FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 32, No 4 (2017), 447–468
https://doi.org/10.22190/FUMI1704447M

A MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR GENERAL INVERSE ORDERED p-MEDIAN LOCATION PROBLEM ON NETWORKS

Iden Mirzapolis Adeh, Fahimeh Baroughi and Behrooz Alizadeh

Abstract. This paper is concerned with a general inverse ordered p-median location problem on network where the task is to change (increase or decrease) the edge lengths and vertex weights at minimum cost subject to given modification bounds such that a given set of p vertices becomes an optimal solution of the location problem, i.e., an ordered p-median under the new edge lengths and vertex weights. A modified particle swarm optimization algorithm is designed to solve the problem under the cost functions related to the sum-type Hamming, bottleneck-type Hamming distances and the rectilinear and Chebyshev norms. By computational experiments, the high efficiency of the proposed algorithm is illustrated.

Keywords: Location problem; Inverse optimization; Ordered p-median; Particle swarm optimization algorithm.

1. Introduction

Location problems play an important role in *Operations Research* with numerous potential applications. In a classical problem, the goal is to find the best locations for establishing the facilities in order to serve the existing customers on the underlying system. Locations of fire stations, hospitals, post offices, shopping centers, schools and etc. are simple examples for applications of classical location problems. The median location problem is one of well-known models in location theory, in which the aim is to determine the best facility locations such that sum of distances between the customers and the closest facility becomes minimum. The center location problem is also another well-known model in location theory in which the task is to find the best places for establishing the facilities on the underlying system so that maximum of distances from the customers and the closest facility is minimized. For a survey on the classical location problems the reader is referred to [16, 23, 24].

Received August 06, 2016; accepted Jun 13, 2017
2010 *Mathematics Subject Classification.* Primary 90B85; Secondary 90B80, 90C27

In contrast to the location models, the inverse location problem is concerned with modifying values of some input parameters of a given location model at minimum total cost within certain modification bounds such that the given locations becomes optimal. In case of inverse location problems on network, the vertex weights and the edge lengths can be changed. In 2004, Burkard et al. [8] investigated the inverse 1-median problem with variable vertex weights and proved that the problem is solvable by a greedy-type algorithm in $O(n \log n)$ time if the underlying network is a tree or the location problem is defined in the plane where distances are measured by the rectilinear or Chebyshev norms. In 2008, the same authors presented an $O(n^2)$ time method for solving the median problem with variable vertex weights on cycles [9]. The inverse 1-maxian problem (i.e., the maximization variant of the 1-median problem) with edge length modification on graphs was investigated by Gassner and the $\mathcal{NP}$-hardness of the problem even on series-parallel graphs was proved. However, for trees a linear time algorithm was suggested [17]. Baroughi et al. [7] considered the inverse p-median location problem on graphs with edge length modifications and proved that the problem is $\mathcal{NP}$-hard. Sepasian and Rahbarnia [31] investigated the inverse 1-median problem on trees with vertex weight and edge length modifications and designed an $O(n \log n)$ time algorithm for solving this problem in case of symmetric bounds on the vertex weights.

The inverse convex ordered median problem on trees investigated by Gassner [18] and for the special case of the inverse unit weight k-centrum problem an $O(n^3k^2)$ time algorithm was developed. Recently, the inverse convex ordered 1-median problem on unweighted trees under the cost functions related to the Chebyshev norm and Hamming distance has been investigated by Nguyena and Chassein [26] and $O(n^2 \log n)$ methods have been introduced for deriving the optimal solutions. They also proved that the problem under weighted sum Hamming distance is $\mathcal{NP}$-hard.

In 1999, Cai et al. [10] showed that the inverse 1-center location problem with vertex weight modifications on directed graphs is $\mathcal{NP}$-hard. Later, Yang and Zhang [34] presented an $O(n^2 \log n)$ algorithm for the inverse vertex center problem on an unweighted tree provided that the modified edge lengths always remain positive. Using a sequence of self-defined AVL-search trees, Alizadeh et al. [4] suggested an exact combinatorial algorithm with time complexity of $O(n \log n)$ for the inverse 1-center location problem with edge length augmentation on tree networks. After that Alizadeh and Burkard developed a combinatorial $O(n^2)$ algorithm for the inverse absolute 1-center location problem in which no topology change occurs on the given tree [3]. Moreover, a linear time method for the inverse obnoxious center location problem on general graphs presented by the same authors [5]. Recently, Nguyen and Sepasian [27] developed efficient algorithms to solve the inverse 1-center problem on trees under Chebyshev norm and Hamming distance in $O(n \log n)$ time, if no topology change occurs during the modification of edge lengths.

In this paper, we consider the general inverse ordered p-median location problem on networks with both edge length and vertex weight modifications under the cost functions related to the sum-type Hamming, bottleneck-type Hamming distances and the rectilinear and Chebyshev norms. We formulate the problem as a nonlinear optimization model and show that it is $\mathcal{NP}$-hard under the rectilinear norm and sum Hamming distance. Hence, we propose a modified particle swarm optimization algorithm with a satisfactory convergence rate which approximate the solutions of the problem efficiently.

To the best of our knowledge, there is only one scientific paper on the implementation of metaheuristic algorithms to the inverse/reverse version of the location problem until now and only Alizadeh [3] investigated the general inverse p-median location problems on networks under different distance norms and developed a self-defined firefly algorithm for it. However, many papers can be found in the literature on the implementation of metaheuristic algorithms for the classical location models, see e.g., [2, 6, 13, 14, 15, 19, 22, 25, 28, 30, 32].

This paper is organized as follows: In the next section we state and formulate the general inverse ordered p-median location Problem under different cost functions. Section 3. provides an overview of the particle swarm optimization algorithm and then we propose a modified particle swarm optimization algorithm for solving the problem under investigation in Section 4.. The computational results are given in Section 5.

2. Problem Definition and Basic Properties

Let $\mathcal{N} = (V, E, w, \ell)$ be a network with vertex set V, $|V| = n$ and edge set E, $|E| = m$. Each vertex $v_i \in V$ associated with a weight $w_i \in \mathbb{R}$ and each edge $e \in E$ has a non-negative length $\ell_e \in \mathbb{R}_+$. Let $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n_+$ be a vector. For $X_p = \{x_k \in \mathcal{N} \mid k = 1, \ldots, p\}$ the ordered p-median function is defined as

$$M_\Lambda(X_p) := \langle \Lambda, d_\leqslant(X_p) \rangle = \sum_{i=1}^{n} \lambda_i d_{(i)}(X_p),$$

with

$$\begin{aligned} d_\leqslant(X_p) &= (w_{(1)} d(v_{(1)}, X_p), \ldots, w_{(n)} d(v_{(n)}, X_p)) \\ &= \left(d_{(1)}(X_p), \ldots, d_{(n)}(X_p)\right). \end{aligned}$$

Note that, for all $v \in V$ the distance from v to the set X_p is defined as

$$d(v, X_p) = d(X_p, v) := \min_{k=1,\ldots,p} d(v, x_k),$$

where $d(v, x)$ is the shortest distance from v to x and the operator $(.)$ is a permutation of $\{1, 2, ..., n\}$ such that

$$w_{(1)} d(v_{(1)}, X_p) \leqslant w_{(2)} d(v_{(2)}, X_p) \leqslant \cdots \leqslant w_{(n)} d(v_{(n)}, X_p).$$

A set of points X_p^* such that $M_\Lambda(X_p^*) \leq M_\Lambda(X_p)$ for all $X_p \subseteq \mathcal{N}$ is called an *ordered p-median* of the network. Notice that the classical center, median and

k-centrum problems correspond respectively, to the cases where $\Lambda = (0, \ldots, 0, 1)$, $\Lambda = (1, \ldots, 1, 1)$ and $\Lambda = (0, \ldots, 0, 1, \overset{k}{\ldots}, 1)$.

In the next lemma, the concept of convexity on trees has been used as defined in [12].

Lemma 2.1. ([20]). *Let $T = (V, E)$ be a tree network and $w_i \geq 0$, $i = 1, \ldots, n$. In the case of ordered 1-median problem, if $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ then the function $M_\Lambda(.)$ is convex on T.*

Now let us state the *general inverse ordered p-median location problem on networks* as follow:

Let $\mathcal{N} = (V, E, w, \ell)$ together with $\Lambda \in \mathbb{R}^n_+$ be an instance of the ordered p-median problem. In addition we are given a set of prespecified vertices X_p^*, a bound $u_e^+ \in \mathbb{R}_+$ for increasing and a bound $u_e^- \in \mathbb{R}_+$ for decreasing the lengths of edge $e \in E$. Also we are given a bound $u_v^+ \in \mathbb{R}_+$ for increasing and a bound $u_v^- \in \mathbb{R}_+$ for decreasing the weight of vertex $v \in V$. Then the task of the inverse ordered p-median location problem is to find the edge length and vertex weight modifications (p, q) such that

- $(p, q) \in \Delta$ with

$$\begin{aligned}\Delta = \{(p, q) \in \mathbb{R}^{2m+2n} \quad | \quad & 0 \leqslant p_e \leqslant u_e^+, 0 \leqslant q_e \leqslant u_e^-, \forall e \in E, \\ & 0 \leqslant p_v \leqslant u_v^+, 0 \leqslant q_v \leqslant u_v^-, \forall v \in V\}.\end{aligned}$$

- The set of prespecified vertices X_p^* becomes an ordered p-median with respect to the new edge lengths $\tilde{\ell}_e = \ell_e + p_e - q_e$ and vertex weights $\tilde{w}_v = w_v + p_v - q_v$.
- The cost function $F(p, q)$ is minimized.

Obviously, every optimal solution (p^*, q^*) satisfies the orthogonality condition

$$p_e^* q_e^* = 0, \quad \forall e \in E,$$

$$p_v^* q_v^* = 0, \quad \forall v \in V.$$

In other words, an optimal solution does not simultaneously consist of both increasing and decreasing modifications on any vertex weight or edge length.

Let $c_e^+, c_e^- \in \mathbb{R}_+$ are the cost coefficients for increasing and decreasing one unit length of edge $e \in E$ and $c_v^+, c_v^- \in \mathbb{R}_+$ are the costs coefficients for increasing and reducing one unit weight of vertex $v \in V$, respectively. Therefore, the objective function (cost function) under sum-type Hamming is written as

$$F(p, q) = \sum_{e \in E} (c_e^+ H(p_e, 0) + c_e^- H(q_e, 0)) + \sum_{v \in V} (c_v^+ H(p_v, 0) + c_v^- H(q_v, 0)), \tag{2.1}$$

and in the case of objective function under bottleneck-type Hamming, the cost of modifications is obtained as

$$F(p,q) = \max_{e\in E, v\in V} \left\{ c_e^+ H(p_e,0), c_e^- H(q_e,0), c_v^+ H(p_v,0), c_v^- H(q_v,0) \right\}, \tag{2.2}$$

where, $H(.,0)$ is the Hamming distance and defined by

$$H(x,0) = \begin{cases} 1 & ; x \neq 0 \\ 0 & ; x = 0. \end{cases}$$

If we use the rectilinear norm, then the objective function is written as

$$F(p,q) = \sum_{e\in E} (c_e^+ p_e + c_e^- q_e) + \sum_{v\in V} (c_v^+ p_v + c_v^- q_v), \tag{2.3}$$

and the objective function under Chebyshev norm can be written as

$$F(p,q) = \max_{e\in E, v\in V} \left\{ c_e^+ p_e, c_e^- q_e, c_v^+ p_v, c_v^- q_v \right\}. \tag{2.4}$$

Based on the above statements, the general inverse ordered p-median location problem (GIOMLP for short) can be formulated as the following model

$$\begin{array}{ll} \min & F(p,q) \\ \text{s.t.} & M_\Lambda(X_p^*) \leq M_\Lambda(X_p) \quad \forall X_p \in \mathcal{X}, \\ & (p,q) \in \Delta, \end{array}$$

where, $\mathcal{X}$ is a collection that contains each set of facility candidate points on network with cardinality of p.

Gassner in [18] proved that inverse ordered 1-median problem with variable edge lengths under the rectilinear norm is $\mathcal{NP}$-hard even for the convex case on weighted trees and for unit weights. She also in [18] showed that the inverse ordered 1-median problem with variable edge lengths under the rectilinear norm is $\mathcal{NP}$-hard even if the underlying location problem is the k-centrum problem. Thus based on the above statements, we immediately conclude the following propositions.

Proposition 2.1. *The general inverse ordered 1-median problem under the rectilinear norm is $\mathcal{NP}$-hard even for the convex case on trees and for unit vertex weights.*

Proposition 2.2. *The general inverse ordered 1-median problem under the rectilinear norm is $\mathcal{NP}$-hard even if the underlying location problem is the k-centrum problem.*

On the other hand, as it was mentioned, Neguyan and Chassein in [26] proved that the inverse convex ordered 1-median problem on unweighted trees under the weighted sum Hamming distance is $\mathcal{NP}$-hard. Therefore we immediately conclude that the problem on weighted trees is also $\mathcal{NP}$-hard. Thus we have

Proposition 2.3. *The general inverse ordered 1-median problem on trees under the weighted sum Hamming distance is $\mathcal{NP}$-hard .*

The above propositions imply that it is not possible to design exact polynomial time methods for solving the general inverse ordered 1-median location problem (also GIOMLP) on networks under the rectilinear norm and sum Hamming distance. To the best of our knowledge, neither the $\mathcal{NP}$-hardnes of the problem under investigation under the bottleneck type Hamming distance and the Chebyshev norm have been proved nor an exact/approximate solution algorithms have been developed for them up to now. Then we get a motivation in order to develop an efficient meta-heuristic algorithm for approximating the optimal solution of GIOMLP on networks with different distance norms.

3. Particle Swarm Optimization Algorithm

The particle swarm optimization is a population-based stochastic meta-heuristic algorithm developed by Eberhart and Kennedy [21] that is inspired by the social behavior of bird flocking. The algorithm is developed based on three simple rules:

1. When one bird locates a target or food (or optimal value of the objective function), it instantaneously transmits the information to all other birds.
2. All other birds gravitate to the target or food (or optimal value of the objective function), but not directly.
3. There is a component of each birds own independent thinking as well as its past memory.

3.1. Computational Implementation of Particle Swarm Optimization

Consider the following problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X \end{aligned} \tag{3.1}$$

where $X = \{x \in \mathbb{R}^n \mid x^{(l)} \leq x \leq x^{(u)}\}$. In particle swarm optimization algorithm each particle represents a candidate solution to the problem. Particles change their positions or states with time and fly around in a multidimensional search space. Each particle moves toward the optimum point based on its present velocity, its previous experience and the experience of its neighbors. A swarm of particles is defined as a set $\Pi = \{x_1, \cdots, x_N\}$, in which N is number of particles and the position and velocity vectors of the jth particle in the n-dimensional search space represented as $x_j = (x_{j1}, \cdots, x_{jn})$ and $v_j = (v_{j1}, \cdots, v_{jn})$, respectively. The new velocities and the positions of the particles in the $(i+1)$th iteration are updated according to the following two equations:

$$v_j(i+1) = v_j(i) + C_1 r_1 \left[P_{\text{best},j}(i) - x_j(i)\right] + C_2 r_2 \left[G_{\text{best}}(i) - x_j(i)\right], \tag{3.2}$$

$$x_j(i+1) = x_j(i) + v_j(i+1), \tag{3.3}$$

where, $P_{\text{best},j}(.)$ denotes the best position that the jth particle has achieved so far and in the $(i+1)$th iteration is updated as follow:

$$P_{\text{best},j}(i+1) = \begin{cases} x_j(i+1) & ; f(x_j(i+1)) \leq f(P_{\text{best},j}(i)) \\ P_{\text{best},j}(i) & ; \text{otherwise} \end{cases}. \tag{3.4}$$

$G_{\text{best}}(.)$ is the global best experience of particles and in the ith iteration is obtained as

$$G_{\text{best}}(i) = P_{\text{best},t}(i), \tag{3.5}$$

with

$$t = \arg\min\{P_{\text{best},j}(.) \ : \ j = 1, \cdots, N\}.$$

Moreover, r_1 and r_2 are uniformly distributed random numbers in the range 0 to 1 and C_1 and C_2 denote the relative importance of the memory (position) of the particle itself to the memory (position) of the swarm. The values of C_1 and C_2 are usually assumed to be 2 so that $C_1 r_1$ and $C_2 r_2$ ensure that the particles would overfly the target about half the time. If "MaxIt" denotes the maximum permissible number of iterations during the execution of the particle swarm optimization algorithm, then based on all considerations above, the particle swarm optimization algorithm (PSO) is summarized in Algorithm 3.1

Algorithm 3.1. Particle swarm optimization algorithm (PSO)

Begin

choose the parameters N, C_1, C_2 and $MaxIt$.

initialize the random feasible position of each particle and calculate the value of objective function of each particle.

the velocity of each particle is initialized to zero.

for $i = 1, ..., MaxIt$ **do**

  for $j = 1, 2, ..., N$ **do**

    compute $P_{\text{best},j}(i)$ by (3.4).

    compute $G_{\text{best}}(i)$ by (3.5).

    update the velocity of the particle by (3.2).

    update the new position $x_j(i+1)$ of the particle by (3.3) and calculate the value of objective function in this position.

    if $x_j(i+1) \notin X$ **then**

set $x_j(i+1) = P_{\text{best},j}(i)$.

end if

end for

end for

return $x^* = G_{\text{best}}(i+1)$ as the solution for optimization problem.

End

4. Modified Particle Swarm Optimization Algorithm

In this section, for improving the ability of PSO algorithm to avoid early local convergence, different strategy have been proposed. The modified particle swarm optimization algorithm (MPSO) will be applied for solving GIOMLP on networks. We propose the modified algorithm as follow:

It is found that usually the particle velocities build up too fast and the maximum of the objective function is skipped. Therefore, an inertia term θ is added to control the influence of the previous velocity value on the updated velocity, which in the ith iteration, θ is given by

$$\theta_i = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{MaxIt}\right) \times i, \tag{4.1}$$

where θ_{max} and θ_{min} are the initial and final values of the inertia weight and usually set to 0.9 and 0.4, respectively. The inertia weight θ was originally introduced by Shi and Eberhart in 1999 [33].

In addition constriction coefficient $\mathcal{K}$ proposed by Clerc and Kennedy [11] in 2002 as follow

$$\mathcal{K} = \frac{2}{\left|2 - C - \sqrt{C^2 - 4C}\right|} \quad , \qquad C = C_1 + C_2. \tag{4.2}$$

Specifically, the application of constriction coefficient allows control over the dynamical characteristics of the particle swarm, including its exploration versus exploitation propensities. In fact the implementation of properly defined constriction coefficient can prevent explosion. Further, these coefficients can induce particles to converge on local optima.

In some cases, it is observed that due to the great distance of the particles form $P_{\text{best},.}$ and G_{best}, convergence of the particles to these points or searching around $P_{\text{best},.}$ and G_{best}, is weak. Therefore, we design a novel strategy to enhance efficiency of algorithm. let $\Pi = \{\Pi_1, \ldots, \Pi_{TeamN}\}$ be a partition of particle swarm and for $k = 1, \ldots, TeamN$, Π_k be a team of particles. Moreover, particles of team Π_k, do not have access to the other teams information. In other words, let $T_{\text{best},k}(.)$

be the best position that the particles of team Π_k have achieved so far and in the ith iteration we have

$$T_{\text{best},k}(i) = P_{\text{best},t}(i), \tag{4.3}$$

with

$$t = \arg\min\{P_{\text{best},j}(i) \ : \ x_j \in \Pi_k\}.$$

Also, let us assume that G_{best} is the best experience of all particles in the end of algorithm and is defined by

$$G_{\text{best}} = T_{\text{best},k'}(i), \tag{4.4}$$

with

$$k' = \arg\min\{T_{\text{best},k}(i) \ : \ k = 1, ..., TeamN\}.$$

Now, we propose the following new formula for computing the velocity of particles:

$$v_j(i+1) = \mathcal{K}\{\theta_i v_j(i) + C_1 \mathcal{N}\left[\mu_j(i), \sigma_j(i)\right] + C_2 \mathcal{N}\left[\mu'_j(i), \sigma'_j(i)\right]\} \tag{4.5}$$

in which

$$\mu_j(i) = P_{\text{best},j}(i) - x_j(i) \quad , \quad \sigma_j(i) = \frac{|P_{\text{best},j}(i) - x_j(i)|}{i}$$

$$\mu'_j(i) = T_{\text{best},k}(i) - x_j(i) \quad , \quad \sigma'_j(i) = \frac{|T_{\text{best},k}(i) - x_j(i)|}{i}$$

and $\mathcal{N}[\mu, \sigma]$ denotes a Gaussian random variable with mean μ and standard deviation σ. Based on considerations above, the modified particle swarm optimization algorithm for solving the optimization problems, in particular for GIOMLP is summarized in Algorithm 4.1.

Algorithm 4.1. The modified particle swarm optimization algorithm (MPSO)

> **Begin**
>
> choose the parameters N, C_1, C_2, $TeamN$ and $MaxIt$.
>
> initialize the random feasible position of each particle and calculate the value of objective function of each particle.
>
> the velocity of each particle is initialized to zero.
>
> **for** $i = 1, ..., MaxIt$ **do**
>
>> **for** $k = 1, 2, ..., TeamN$ and $x_j \in \Pi_k$ **do**
>>
>> compute $P_{\text{best},j}(i)$ by (3.4).
>>
>> compute $T_{\text{best},k}(i)$ by (4.3).
>>
>> update the velocity of the particle by (4.5).
>>
>> update the new position $x_j(i+1)$ of the particle by (3.3) and calculate the value of objective function in this position.
>>
>> **if** $x_j(i+1) \notin X$ **then**

set $x_j(i+1) = P_{\text{best},j}(i)$.

end if

end for

end for

compute G_{best} by (4.4).

return $x^* = G_{\text{best}}$ as the solution for optimization problem.

End

Recall that MPSO and PSO are originally introduced for unconstrained optimization problems. Moreover, the inverse ordered p-median problem is a constrained optimization model. Therefore to achieve this aim, we use nonstationary penalty function where penalty parameter values changing dynamically with the iteration number during optimization [29]. Consider an constrained minimization problem is given by

$$\begin{aligned} \text{(4.6)} \qquad & \min \quad f(x) \\ & \text{s.t.} \quad g_j(x) \leq 0, \quad j = 1, ..., m \\ & \qquad x \in X \end{aligned}$$

where $X = \{x \in \mathbb{R}^n \mid x^{(l)} \leq x \leq x^{(u)}\}$. The equivalent unconstrained function $F(x)$ is constructed by using a penalty function for the constraints and is obtained as

$$F(x) = f(x) + \mathcal{C}(i)H(x),$$

where $\mathcal{C}(i)$ denotes a dynamically modified penalty parameter that changes with the iteration number i, that is assumed as $\mathcal{C}(i) = (ci)^{\alpha}$, and $H(x)$ represents the penalty factor associated with the constraints that is defined as

$$H(x) = \sum_{j=1}^{m} \left\{ \varphi(q_j(i)) \left[q_j(i)\right]^{\gamma(q_j(i))} \right\},$$

with

$$q_j(i) = \max\{0, g_j(x)\}, \qquad j = 1, ..., m,$$

$$\varphi(q_j(i)) = a\left(1 - \frac{1}{e^{q_j(i)}}\right) + b,$$

$$\gamma(q_j(i)) = \begin{cases} 1 & ; q_j(i) \leqslant 1 \\ 2 & ; q_j(i) > 1 \end{cases}.$$

Note that c, α, a, and b are constants. Therefore, the model (4.6) can be reconstructed as a equivalent model with new objective function and no constraints, which is expressed as

$$\begin{aligned} \text{(4.7)} \qquad & \min \quad F(x) \\ & \text{s.t.} \quad x \in X. \end{aligned}$$

However, the PSO and MPSO algorithms can be applied to model (4.7).

5. Computational Experiments

In this section, we first compare the modified particle swarm optimization algorithm (MPSO) with particle swarm optimization algorithm (PSO). A series of computational experiments are conducted in order to measure the effectiveness of the proposed algorithm MPSO. Finally, MPSO and PSO will be applied to GIOMLP models on network and the results will be compared. The algorithms are coded in MATLAB 8.1.0.604(R2013a) and run on a PC with processor Intel(R) Core(TM) i7 CPU 2.10GHZ and 6GB of RAM under windows 7.

5.1. Comparison of MPSO with PSO

We used some well-known test functions, the so called benchmark functions, to evaluate the performance of MPSO as compared to PSO algorithm. The benchmark functions that used, is shown in Table 5.1 with their names, variable limits and value of the global minimum (G. m. for short). We consider the same population size $N = 30$ and the permissible iteration number $MaxIt = 1000$ with different dimension $n = 20, 25, 30, 35$ for all algorithms in all our simulations. Note that each of the numerical experiments was repeated 25 times in order to compute the best objective function values. In MPSO, assumed that the relative importance of the memories $C_1 = C_2 = 2.05$, initial values of the inertia weight $\theta_{max} = 0.9$ and final values of the inertia weight $\theta_{min} = 0.4$. Also in PSO, assumed that $C_1 = C_2 = 2$.

Table 5.1: Benchmark functions with dimensional 'n' used for testing the performance of MPSO and PSO algorithms.

Name	Objective function	Bounds	G. m.
Sphere	$\sum_{i=1}^{n} x_i^2$	$[-100, +100]$	0
Ackley	$20 + e - 20\left(e^{-0.2\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}}\right) - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	$[-35, +35]$	0
Rosenbrock	$\sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right)$	$[-50, +50]$	0
Zacharov	$\sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	$[-5, +10]$	0

The results are presented in Table 5.2. According to the results of our experiments, we conclude that MPSO is much more efficient in obtaining the global optimal solution in comparing with PSO algorithm.

Table 5.2: The results obtained by applying MPSO and PSO algorithms to the benchmark functions of Table 5.1.

Function	Dim	PSO best	PSO mean	MPSO best	MPSO mean
Sphere	20	4.3693e-34	6.7051e-28	1.0881e-58	5.0242e-53
	25	2.0872e-23	1.2696e-14	8.8548e-40	2.2157e-31
	30	6.7193e-17	5.3194e-11	1.9123e-27	8.8264e-21
	35	1.0513e-11	4.8426e-06	2.0647e-19	3.1435e-11
Ackley	20	1.8212e-07	6.4835e-06	6.2172e-15	2.6429e-12
	25	5.9682e-05	6.5299e-04	1.7983e-11	1.9955e-10
	30	1.0212e-03	7.7100e-02	2.0819e-08	3.8244e-06
	35	4.5860e-03	4.6431e-01	8.1800e-05	1.2000e-03
Zacharov	20	4.6763e+01	1.8413e+02	1.7490e-06	8.3120e+00
	25	9.0499e+01	2.8004e+02	2.7000e-03	1.3068e+01
	30	2.4834e+02	3.9350e+02	3.8027e+00	4.9468e+01
	35	3.2563e+02	4.9816e+02	3.6126e+01	1.1835e+02
Rosenbrock	20	5.2131e+00	5.0978e+01	1.2080e-01	8.7419e+00
	25	1.7359e+00	7.2851e+01	7.7580e-01	1.2328e+01
	30	1.6682e+01	1.3993e+02	1.5130e-01	1.7710e+01
	35	9.8392e+01	2.8410e+02	8.7587e+00	2.5213e+01

5.2. The Execution of PSO and MPSO on GIOMLP Models

In this section, we present the application of PSO and MPSO algorithms for solving GIOMLP on networks under four cost functions given in (2.1), (2.2), (2.3) and (2.4). To test the performance of the method, computational experiment has been carried out on a set of randomly generated instances. We applied PSO and MPSO for different instance of GIOMLP and observed that the MPSO can efficiently solve these models with a higher convergence speed in comparison with PSO. In the following, we present the computational results of the performance of PSO and MPSO on an instance of the inverse ordered 2-median problem on the given network $\mathcal{N}$ of Figure 5.1.

The algorithms are executed for population size $N = 10, 30, 50$ with $TeamN = 2, 5, 10$ and $MaxIt = 300$. The input data of network $\mathcal{N}$ are given in the Table 5.3.

Recall that the aim is to modify the vertex weights w_v and edge lengths ℓ_e at minimum total cost with respect to modification bounds until x_1 and x_2 become an ordered 2-median of network $\mathcal{N}$ under the new vertex weights and edge lengths. It should be notified that in applying PSO and MPSO, we assume that

$$(x_1, \cdots, x_m) = (q_e)_{e\in E},$$
$$(x_{m+1}, \cdots, x_{2m}) = (p_e)_{e\in E},$$
$$(x_{2m+1}, \cdots, x_{2m+n}) = (q_v)_{v\in V},$$
$$(x_{2m+n+1}, \cdots, x_{2m+2n}) = (p_v)_{v\in V}.$$

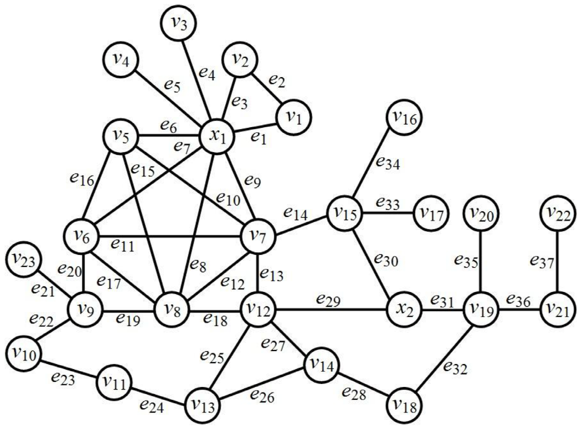

FIG. 5.1: *Network $\mathcal{N}$ and the set of prespecified vertices* $\{x_1, x_2\}$

Table 5.3: The input data for the general inverse ordered 2-median problem on network $\mathcal{N}$

$(\lambda_1, \cdots, \lambda_{25})$	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)
$\ell_1, \cdots, \ell_{37}$	7, 4, 2, 1, 5, 3, 2, 3, 4, 1, 1, 5, 3, 3, 2, 3, 4, 2, 4, 2, 3, 6, 3, 3, 1, 6, 4, 1, 5, 2, 2, 7, 2, 2, 3, 1, 4
$u^+_{e_1}, \cdots, u^+_{e_{37}}$	4, 6, 7, 1, 3, 4, 6, 8, 1, 2, 4, 6, 2, 3, 7, 1, 9, 2, 4, 3, 5, 7, 1, 1, 4, 3, 3, 5, 2, 9, 1, 2, 7, 1, 1, 2, 3
$u^-_{e_1}, \cdots, u^-_{e_{37}}$	4, 2, 1, 0.5, 4, 1, 1, 2, 1, 0.5, 0.5, 3, 2, 1, 1, 1, 1, 1, 3, 1, 2, 3, 1, 2, 0.5, 4, 2, 0.5, 2, 1, 1.5, 5, 1, 1, 1, 0.5, 3
$c^+_{e_1}, \cdots, c^+_{e_{37}}$	25, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 10, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5
$c^-_{e_1}, \cdots, c^-_{e_{37}}$	1, 2, 3, 4, 5, 6, 7, 45, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 35
$w_{x_1}, w_{x_2}, w_{v_1}, \cdots, w_{v_{23}}$	1, -1, -9, -9, -5, 2, -3, 6, 4, 2, -6, 7, 5, 3, -4, -3, -6, 1, 2, 4, -2, 4, 1, 2, 11
$u^+_{x_1}, u^+_{x_2}, u^+_{v_1}, \cdots, u^+_{v_{23}}$	5, 0.5, 4, 2, 3, 4, 2, 5, 4, 3, 4, 6, 1, 2, 2, 2, 4, 9, 8, 5, 1, 4, 8, 6, 1
$u^-_{x_1}, u^-_{x_2}, u^-_{v_1}, \cdots, u^-_{v_{23}}$	0.5, 2, 4, 3, 1, 1, 3, 5, 2, 1, 1, 5, 4, 2, 1, 1, 1, 0.5, 1, 2, 5, 3, 0.5, 1.5, 4
$c^+_{x_1}, c^+_{x_2}, c^+_{v_1}, \cdots, c^+_{v_{23}}$	1, 2, 3, 4, 5, 6, 15, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 20
$c^-_{x_1}, c^-_{x_2}, c^-_{v_1}, \cdots, c^-_{v_{23}}$	30, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 40, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6

Therefore, $x = (x_1, \cdots, x_{2m+2n})$ denotes the decision vector of GIOMLP that used in PSO and MPSO. Moreover, due to the orthogonality condition

- if $q_e > p_e$, then $q_e = q_e - p_e \quad , \quad p_e = 0$.
- if $p_e > q_e$, then $q_e = 0 \quad , \quad p_e = p_e - q_e$.

- if $q_v > p_v$, then $q_v = q_v - p_v \quad , \quad p_v = 0$.
- if $p_v > q_v$, then $q_v = 0 \quad , \quad p_v = p_v - q_v$.

The computational results are presented in Tables 5.4, 5.5, 5.6, 5.7 and the best solutions are presented in Tables 5.8, 5.9, 5.10, 5.11.

Table 5.4: The results of the performance of PSO and MPSO on the GIOMLP example under the bottleneck-type Hamming cost function

		PSO			MPSO		
N		worst	mean	best	worst	mean	best
10	f^*	45	41.5	35	45	34	20
10	CPU	34.1799 s	33.4014 s	32.9356 s	35.8091 s	33.9421 s	33.4250 s
30	f^*	45	39.5	30	40	29.5	7.5
30	CPU	100.8156 s	98.9705 s	97.5049 s	100.2953 s	99.3108 s	98.0074 s
50	f^*	45	39	30	35	26.5	15
50	CPU	183.3986 s	173.9812 s	171.0185 s	165.6220 s	164.8333 s	164.1740 s

Table 5.5: The results of the performance of PSO and MPSO on the GIOMLP example under the sum-type Hamming cost function

		PSO			MPSO		
N		worst	mean	best	worst	mean	best
10	f^*	386	332.6	290	331	286.3	237
10	CPU	33.4300 s	33.0101 s	32.8066 s	33.8385 s	33.3408 s	33.1606 s
30	f^*	348	328.25	289.5	299	276.55	250.5
30	CPU	98.7662 s	97.8620 s	97.4905 s	95.2831 s	94.3120 s	93.9202 s
50	f^*	342	313.95	274.5	278.5	258.4	239
50	CPU	172.2025 s	164.5040 s	162.7221 s	158.3399 s	156.2106 s	155.4645 s

Table 5.6: The results of the performance of PSO and MPSO on the GIOMLP example under the rectilinear norm

		PSO			MPSO		
N		worst	mean	best	worst	mean	best
10	f^*	614.8122	465.0678	385.3710	410.4581	370.6804	324.1834
10	CPU	35.4416 s	34.2960 s	32.9044 s	33.7537 s	33.3754 s	33.0090 s
30	f^*	540.8529	443.1780	355.2883	356.8444	312.4536	216.4779
30	CPU	99.7806 s	97.5805 s	97.0867 s	94.3329 s	93.2957 s	92.8414 s
50	f^*	550.0933	413.87712	340.9751	331.6956	300.74537	271.5626
50	CPU	173.2479 s	165.0412 s	161.5299 s	157.0809 s	155.7778 s	154.7668 s

Table 5.7: The results of the performance of PSO and MPSO on the GIOMLP example under the Chebyshev norm

		PSO			MPSO		
N		worst	mean	best	worst	mean	best
10	f^*	106.7008	74.4764	49.3811	66.1619	46.0691	31.0313
10	CPU	32.4447 s	31.9141 s	31.5278 s	33.9445 s	32.7519 s	31.9066 s
30	f^*	87.2799	68.8894	46.2844	65.9527	44.7584	32.3251
30	CPU	101.1869 s	98.1254 s	95.0506 s	102.8937 s	99.2875 s	98.1751 s
50	f^*	136.7286	72.4666	40.6639	44.1767	37.6766	17.8963
50	CPU	169.5950 s	164.7514 s	162.1629 s	164.3641 s	163.2033 s	162.4201 s

Table 5.8: The best modified edge lengths and vertex weights obtained by applying PSO and MPSO to GIOMLP example under the bottleneck-type Hamming cost function

	f^*	30
PSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	3.9957, 5.5349, 5.2650, 1.3228, 6.0281, 2.3940, 1.4381, 6.3247, 4.9072, 0.8736, 2.3215, 8.2978, 1.7203, 3.7792, 7.8989, 3.9749, 9.3988, 3.8180, 2.2832, 2.4835, 1.6167, 5.3129, 2.0741, 2.5595, 0.9679, 7.4839, 5.2448, 3.3956, 4.6633, 4.6957, 2.2055, 7.0396, 8.7899, 2.5501, 3.1941, 0.9452, 6.5723
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	1.1543,-0.9428,-8.9951,-7.0265,-5.1512, 2.1345, -4.8398, 3.2897, 4.4961, 1.5575,-2.7024, 6.6060, 4.8518, 2.2633,-2.6164,-3.6985,-6.5423, 9.5022, 1.3272, 6.6897,-3.8104, 6.6771, 2.9683, 4.9291, 11.8109
	f^*	7.5
MPSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	5.4131, 2.0795, 2.1704, 1.7698, 5.0260, 2.0074, 1.5192, 5.4912, 4.5594, 0.5337, 0.5245, 2.9313, 3.1280, 4.6399, 2.4606, 2.9181, 4.0192, 2.0352, 5.8188, 4.1860, 2.5253, 5.2775, 3.1067, 1.7995, 0.9048, 7.3195, 4.8745, 0.7116, 3.1557, 3.8246, 2.9545, 8.2018, 8.0591, 1.3286, 2.2632, 2.9204, 4.2338
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.5498,-0.5250,-9.4793,-7.6072,-4.8497, 1.1617, -1.3200, 7.1609, 2.6806, 1.5657,-4.0027, 9.0178, 2.4682, 4.2959,-4.3852,-3.8455,-4.3483, 1.9846, 8.7074, 3.4119,-1.0796, 3.2465, 4.3252, 7.5370, 11.3707

Table 5.9: The best modified edge lengths and vertex weights obtained by applying PSO and MPSO to GIOMLP example under the sum-type Hamming cost function

	f^*	274.5
PSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	6.9280, 2.1022, 1.7082, 0.6251, 6.5569, 4.2879, 3.9377, 3.4340, 3.9385, 2.8432, 4.3730, 10.7515, 2.7660, 5.6263, 1.3859, 2.1053, 11.1039, 1.1470, 7.3450, 1.0759, 1.7073, 3.2630, 2.0762, 2.2526, 4.9733, 6.9422, 5.9498, 4.1069, 6.7585, 5.4904, 2.8336, 8.4067, 1.0456, 1.5655, 2.8565, 0.9097, 5.7272
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.8424,-2.0572,-6.1387,-8.8908,-5.7490, 1.4860, -4.7024, 10.8361, 6.0319, 4.4278,-2.9040, 2.8756, 4.3022, 4.7467,-2.9920,-1.4949,-4.6175, 0.8142, 9.1615, 3.7006,-2.5933, 3.4823, 4.9891, 5.8350, 8.3513
	f^*	237
MPSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	3.9733, 3.7007, 7.2464, 1.7367, 4.4493, 2.3735, 1.3809, 6.4961, 3.3953, 0.8953, 1.8755, 7.9932, 3.1008, 2.7819, 1.7208, 2.4254, 3.4451, 3.0931, 6.9807, 1.5800, 2.4182, 4.5825, 2.4932, 3.4434, 1.9803, 5.6878, 4.0655, 1.1364, 3.9626, 1.4323, 1.4005, 8.8264, 7.2291, 2.7183, 3.4871, 1.6628, 5.3553
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.8573,-1.0173,-11.4074,-10.5431,-5.4514, 1.4419, -2.5064, 10.6726, 2.7701, 3.3735,-6.1303, 6.6996, 2.4295, 2.3503,-4.2470,-1.7266,-6.2679, 2.2468, 7.3646, 6.8674,-2.9097, 3.7274, 0.7271, 1.4830, 11.1069

Table 5.10: The best modified edge lengths and vertex weights obtained by applying PSO and MPSO to GIOMLP example under the rectilinear norm

	f^*	340.9751
PSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	3.0000, 7.4849, 1.0630, 1.6160, 2.0767, 4.2406, 1.1479, 3.9493, 3.0000, 0.8342, 3.7723, 5.4344, 2.9001, 2.8535, 7.2121, 3.9425, 5.9201, 2.0000, 4.2327, 2.0067, 2.2089, 4.1441, 2.0000, 3.0000, 1.6163, 4.5997, 3.8612, 1.3342, 4.8958, 5.9822, 0.8137, 6.2070, 1.2190, 1.3364, 3.0000, 0.6739, 6.9414
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	1.4727,-0.5000,-10.2515,-9.3787,-4.6767, 2.0000, -2.7817, 1.0000, 4.2457, 2.4819,-6.7155, 6.4233, 4.9308, 3.0000,-4.0000,-3.1801,-6.8133, 7.2579, 1.0129, 3.8452,-1.0000, 3.7109, 0.5339, 6.5549, 9.6836
	f^*	216.4779
MPSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	7.1610, 3.7800, 1.0321, 1.0196, 5.0696, 3.2558, 1.8835, 3.4296, 4.5828, 2.8922, 1.2972, 4.7999, 3.1403, 5.5320, 6.3361, 2.9997, 4.0461, 1.0420, 1.8962, 2.2183, 1.5776, 4.9583, 2.9287, 3.3885, 1.0659, 6.9383, 4.7625, 4.6697, 6.6221, 1.5126, 2.9436, 5.9429, 4.8476, 2.0049, 3.1426, 0.9623, 3.9039
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.9841,-0.8065,-9.3425,-8.8987,-4.5077, 2.1082, -2.8612, 7.5922, 4.3347, 2.0299,-5.5265, 6.8365, 5.1809, 3.2152,-4.0152,-2.9621,-6.3087, 1.0430, 7.2844, 7.5472,-1.9110, 3.3971, 0.9767, 1.9864, 11.1937

Table 5.11: The best modified edge lengths and vertex weights obtained by applying PSO and MPSO to GIOMLP example under the Chebyshev norm

	f^*	40.6639
PSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	6.7602, 2.4178, 8.8492, 0.9943, 3.6583, 2.1640, 1.0502, 2.6186, 4.1690, 1.0860, 2.7399, 3.7135, 3.9706, 5.4271, 1.0128, 2.9421, 3.2866, 1.5456, 3.1585, 1.3654, 1.6131, 9.3924, 3.6730, 2.2756, 0.9992, 6.0222, 3.6426, 0.8579, 4.3919, 1.6470, 2.1530, 3.2351, 1.7649, 2.6891, 2.7808, 0.8578, 4.2898
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.8285,-0.9632,-6.2013,-7.9659,-5.2530, 1.3292, -5.7107, 10.1999, 7.0155, 1.6262,-2.6145, 2.9350, 3.6095, 3.6665,-3.8071,-2.2923,-6.2158, 9.1328, 7.2305, 8.8685,-2.1512, 2.0920, 5.9968, 2.7478, 10.1456
	f^*	17.8963
MPSO	$\tilde{\ell}_{e_1}, \ldots, \tilde{\ell}_{e_{37}}$	4.9910, 3.0089, 6.4474, 0.7828, 1.4928, 3.3260, 1.0554, 3.1087, 3.1294, 1.7929, 2.5441, 7.0053, 2.8292, 5.2849, 1.1106, 2.8448, 7.3403, 1.3114, 7.8149, 1.5665, 1.1850, 8.1159, 2.0853, 3.6444, 1.9912, 3.2023, 2.5338, 3.7774, 4.8922, 1.8081, 2.8671, 3.2557, 1.5375, 2.6482, 2.3797, 2.6338, 3.9786
	$\tilde{w}_{x_1}, \tilde{w}_{x_2}, \tilde{w}_{v_1}, \ldots, \tilde{w}_{v_{23}}$	0.7532,-1.7722,-8.2090,-7.9765,-4.7299, 4.8398, -3.1691, 8.8504, 5.6717, 1.0082,-2.4207, 4.2828, 5.1187, 2.1719,-4.4680,-3.8398,-2.1350, 0.8588, 2.9644, 4.6886,-1.5514, 6.6374, 0.6397, 5.0961, 11.6101

6. Conclusions

In this paper, we investigated the general inverse ordered p-median location problem with both edge length and vertex weight modifications on networks under the cost functions related to the sum-type Hamming, bottleneck-type Hamming, rectilinear and Chebyshev distance norms. We proposed an efficient modified particle swarm optimization algorithm to approximate the optimal solutions.

REFERENCES

1. B. Alizadeh and R.E. Burkard: *Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees*, Networks, 58 (2011) 190-200.
2. J. Alcaraz, L. Mercedes and F.M. Juan: *Design and analysis of hybrid metaheuristics for the reliability p-median problem*, European Journal of Operational Research, **222** (2012), 54-64.
3. B. Alizadeh and S. Bakhteh: *A modified firefly algorithm for general inverse p-median location problems under different distance norms*, Opsearch, DOI 10.1007/s12597-016-0296-z.
4. B. Alizadeh, R.E. Burkard and U. Pferschy: *Inverse 1-center location problems with edge length augmentation on trees*, Computing, **86** (2009) 331-343.
5. B. Alizadeh and R.E. Burkard: *A linear time algorithm for inverse obnoxious center location problems on networks*, Central European Journal of Operations Research, **21** (2013) 585-594.
6. M. Bashiri and M.H. Bakhtiarifar: *Finding the optimum location in a one-median network problem with correlated demands using simulated annealing*, Scientia Iranica, **20** (2013), 793-800.
7. F. Baroughi-Bonab, R.E. Burkard and E. Gassner: *Inverse p-median problems with variable edge lengths*, Mathematical Methods of Operations Research, **73** (2011) 263-280.
8. R.E. Burkard, C. Pleschiutschnig and J. Zhang: *Inverse median problems*, Discrete Optimization, **1** (2004) 23-39.
9. R.E. Burkard, C. Pleschiutschnig and J. Zhang: *The inverse* 1*-median problem on a cycle,* Discrete Optimization, **5** (2008) 242-253.
10. M.C. Cai, X.G. Yang and J.Z. Zhang: *The complexity analysis of the inverese center location problem*, Journal of Global Optimization, **15** (1999) 213-218.
11. M. Clerc and J. Kennedy: *The particle swarm explosion, stability and convergence in a multidimensional complex space*, Transactions on Evolutionary Computation , **6** (2002) 58-73.
12. P.M. Dearing, R.L. Francis and T.J. Lowe: *Convex location problems on tree networks,* Operations Research, **24** (1976) 628-642.
13. Z. Drezner, J. Brimberg, N. Mladenović and S. Salhi: *New heuristic algorithms for solving the planar p-median problem*, Computers and Operations Research, **62** (2015), 296-304.

14. J. Fathali and H. Taghizadeh: *Solving the p-median problem with pos/neg weights by variable neighborhood search and some results for special cases*, European Journal of Operational Research, **170** (2006), 440-462.

15. J. Fathali: *A genetic algorithm for the p-median problem with pos/neg weights*, Applied Mathematics and Computation, **183** (2006), 1071-1083.

16. R.L. Francis, L.F. McGinnis and J.A. White: *Facility Layout and Location, An Analytical Approach*, Prentice Hall, Englewood Cliffs, 1992.

17. E. Gassner: *The inverse 1-maxian problem with edge length modification,* Journal of Combinatorial Optimization, **16** (2007) 50-67.

18. E. Gassner: *An inverse approach to convex ordered median problems,* Journal of Combinatorial Optimization, **23** (2012) 261-273.

19. A. Ghaderi and R. Rahmaniani: *Meta-heuristic solution approaches for robust single allocation p-hub median problem with stochastic demands and travel times*, The International Journal of Advanced Manufacturing Technology, **82** (2016), 1627-1647.

20. J. Kalcsics, S. Nickel, J. Puerto and A. Tamir: *Algorithmic results for ordered median problems,* Operations Research Letters, **30** (2002) 149-158.

21. J. Kennedy and R.C. Eberhart: *Particle swarm optimization*, Proceeding of IEEE International Conference on Neural Networks, **4** (1995) 1942-1948.

22. J. Kratica, Z. Stanimirović, D. Tošić and V. Filipović: *Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem,* European Journal of Operational Research, **182** (2007), 15-28.

23. R.F. Love, J.G. Morris and G.O. Wesolowsky: *Facilities Location: Models and Methods*, North-Holland, New York, 1988.

24. B.P. Mirchandani and R.L. Francis: *Discrete Location Theory*, John Wiley, New York, 1990.

25. N. Mladenović, J. Brimberg, P. Hansen and J. A. Moreno-Pérez: *The p-median problem: A survey of metaheuristic approaches*, European Journal of Operational Research, **179** (2007), 927-939.

26. K.T. Nguyen and A. Chassein: *The inverse convex ordered 1-median problem on trees under Chebyshev norm and Hamming distance,* European Journal of Operational Research, **247** (2015) 774-781.

27. K.T. Nguyen and A.R. Sepasian: *The inverse 1-center problem on trees with variable edge lengths under Chebyshev norm and Hamming distance*, Journal of Combinatorial Optimization, (2015) Doi: 10.1007/s10878-015-9907-5.

28. J. Puerto, D. Pérez-Brito and C. G. García-González: *A modified variable neighborhood search for the discrete ordered median problem*, European Journal of Operational Research, **234** (2014), 61-76.

29. S. S. Rao: *Engineering Optimization: Theory and Practice*, John Wiley, New York, NY, USA, 4th edition, 2009.

30. M. Sevkli, R. Mamedsaidov and F. Camci: *A novel discrete particle swarm optimization for p-median problem*, Journal of King Saud University-Engineering Sciences, **26** (2014), 11-19.

31. A.R. Sepasian and F. Rahbarnia: *An algorithm for the Inverse 1-median problem on trees with variable vertex weights and edge reductions*, Optimization: A Journal of Mathematical Programming and Operations Research, **64** (2015) 595-602.

32. Z. Stanimirović, J. Kratica and D. Dugošija: *Genetic algorithms for solving the discrete ordered median problem*, European Journal of Operational Research, **182** (2007), 983-1001.

33. Y. Shi and R.C. Eberhart: *Parameter selection in particle swarm optimization*, Proceedings of Evolutionary Programming, **7** (1999) 591-600.

34. X. Yang and J. Zhang: *Inverse center location problem on a tree*, Journal of Systems Science and Complexity, **21** (2008) 651-664.

Iden Mirzapolis Adeh
Faculty of Basic Sciences
Department of Applied Mathematics
Sahand University of Technology
Sahand New Town, Tabriz, Iran
iden_mirzapolis@yahoo.com

Fahimeh Baroughi
Faculty of Basic Sciences
Department of Applied Mathematics
Sahand University of Technology
Sahand New Town, Tabriz, Iran
baroughi@sut.ac.ir
(Corresponding author)

Behrooz Alizadeh
Faculty of Basic Sciences
Department of Applied Mathematics
Sahand University of Technology
Sahand New Town, Tabriz, Iran
alizadeh@sut.ac.ir