algorithms IFLP2 and IFLP3 is $O(n^2)$, the same as time complexity of IFLP1, which earlier discussed.

Theorem 2. If all the weights and the cost of changing the coordinates of the points are the same, then the three algorithms IFLP1, IFLP2 and IFLP3 find the same results. Moreover, the obtained solution is optimal.

Algorithm 4 IFLP1: Inverse Facility Location for Weight Balancing via Coordinate Modification

1. **Input:** n points $A_1, \ldots, A_n$ with weights w_i; two facility locations m_1 and m_2; coordinate modification cost c_i for each point.
2. **Output:** New coordinates for the points that balance the total weight assigned to facilities m_1 and m_2.
3. $(\hat{N}_1, \hat{N}_2) \leftarrow$ **Initial**(m_1, m_2, Ω).
4. $K^{(0)} \leftarrow W(\hat{N}_1) - W(\hat{N}_2)$.
5. $\hat{N}_1 \leftarrow \hat{N}_1 \setminus \{m_1\}$.
6. **For** each point $\hat{A}_i \in \hat{N}_1$ **do:**
   - $k_i^{(1)} \leftarrow |K^{(0)} - 2w_i|$,
   - $r_i \leftarrow |\hat{a}_i|\, c_i\, w_i$.
7. $M \leftarrow \{s \mid k_s^{(1)} = \min_{\hat{A}_i \in \hat{N}_1} k_i^{(1)}\}$.
8. $t \leftarrow \arg\min\{r_i \mid i \in M\}$.
9. $K^{(1)} \leftarrow k_t^{(1)}$.
10. $j \leftarrow 0, \quad f \leftarrow 0$.
11. **While** $K^{(j)} \neq 0$ **and** ($j = 0$ **or** $K^{(j)} < K^{(j-1)}$) **do:**

    11.1 $\hat{N}_1 \leftarrow \hat{N}_1 \setminus \{\hat{A}_t\}$,

    11.2 $\hat{N}_2 \leftarrow \hat{N}_2 \cup \{\hat{A}_t\}$.

    11.3 **If** $W(\hat{N}_1) < W(\hat{N}_2)$ **then:**

    $\hat{N}_3 \leftarrow \hat{N}_1, \quad \hat{N}_1 \leftarrow \hat{N}_2, \quad \hat{N}_2 \leftarrow \hat{N}_3$.

    11.4 $\hat{a}_t \leftarrow 0, \quad f \leftarrow f + r_t, \quad j \leftarrow j + 1$.

    11.5 **For** each point $\hat{A}_i \in \hat{N}_1$ **do:** $k_i^{(j)} \leftarrow |K^{(j-1)} - 2w_i|$.

    11.6 $M \leftarrow \{s \mid k_s^{(j)} = \min_{\hat{A}_i \in \hat{N}_1} k_i^{(j)}\}$.

    11.7 $t \leftarrow \arg\min\{r_i \mid i \in M\}$.

    11.8 $K^{(j)} \leftarrow k_t^{(j)}$.
12. **Return** the new coordinates of all points (with modified $\hat{a}_t$ values) and the total cost f.

Proof. Let for $i = 1, ..., n$, $w_i = w$ and $c_i = c$. Then in iteration j, for all $i = 1, .., n$, we obtain $k_i^{(j)} = |K^{(j-1)} - 2w|$ and $r_i = |\hat{a}_i|cw$, therefore, in the three algorithms, the index t is chosen respect to $|\hat{a}_i|$ which is the same in all algorithms. Thus, the results of the three algorithms will be the same. Moreover, since in this case, in each iteration we move the point with minimum distance to the line L, so the final obtained solution is optimal. □

Using Theorem 2 and the preceding discussion on time complexity of the presented algorithms the following theorem can be stated.

Theorem 3. In the case that all the weights and the cost of changing the coordinates of the points are the same, the inverse balanced facility location problem with variable coordinates and Euclidean norm can be solved in $O(n^2)$ time.

To illustrate the presented algorithms consider the following example.

Example 1. Consider a problem with 14 points whose coordinates, weights, and coordinate modification costs are given in Table 2. Let $m_1 = A_8$ and $m_2 = A_{14}$. The midpoint of m_1 and m_2 is $O_1 = (2, 2)$ and perpendicular bisector of the line connecting m_1 and m_2 is parallel to y-axis. Thus, no rotation is needed. We just need to move all a_i, for $i = 1, \ldots, 14$. The new values are shown in Table 3.

Table 2: Coordinates, weights, and coordinate modification costs of points.

i	A_i	w_i	c_i
1	$(3, 5)$	4	5
2	$(5, 4)$	3	2
3	$(6, 4)$	2	2
4	$(3, 3)$	2	4
5	$(4, 2)$	4	5
6	$(3, -2)$	3	3
7	$(4, -1)$	2	2
8	$(5, 2)$	2	1
9	$(-2, -1)$	2	4
10	$(1, 1)$	2	5
11	$(0, 2)$	1	1
12	$(1, 4)$	1	4
13	$(-1, 3)$	2	2
14	$(-1, 2)$	2	3

Table 3: The new x-values of points.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}
1	3	4	1	2	1	2	3	-4	-1	-2	-1	-3	-3

Observe that $N_1 = \{\hat{A}_1, \ldots, \hat{A}_7\}$, $N_2 = \{\hat{A}_9, ..., \hat{A}_{13}\}$, $W(\hat{N}_1) = 22$ and $W(\hat{N}_2) = 10$. Thus, $K^{(0)} = 12$ and $k_1^{(1)} = 4,\ k_2^{(1)} = 6, k_3^{(1)} = 8, k_4^{(1)} = 8, k_5^{(1)} = 4, k_6^{(1)} = 6, k_7^{(1)} = 8$.

Using the first strategy, since $min_{i=1,\ldots,7} k_i^{(1)} = 4$ happens for $i = 1, 5$, we choose from these points, the point with minimum transfer cost, i.e. $min_{i=1,5} r_i = min\{20, 40\} = 20$. Thus the point $\hat{A}_1$ is chosen and $t = 1$. Then we set $\hat{a}_1 = 0$, $f = 20$, $K^{(1)} = 4$ and $\hat{N}_1 = \{A_2, ..., A_6, A_7\}$, $N_2 = \{A_1, A_9, ..., A_{13}\}$. Table 4 shows the obtained results in each step of the Algorithm IFLP1. The results show that by the first strategy the points $\hat{A}_1$ and $\hat{A}_4$ should be moved to the y-axis with transporting cost $f = 28$.

Table 4: The running results of Algorithm IFLP1.

j	$\{i \mid \hat{A}_i \in \hat{N}_1\}$	$K^{(j)}$	$k_1^{(j)}$	$k_2^{(j)}$	$k_3^{(j)}$	$k_4^{(j)}$	$k_5^{(j)}$	$k_6^{(j)}$	$k_7^{(j)}$	t	r_t	f
1	$\{1, 2, \ldots, 7\}$	12	4	6	8	8	4	6	8	1	20	20
2	$\{2, \ldots, 7\}$	4	–	2	0	0	4	2	0	4	8	28
3	$\{2, 3, 5, 6, 7\}$	0	–	–	–	–	–	–	–	–	–	28

Tables 5 and 6 contain the obtained results of Algorithms IFLP2 and IFLP3 in each step, respectively. Note that in the 4th iteration of IFLP2, we obtain $W(\hat{N}_1) = 15 < W(\hat{N}_2) = 17$, thus we should exchange the set $\hat{N}_1$ and $\hat{N}_2$, (see Step 2 of the algorithm). The results show that for this example the third strategy finds a better solution than the first, and terminates in fewer iterations than the second strategy.

Table 5: The running results of Algorithm IFLP2.

j	$\{i \mid \hat{A}_i \in \hat{N}_1\}$	$K^{(j)}$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	t	r_t	f
1	$\{1, 2, \ldots, 7\}$	12	20	18	16	8	40	9	8	4	8	8
2	$\{1, 2, 3, 5, 6, 7\}$	8	20	18	16	–	40	9	8	7	8	16
3	$\{1, 2, 3, 5, 6\}$	4	20	18	16	–	40	9	–	6	9	25
4	$\{1, 2, 3, 5\}$	2	20	18	16	–	40	–	–	11	2	27
5	$\{4, 6, 7, 8, 9, 10, 12, 13, 14\}$	0	20	18	16	–	40	–	–	–	–	27

Table 6: The running results of Algorithm IFLP3.

j	$\{i \mid \hat{A}_i \in \hat{N}_1\}$	$K^{(j)}$	$r_1k_1^{(j)}$	$r_2k_2^{(j)}$	$r_3k_3^{(j)}$	$r_4k_4^{(j)}$	$r_5k_5^{(j)}$	$r_6k_6^{(j)}$	$r_7k_7^{(j)}$	t	r_t	f
1	$\{1, 2, \ldots, 7\}$	12	80	108	128	64	160	54	64	6	9	9
2	$\{1, \ldots, 5, 7\}$	6	40	0	32	16	80	–	16	2	18	27
3	$\{1, 3, 4, 5, 7\}$	0	–	–	–	–	–	–	–	–	–	27

5 Computational Results

We tested the presented algorithms by two kinds of test problems. The first type contains four instances whose coordinates can be found in Beasley [2]. The second type is some randomly generated instances. The coordinate ranges of existing points are given in Table 7. The algorithms have been implemented in MATLAB (R2016). All the experiments were run on a PC with Intel Core i5 processor, 4 GB of RAM and CPU 2.5 GHz.

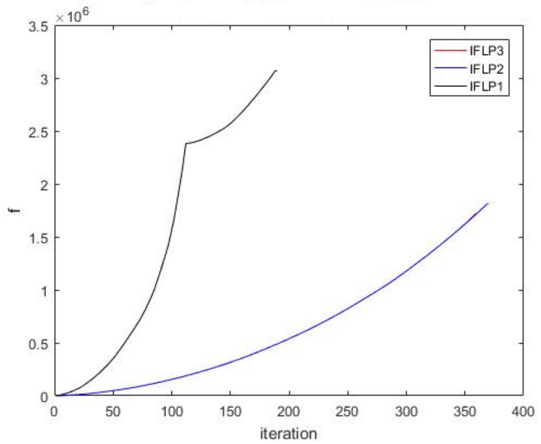
Table 7: The coordinate ranges of existing points of instances.

Instance	n	Coordinate ranges
Ruspini	75	$(4, 4)$ to $(117, 156)$
Bongartz	287	$(5, 5)$ to $(48, 48)$
TSPLIB	654	$(1000, 1000)$ to $(5000, 5000)$
Tailard	2863	$(5000, 700)$ to $(8000, 3000)$
Test1	101	$(1, 1)$ to $(5, 5)$
Test2	501	$(1, 1)$ to $(100, 100)$
Test3	700	$(1, 1)$ to $(500, 500)$
Test4	800	$(1, 1)$ to $(10, 10)$

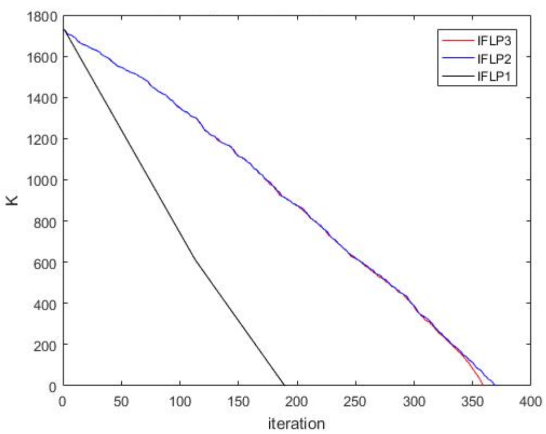
Table 8 reports the performance of the presented algorithms when point weights and modification costs are drawn randomly from a uniform distribution on [1, 5]. In these tables, W_i, for $i = 1, 2$, indicate the weight of N_i and the column with the heading (i_{m_1}, i_{m_2}) indicates the indices of facility pairs. The column headed $iter$. indicates the number of iterations until termination.

We also examined the algorithms for the case that all the weights and cost of coordinate modification of points in the instances are equal to one. In this case, the three algorithms obtained the same solutions. The results are reported in Table 9.

Figure 3 shows diagrams of the results that obtained by the three algorithms for the case that weights and cost of coordinate modification of points are generated randomly for TSPLIB instance. The diagram indicates that the behavior of IFLP2 and IFLP3 are very near to each other.



(a)



(b)

Figure 3: The results obtained by the three given algorithms for TSPLIB instance with randomly generated weights and costs of coordinate modification. (a) The change of transfer cost. (b) The change of unbalancing number.

Table 8: The results for the instances with randomly generated weights and costs of coordinate modification

			IFLP1			IFLP2			IFLP3		
Instance	(i_{m_1}, i_{m_2})	(W_1, W_2)	iter.	K	f	iter.	K	f	iter.	K	f
Ruspini	$(70, 17)$	(212,5)	25	1	18196.8	43	5	4794.4	43	3	5071.4
	$(19, 15)$	(146,71)	10	1	3376.2	17	1	790.0	15	1	815.1
	$(40, 8)$	(106,111)	3	1	170.6	3	1	54.6	3	1	54.6
	$(39, 75)$	(113,104)	3	1	35.8	5	1	40.9	5	1	46.9
Bongartz	$(280, 124)$	(168,6152)	12	0	7849.6	230	286	13375.6	219	0	11357.7
	$(12, 98)$	(6294,26)	18	0	326016.0	253	186	281389.9	260	0	276254.8
	$(55, 112)$	(5730,590)	10	0	11775.9	168	116	13929.8	158	36	11735.1
	$(49, 221)$	(1744,4576)	5	0	54281.0	159	110	43687.5	152	0	50383.8
TSPLIB	$(85, 636)$	(78,1810)	190	0	3066957.0	370	4	1813473.5	360	0	1717553.1
	$(13, 157)$	(8,1880)	206	0	9997564.2	408	6	7227146.8	398	0	6895578.7
	$(354, 60)$	(1411,477)	97	0	1086953.8	204	4	229604.1	194	0	204264.9
	$(536, 295)$	(960,928)	5	0	65.9	9	2	120.9	7	0	4669.7
Tailard	$(99, 929)$	(138,8415)	899	1	45747044.3	1731	1	34662189.0	1720	3	34255916.1
	$(2430, 1478)$	(2103,6450)	438	1	18321844.3	990	3	10216114.8	983	1	10100273.2
Test1	$(63, 15)$	(504,12)	29	0	8946.3	59	6	4745.9	57	0	4386.2
	$(8, 59)$	(102,414)	18	0	1653.6	41	0	637.4	38	0	664.3
	$(39, 30)$	(215,301)	6	0	183.1	17	16	162.9	12	0	132.2
	$(42, 28)$	(228,288)	4	0	108.6	11	0	73.3	8	0	56.8
Test2	$(320, 111)$	(286,12457)	149	1	59042.4	299	65	2960999.9	292	11	2822267.2
	$(493, 487)$	(3091,9652)	77	1	1133492.0	161	17	184470.8	157	3	175846.9
	$(128, 324)$	(8611,4132)	52	1	731431.8	120	35	129296.9	113	3	114698.2
	$(296, 74)$	(6435,6308)	4	1	1083.3	8	25	409.9	5	1	737.4
Test3	$(203, 470)$	(50212,1897)	193	31	144966485.0	416	79	60746073.0	411	97	59104467.0
	$(687, 358)$	(46384,5725)	159	3	102779280.1	357	137	35826843.9	349	3	34012343.3
	$(347, 519)$	(32273,19836)	46	1	17723034.1	128	77	2055326.6	117	7	1742600.7
	$(102, 512)$	(25116,26993)	9	1	3628271.7	32	25	320718.3	23	3	349676.4
Test4	$(73, 368)$	(10137,51)	231	0	603002.3	491	36	240184.7	483	0	230871.6
	$(256, 146)$	(7916,2272)	123	0	195519.4	301	6	52122.4	291	0	48886.1
	$(362, 425)$	(6046,4142)	40	0	86404.1	112	0	6314.9	106	0	5698.2
	$(241, 593)$	(4970,5218)	6	0	2742.1	17	2	307.2	16	0	557.9
Total				46	350562811.3		1252	157035399.8		174	152147865.5

The results in Table 9 (for uniformly weighted points with identical modification costs) validate our theoretical findings in Theorem 2. The average CPU time of three algorithms are shown in Table 10.

To compare the results obtained by the three presented algorithms, the total values of unbalancing number, transfer cost and CPU time are computed. The results show the total CPU time and unbalancing number obtained by IFLP1 is less than two other algorithms, whereas the total transfer cost obtained by IFLP3 is better than the other two algorithms. Furthermore, Table 8 shows that the hybrid strategy (IFLP3) obtains lower transfer costs than the other two strategies in most instances. Additionally, the unbalancing number obtained by this strategy in most cases are close to IFLP1 but its transfer cost is very lower than IFLP1. These results indicate that IFLP3 is more suitable and recommended for practical applications.

Sensitivity Analysis

To examine the effect of the given facilities (i_{m_1}, i_{m_2}) on the number of iterations, the unbalancing number, and the transfer cost, we ran Algorithms IFLP1–IFLP3 for varying (i_{m_1}, i_{m_2}). Obviously, if W_1 is very close to W_2 then the algorithms terminate in low iterations. Thus, the transfer cost also will be small. On the other hand, when W_1 is far from W_2 the termination iteration and transfer cost will be high. The results reported in Tables 5 and 6 confirm these observations. However, the results show that there is no meaningful relation between choosing given facilities (i_{m_1}, i_{m_2}) and the unbalancing number.

Table 9: The results for the instances with the same weights of points and costs of coordinate modification

Instance	n	(i_{m_1}, i_{m_2})	(W_1, W_2)	iter.	K	f
Ruspini	75	$(65, 11)$	(70,5)	34	1	1460.6
		(71,68)	(54,21)	18	1	800.8
		(70,20)	(47,28)	11	1	74.7
		(26,17)	(40,35)	4	1	20.9
Bongartz		(130,124)	(2,285)	143	1	3135.7
		(200,261)	(281,6)	139	1	5550.8
		(105,61)	(16,271)	129	1	4589.5
		(12,59)	(62,225)	83	1	2284.7
TSPLIB	654	$(501, 177)$	(52,602)	276	0	184476.7
		(638,189)	(513,141)	187	0	131393.8
		(620,589)	(285,369)	43	0	3803.0
		(300,600)	(314,340)	14	0	363.8
Tailard	2863	$(1341, 269)$	(2854,9)	1424	1	5805340.4
		(2057,1033)	(678,2185)	755	1	291908.4
Test1	101	$(58, 70)$	(85,16)	36	1	156.6
		$(59, 8)$	(80,21)	31	1	107.1
		$(22, 68)$	(32,69)	20	1	35.1
		$(12, 65)$	(50,51)	1	1	0.0
Test2	501	$(321, 149)$	(497,4)	248	1	93703.5
		$(126, 14)$	(4,497)	248	1	94798.2
		$(89, 400)$	(468,33)	219	1	22501.1
		$(22, 487)$	(266,235)	17	1	11.8
Test3	700	$(626, 184)$	(4,696)	347	0	290793.6
		$(520, 518)$	(200,500)	151	0	5330.3
		$(431, 58)$	(267,433)	84	0	13187.7
		$(548, 324)$	(291,409)	60	0	12640.9
Test4	800	$(287, 89)$	(796,4)	397	0	42842.5
		$(307, 783)$	(584,216)	185	0	2105.5
		$(631, 258)$	(255,545)	146	0	1658.3
		$(512, 763)$	(438,362)	39	0	109.7

Table 10: The CPU time (in seconds) of running algorithms on the instances.

Instance	n	IFLP1	IFLP2	IFLP3
Ruspini	75	0.0569	0.0651	0.1703
Bongartz	287	3.3094	4.0972	5.2912
TSPLIB	654	10.2061	14.9453	24.5801
Tailard	2863	394.9931	511.6728	639.0988
Test1	101	0.0695	0.1493	0.2660
Test2	501	2.1998	3.9191	6.5994
Test3	700	10.8089	17.2366	22.9224
Test4	800	9.8151	14.8568	20.9958
Total		**431.4588**	**566.9422**	**719.9240**

6 Summary and Conclusion

This paper introduced the inverse balanced two-facility location problem in the Euclidean plane and developed both exact and heuristic solution approaches for two problem variants. For the weight-modification variant, we established that the planar formulation is equivalent to its network counterpart, yielding an $O(n \log n)$-time solution under any L_p norm. For the coordinate-modification variant, we proved a dimensional reduction theorem—exploiting the isometric property of Euclidean rotations—that collapses the planar problem into a one-dimensional counterpart without loss of generality. Three greedy algorithms (IFLP1–IFLP3) were subsequently proposed, targeting the unbalancing number, transfer cost, and a hybrid criterion, respectively; all three are proven optimal in $O(n^2)$ time under uniform weights and identical modification costs. Computational experiments confirmed that IFLP1 is preferable when runtime and balance quality are the primary concerns, whereas IFLP3 is recommended when minimizing total transfer cost is the priority.

Several directions for future research are worth pursuing:

- *Extension to p facilities.* Generalizing the model to $p \geq 3$ facilities—where the objective is to minimize the maximum or the total variance of demand loads through weight or coordinate modification—is a natural and practically significant direction.
- *Robust and stochastic extensions.* Incorporating uncertainty in client weights or locations—via robust optimization or stochastic programming frameworks—would enhance the applicability of the model to real-world settings such as logistics network design and load balancing in communication systems.
- *Multi-criteria optimization.* A formal bi-objective treatment of the transfer cost and unbalancing number, yielding a Pareto frontier rather than a single compromise solution, represents a promising avenue for both theoretical and applied investigation.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author Contributions

Conceptualization and Methodology: **Sina Nemati, Jafar Fathali** & **Abolfazl Pouredi**; Formal Analysis: **Jafar Fathali** & **Abolfazl Pouredi**; Investigation and Data Curation: **Sina Nemati**; Writing Original Draft: **Jafar Fathali**; Writing Review & Editing: All authors; Supervision: Corresponding Author.

Artificial Intelligence Statement

Artificial intelligence (AI) tools, including large language models, were used solely for language editing and improving readability. AI tools were not used for generating ideas, performing analyses, interpreting results, or writing the scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

[1] Baroughi-Bonab F., Burkard R.E., Alizadeh B. (2010). "Inverse median location problems with variable coordinates". *Central European Journal of Operations Research*, 18, 365–381. https://doi.org/10.1007/s10100-009-0114-2

[2] Beasley J.E. (1990). "OR-Library: Distributing test problems by electronic mail". *Journal of the Operational Research Society*, 41, 1069–1072. https://doi.org/10.1057/jors.1990.166

[3] Berman O., Drezner Z., Tamir A., Wesolowsky G.O. (2009). "Optimal location with equitable loads". *Annals of Operations Research*, 167, 307–325. https://doi.org/10.1007/s10479-008-0339-9

[4] Burkard R.E., Galavii M., Gassner E. (2010). "The inverse Fermat-Weber problem". *European Journal of Operational Research*, 206, 11–17. https://doi.org/10.1016/j.ejor.2010.01.046

[5] Barbati M., Piccolo C. (2015). "Equality measures properties for location problems". *Optimization Letters*, 10, 903–920. https://doi.org/10.1007/s11590-015-0968-2

[6] Burkard R.E., Pleschiutschnig C., Zhang J.Z. (2004). "Inverse median problems". *Discrete Optimization*, 1, 23–39. https://doi.org/10.1016/j.disopt.2004.03.003

[7] Burkard R.E., Pleschiutschnig C., Zhang J.Z. (2008). "The inverse 1-median problem on a cycle". *Discrete Optimization*, 5, 242–253. https://doi.org/10.1016/j.disopt.2006.11.008

[8] Eiselt H.A., Laporte G. (1995). "Objectives in location problems". In: Drezner Z. (Ed.), *Facility Location: A Survey of Applications and Methods*. Springer, Berlin, 151–180. https://doi.org/10.1007/978-1-4612-5355-6_9

[9] Fathali J. (2022). “A row generation method for the inverse continuous facility location problems”. *Computers & Industrial Engineering*, 171, 108482. https://doi.org/10.1016/j.cie.2022.108482

[10] Fathali J., Zaferanieh M. (2023). “The balanced 2-median and 2-maxian problems on a tree”. *Journal of Combinatorial Optimization*, 45, 69. https://doi.org/10.1007/s10878-023-00997-9

[11] Galavii M. (2010). “The inverse 1-median problem on a tree and on a path”. *Electronic Notes in Discrete Mathematics*, 36, 1241–1248. https://doi.org/10.1016/j.endm.2010.05.157

[12] Gavalec M., Hudec O. (1995). “Balanced location on a graph”. *Optimization*, 35, 367–372. https://doi.org/10.1080/02331939508844156

[13] Gholami M., Fathali J. (2021). “Mathematical models for the variable weights version of the inverse minimax circle location problem”. *Journal of Mathematical Modeling*, 9, 137–144. https://doi.org/10.22124/jmm.2020.16786.1455

[14] Gholami M., Fathali J. (2022). “The inverse minisum circle location problem”. *Yugoslav Journal of Operations Research*, 32, 153–165. https://doi.org/10.2298/YJOR200715027G

[15] Gholami M., Fathali J. (2022). “Inverse minimax circle location problem with variable coordinates”. *AUT Journal of Mathematics and Computing*, 3, 137–146. https://doi.org/10.22060/ajmc.2022.20756.1075

[16] Golpayegani M., Fathali J. (2026). “Greedy algorithms for the inverse center line location problem”. *Expert Systems with Applications*, 296, 129064. https://doi.org/10.1016/j.eswa.2025.129064

[17] Guan X.C., Zhang B.W. (2012). “Inverse 1-median problem on trees under weighted Hamming distance”. *Journal of Global Optimization*, 54, 75–82. https://doi.org/10.1007/s10898-011-9742-x

[18] Landete M., Marin A. (2014). “Looking for edge-equitable spanning trees”. *Computers & Operations Research*, 41, 44–52. https://doi.org/10.1016/j.cor.2013.07.023

[19] Lejeune M.A., Prasad S.Y. (2013). “Effectiveness-equity models for facility location problems on tree networks”. *Networks*, 62, 243–254. https://doi.org/10.1002/net.21510

[20] Marin A. (2011). “The discrete facility location problem with balanced allocation of customers”. *European Journal of Operational Research*, 210, 27–38. https://doi.org/10.1016/j.ejor.2010.10.012

[21] Marsh M.T., Schilling D.A. (1994). “Equity measurement in facility location analysis: A review and framework”. *European Journal of Operational Research*, 74, 1–17. https://doi.org/10.1016/0377-2217(94)90200-3

[22] Nazari M., Fathali J. (2023). “Inverse and reverse 2-facility location problems with equality measures on a network”. *Iranian Journal of Mathematical Sciences and Informatics*, 18, 211–225. https://doi.org/10.52547/ijmsi.18.1.211

[23] Nguyen K.T., Hung N.T., Nguyen-Thu H., Le T.T., Pham V.H. (2019). “On some inverse 1-center location problems”. *Optimization*, 68, 999–1015. https://doi.org/10.1080/02331934.2019.1571056

[24] Omidi S., Fathali J. (2022). “Inverse single facility location problem on a tree with balancing on the distance of server to clients”. *Journal of Industrial and Management Optimization*, 18, 1247–1259. https://doi.org/10.3934/jimo.2021017

[25] Omidi S., Fathali J., Nazari M. (2020). “Inverse and reverse balanced facility location problems with variable edge lengths on trees”. *OPSEARCH*, 57, 261–273. https://doi.org/10.1007/s12597-019-00428-6

[26] Sayar T., Fathali J., Ghiyasi M. (2024). "The inverse 1-median problem on a tree with transferring the weight of vertices". *Transactions on Combinatorics*, 13, 335–350. https://doi.org/10.22108/toc.2023.136964.2047

[27] Tour-Savadkoohi N., Fathali J. (2025). "Inverse single facility location problem in the plane with variable coordinates". *arXiv:2503.07016* [math.OC]. https://doi.org/10.48550/arXiv.2503.07016

[28] Wu L., Lee J., Zhang J., Wang Q. (2013). "The inverse 1-median problem on tree networks with variable real edge lengths". *Mathematical Problems in Engineering*, Article ID 313868. https://doi.org/10.1155/2013/313868

Authors Bio-sketches

Sina Nemati is a PhD student since 2020 in the field of operations research at the Faculty of Mathematical Sciences in Shahrood University of Technology, Iran. He received his M.Sc. degree in 2013 on operations research from Damghan University, Iran. During his PhD he is working on location problems.

Jafar Fathali is a professor of operations research at the Faculty of Mathematical Sciences in Shahrood University of Technology, Shahrood, Iran. He received his PhD degree in 2005 on operations research from Ferdowsi University of Mashad, Iran. He is a life member of the Iranian Operations Research Society, Iranian Mathematical Society, and Iranian Statistical Society. From 2019 to 2022 he was a board member of directors of Iranian Operations Research Society. His research interests include location theory, scheduling problem, and discrete optimization. Corresponding author: Email: fathali@shahroodut.ac.ir

Abolfazl Poureidi is an associate professor of computer science at the Faculty of Mathematical Sciences in Shahrood University of Technology, Shahrood, Iran. He received his PhD degree in 2017 on computer science from Yazd University. His research interests include combinatorial optimization, applied mathematics, and computational methods.