https://toc.ui.ac.ir

Transactions on Combinatorics
ISSN (print): 2251-8657, ISSN (on-line): 2251-8665
Vol. 13 No. 4 (2024), pp. 335-350.

www.ui.ac.ir

THE INVERSE 1-MEDIAN PROBLEM ON A TREE WITH TRANSFERRING THE WEIGHT OF VERTICES

TAHERE SAYAR, JAFAR FATHALI* AND MOJTABA GHIYASI

Abstract. In this paper, we investigate a case of the inverse 1-median problem on a tree by transferring the weights of vertices which has not been raised so far. This problem considers modifying the weights of vertices via transferring weights of the vertices with the minimum cost such that a given vertex of the tree becomes the 1-median with respect to the new weights. A linear programming model is proposed for this problem. The applicability and efficiency of the presented model are shown in numerical examples and a real-life problem dealing with transferring users in a social network.

1. Introduction

Location theory is one of the most commonly used topic in the operations research. In most of the location problems, we find the optimal location of one or more servers based on the factors and variables that effect transportation and establishing costs. Location problem was introduced by Weber in 1909 [20]. However, serious studies on location theory began in 1964, when Hakimi [12] proposed minisum and minimax objective functions for location models. Hakimi introduced the p-median and p-center problems on networks. The p-median problem is one of the most important classic facility location models. This problem tries to find the location of p facilities such that the sum of weighted distances from clients to the closest facility is minimized. If the objective function

Keywords: median problem; inverse optimization; location models.

MSC(2010): Primary: 90B80; Secondary: 90B06.

Communicated by Alireza Abdollahi.

Article Type: Research Paper.

*Corresponding author.

Received: 25 February 2023, Accepted: 24 August 2023.

Cite this article: T. Sayar, J. Fathali and M. Ghiyasi, The inverse 1-median problem on a tree with transferring the weight of vertices, Trans. Comb., **13** no. 4 (2024) 335–350. http://dx.doi.org/10.22108/toc.2023.136964.2047 .

is replaced by minimizing the maximum weighted distance from clients to the closest facility, then the problem is called p-center. For more details in facility location problems, we refer to the books of Daskin [6] and Eiselt and Mariannov [7].

The classical location models deal with finding the optimal locations of the facilities. However, in some cases the facilities may already exist and the problem is to improve the given locations by changing some parameters. If we want to change the parameters with minimum cost such that the given locations become optimal, then the problem is called the inverse location problem. Many authors have considered the inverse location models. Cai et al. [5] showed that the inverse center problem is NP-hard. Burkard et al. [3] investigated the inverse p-median problem and presented an $O(nlogn)$ algorithm for the inverse 1-median problem on a tree and in the plane. Galavii [9] improved the time complexity of the inverse 1-median problem on a tree to linear time. Sepasian and Rahbarnia [19] proposed an $O(nlogn)$ algorithm for the inverse 1-median problem on trees with variable vertex weights and edge reductions. The inverse 1-median problem on trees under weighted Hamming distance is considered by Guan and Zhang [11]. Burkard et al. [4] developed an $O(n^2)$ algorithm for the inverse 1-median problem on a cycle. The inverse 1-median problem on block graphs with variable vertex weights has been investigated by Nguyen [14]. Alizadeh et al. [1] considered the inverse 1-center location problem with edge length augmentation on trees and presented an $O(nlogn)$ time algorithm. Later, Alizadeh and Burkard [2] proved that the inverse 1-center problem can be solved in $O(n^2)$ time. The inverse 1-center problem on trees under Chebyshev norm and Hamming distance has been considered by Nguyen and Sepasian [16]. Nguyen et al. [15] developed an $O(dn^2logn)$ algorithm for the inverse 1-center problem on R^d. Nazari and Fathali [13] considered the inverse of balanced 2-facility location problem with variable vertex weights on a network. Recentely, Omidi et al. [18] and Omidi and Fathali [17] investigated the inverse of balanced facility location models with varying edge lengths on trees.

In most inverse location models with varying weights, the weights of vertices are modified such that the given location of facilities become optimal. However, in many real world applications, the weights of vertices usually are referred to the clients on the vertices, therefore we cannot eliminate or add the clients, we just could transfer clients between vertices to make the location of given facilities optimal. Moreover, transferring population between vertices needs cost. Therefore, in this article we study the problem of the Inverse 1-Median Problem with Transferring (IMPT) weights between vertices on a tree. In other words, we consider transferring the clients between vertices of the tree with minimum cost such that a given vertex become 1-median. Therefore, the weights of some vertices are increased and some other are reduced, however, the total weight of the vertices in the tree remain unchanged. Note that this problem is more difficult than the classical inverse 1-median problem with vertices weight modification. As far as we know this kind of inverse facility location problem has not been considered by any researcher.

http://dx.doi.org/10.22108/toc.2023.136964.2047

In what follows, problem definition and some properties of IMPT are given in Section 2. Also, a linear programming model is presented in Section 2. A real life application on transferring users in a social network is presented in Section 3. Sections 4 and 5 contain numerical examples and the summary with concluding remarks, respectively.

2. Problem definition and its properties

Consider the tree $T = (V, E)$ with the set of vertices V, $|V| = n$, and the edge set E. Each vertex $v_i \in V$ has a positive weight w_i. Let W be the total weight of the vertices of T, i.e.

$$W = \sum_{i=1}^{n} w_i$$

The classical 1-median problem asks to find a vertex m of T, such that the sum of weighted distances from m to all other vertices in T is minimized. In the other words, the aim is to solve the following problem

$$\min \sum_{i=1}^{n} w_i d(m, v_i), \tag{2.1}$$

where $d(u, v)$ is the length of the shortest path between the vertices u and v. Let h be the degree of m and $T_1^m, \dots, T_h^m$ be the sub-trees of T, which are obtained by deleting the vertex m and all edges connected to m. For any sub-tree $T' \subseteq T$, let $W(T') = \sum_{v_i \in T'} w_i$. Goldman [10] presented the following optimality properties for the 1-median problem.

Lemma 2.1. *The vertex m is a 1-median of T if and only if*

$$\max_{1 \le i \le h} W(T_i^m) \le \frac{W}{2}.$$

Lemma 2.2. *Suppose T_a is a sub-tree of T. Then T_a contains the optimal solution of 1-median problem if and only if*

$$\sum_{\{i : v_i \in T_a\}} w_i \ge \frac{W}{2}.$$

Let the vertex $s \in T$ be a prespecified facility location. In the IMPT we want to move a part of the weight of some vertices into some other vertices with minimum cost so that s becomes the 1-median of T. Assume that the vertex m is the 1-median of the tree T relative to the current weights of the vertices in T. Let c_{ij} be the cost of transferring one unit of the weight of the vertex v_i to the vertex v_j, and p_{ij} denote the amount by which the weight of the vertex v_i is transferred to v_j. Then the

http://dx.doi.org/10.22108/toc.2023.136964.2047

IMPT can be written as the following linear programming.

$$(P_1)\ \min \sum_{v_i \in T} \sum_{v_j \in T} c_{ij} p_{ij}$$

$s.t$

$$\sum_{v_i \in T} \left(w_i - \sum_{v_j \in T} p_{ij} \right) d(v_i, s) + \sum_{v_i \in T} \left(w_i + \sum_{v_j \in T} p_{ji} \right) d(v_i, s)$$

$$\leq \sum_{v_i \in T} \left(w_i - \sum_{v_j \in T} p_{ij} \right) d(v_i, v_k) + \sum_{v_i \in T} \left(w_i + \sum_{v_j \in T} p_{ji} \right) d(v_i, v_k) \qquad k = 1, \dots, n, \tag{2.2}$$

$$w_i - \sum_{v_j \in T} p_{ij} \geq \underline{w_i} \qquad v_i \in T, \tag{2.3}$$

$$w_i + \sum_{v_j \in T} p_{ij} \leq \overline{w_i} \qquad v_i \in T, \tag{2.4}$$

$$p_{ij} \geq 0 \qquad v_i \in T \ \ , v_j \in T, \tag{2.5}$$

where $\overline{w_i}$ and $\underline{w_i}$ ($\underline{w_i} > 0$) are the upper and lower bounds of the weight of the vertex v_i. The objective function indicates minimizing the the total cost of transferring the weights of vertices. Constraints (2.2) imply that after optimal transferring the given vertex s is better than any other vertex in T. Note that in these constraints, $w_i - \sum_{v_j \in T} p_{ij}$ is the new weight of vertex v_i after transferring part of its weight to other vertices, and $w_i + \sum_{v_j \in T} p_{ij}$ is the new weight of vertex v_i after transferring some weights of other vertices to it. The lower and upper bounds of the weights of vertices are guaranty by constraints (2.3) and (2.4), respectively. Model (P_1) is a linear programming with n^2 variables and $3n$ constraints. In the following of this section we are going to reduce the number of variables and constraints by using some properties for the considered problem.

If $s = m$ then the transfer doesn't need. Therefore, we suppose $s \neq m$. Let P be the path between s and m. Also, let $e_1 \in P$ and $e_2 \in P$ be the edges adjacent to m and s, respectively. Assume T_1 is the subtree containing m which is obtained by deleting e_1 and T_2 is the subtree containing s which is obtained by deleting e_2 (see Figure 1).

Let A and B be the set of vertices in T_1 and T_2, respectively. Then by lemmas 2.1 and 2.2,

$$W(T_2) < \frac{W}{2},$$

$$W(T_1) \geq \frac{W}{2}.$$

Let $H = \frac{W}{2}$ *and* $D = H - W(T_2)$. Then Burkard et al. [3] showed that s is a 1-median of T if and only if $D = 0$. Therefore, we should transfer the weight of some vertices in T_1 to the vertices in

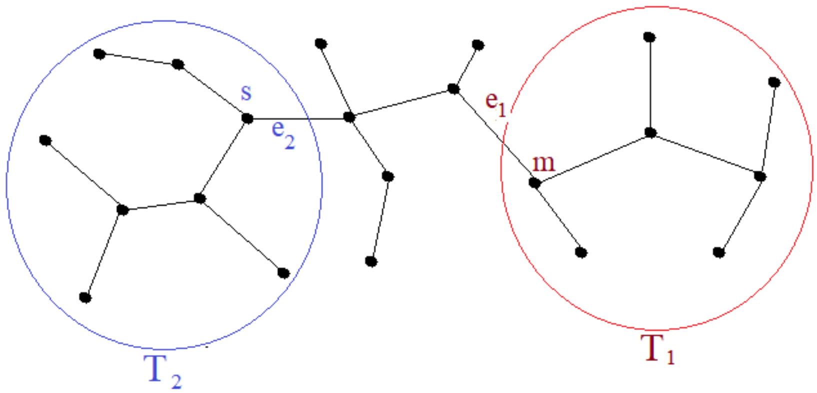

FIGURE 1. The sub-trees T_1 and T_2.

T_2 such that with respect to the new weights we obtain $W(T_2) \geq \frac{W}{2}$. The following properties are immediately concluded.

Lemma 2.3. *In the inverse 1-median problem via transfer vertices weight, to make the vertex s the 1-median of T, it suffices to transfer some weight of the vertices in A to the vertices in B.*

Lemma 2.4. *Transferring the weight of the vertices induced by the path $P' = P - \{m, s\}$, does not help to make the vertex s as the median.*

Proof. Suppose $W(T_2) < \frac{W}{2}$, then transferring the weight of some vertices in T_1 to P' and vise versa, does not affect on the weight of T_2. On the other side, since m is a 1-median thus $W(T_1) \geq \frac{W}{2}$. Therefore, transferring the weight of some vertices in p' to T_2 will not result that the weight ot T_2 reaches $\frac{W}{2}$. □

Using these properties the model of IMPT can be written as follows:

$$(P_2)\ \min \sum_{v_i \in A} \sum_{v_j \in B} c_{ij} p_{ij}$$

$s.t$

$$\sum_{v_i \in A} \left(w_i - \sum_{v_j \in B} p_{ij} \right) d(v_i, s) + \sum_{v_i \in B} \left(w_i + \sum_{v_j \in A} p_{ji} \right) d(v_i, s)$$

http://dx.doi.org/10.22108/toc.2023.136964.2047

$$\leq \sum_{v_i \in A}\left(w_i - \sum_{v_j \in B} p_{ij}\right) d\left(v_i, v_k\right) + \sum_{v_i \in B}\left(w_i + \sum_{v_j \in A} p_{ji}\right) d\left(v_i, v_k\right), \qquad k = 1, \ldots, n, \tag{2.6}$$

$$w_i - \sum_{v_j \in B} p_{ij} \geq \underline{w_i} \qquad v_i \in A, \tag{2.7}$$

$$w_i + \sum_{v_j \in A} p_{ij} \leq \overline{w_i} \qquad v_i \in B, \tag{2.8}$$

$$p_{ij} \geq 0 \qquad v_i \in A \ \ , v_j \in B. \tag{2.9}$$

Lemma 2.5. *The models* (P_1) *and* (P_2) *are equivalence.*

Proof. By Lemma 2.4, transferring the weight of vertices in P' to other vertices and vise verse does not affect on the optimality a solution. Moreover, by Lemma 2.3, to reach an optimal solution it suffices to transfer the weight of some vertices in A to some vertices in B. Thus we can remove the unnecessary terms in constraints 2.2 in model (P_1). □

Note that, since $|A| + |B| \leq n$ and constraints (2.7) and (2.8) indicate the limitations on upper and lower bounds of the weights of vertices, then model (P_2) contains at most $2n$ constraints. The worst case for variables is happen when $|A| = |B| = \frac{n}{2}$. In this case number of variables is $\frac{n^2}{4}$. The model (P_2) is better than (P_1), however model (P_2) can be replaced by the following model.

$$(P_3)\ min \sum_{v_i \in A} \sum_{v_j \in B} c_{ij} p_{ij}$$

$$s.t.$$

$$\sum_{v_i \in A} \sum_{v_j \in B} p_{ij} = \mathrm{D},$$

$$\sum_{v_j \in B} p_{ij} \leq w_i - \underline{w_i} \qquad v_i \in A,$$

$$\sum_{v_i \in A} p_{ij} \leq \overline{w}_j - w_j \qquad v_j \in B,$$

$$p_{ij} \geq 0 \qquad v_i \in A, \quad v_j \in B.$$

This model contains at most $\frac{n^2}{4}$ variables and $n + 1$ constraints.

Lemma 2.6. *The models* (P_2) *and* (P_3) *are equivalence.*

Proof. Since by transferring the weight of the vertices, the values of W and H do not change, then if d units of the weight of a vertex in A is transferred to a vertex in B, then the value of $W(T_1)$ as well as the value of D reduced and $W(T_2)$ increased by d units. Therefore, $D_{new} = H - (W(T_2) + d) = D - d$.

Thus the vertex s will be median if the sum of the weights of the vertices that transferred from the set A to the set B is equal to D. Therefore constraints (2.6) can be replaced by the following constraint

$$\sum_{v_i \in A} \sum_{v_j \in B} p_{ij} = \text{D}.$$

□

The following properties conclude from the previous discussions.

Lemma 2.7. *The following conditions are necessary for a feasible solution of IMPT:*

(1) $\sum_{v_i \in A} (w_i - \underline{w_i}) \geq D$,
(2) $\sum_{v_i \in B} (\overline{w_i} - w_i) \geq D$.

Lemma 2.7 indicates that for the given weights and upper and lower bounds, if any of the conditions in this lemma does not hold then the problem is infeasible, and by modifying the weights of vertices, s can not become median.

Note that the problem (P_3) can be interpreted as a transportation problem, where some products should be transferred from vertices in A to B. However, this model can not be solved by the classic method of solving transportation problem. This model differ from the classical model of transportation problem, because

(1) In the classical transportation model, there is a redundant constraint. Therefore, the dual variables can be found easily. But in model (P_3) the number of constraints depends on the number vertices in the sets A and B.
The dual of model (P_3) can be written as follow

$$\min \sum_{v_i \in A} (w_i - \underline{w_i})\, u_i + \sum_{v_j \in B} (\overline{w_j} - w_j)\, v_j + Da$$

$$s.t.$$

$$u_i + v_j + a \leq c_{ij} \qquad v_i \in A \quad v_j \in B,$$

$$u_i \leq 0 \qquad v_i \in A,$$

$$v_j \leq 0 \qquad v_j \in B,$$

This problem can not be solved as easy as the dual of classic transportation problem.

(2) In the classical transportation problem the units of commodity that supplied and requested by each of origins and destinations are fixed and the constraints are in equality form. But in model (P_3) the constraints can be in inequality form.

http://dx.doi.org/10.22108/toc.2023.136964.2047

3. A real world application

In this section an application of IMPT in the social networks is presented. Let SN be a social network which has some servers and many users through all the world. There is also a main server which is connected to all servers and support them. Suppose the connected network of servers is a tree and the number of users that connected to each server is considered as the weight of server. To optimize the time of servicing in this network, the main server should be located on the median vertex of the network. Let vertex m be the median of this network respect to the current assigning users to the servers. However, because of some political reasons and better control on network, the owners of SN like to put the main server in the vertex s. Therefore, they should transfer the weights of vertices by changing some assigning users to the servers such that the vertex s become median. Moving users from one server to another can result in the loss of a number of users. So for keeping the users some discounts are considered for the transferred users. These discounts are cussed some costs to the owners.

The upper bounds of the weights of vertices are depending on the capacity of the servers and bandwidth of lines between servers and main server. On the other side, low number of users causes low benefit for servers. Thus, the lower bounds of the weights of vertices are determined by the cost of servers.

FIGURE 2. The tree of social network SN on the world map.

As an example, consider the servers of the social network depicted in Figure 2 (the map is taken from geology.com). Table 1 contains the number of users that assigned to the servers of this network and the upper and lower bounds of the vertices.

http://dx.doi.org/10.22108/toc.2023.136964.2047

Table 1. Upper and lower bounds of weights of vertices and number of assigning users to the servers in social network SN.

Country	Vertex No.	Number of users (w_i)	$\underline{w_i}$	$\bar{w_i}$
Australia	v_1	3191000	1110000	6945000
India	v_2	12086000	1026000	17750000
China	v_3	11129000	6030000	15200000
Qatar	v_4	1086000	542000	8310000
Iran	v_5	1827000	742000	3034000
Kazakhstan	v_6	747000	124000	1136000
Russia	v_7	4138000	1027000	9829000
Egypt	v_8	1522000	645000	3290000
Nigeria	v_9	812000	231000	3430000
Angola	v_{10}	3672000	1524000	33500000
South Africa	v_{11}	2355000	982000	4500000
Algeria	v_{12}	423000	115000	2460000
Turkey	v_{13}	3230000	1287000	9032000
Ukraine	v_{14}	2972000	987000	8075000
Belarus	v_{15}	8365000	4186000	10123000
Sweden	v_{16}	3476000	538000	7598000
Germany	v_{16}	13745000	3874000	17543000
Italy	v_{18}	12821000	2439000	56473000
France	v_{19}	8782000	3260000	11234000
Canada	v_{20}	4236000	3470000	9876000
USA	v_{21}	11265000	5680000	15345000
Colombia	v_{22}	8763000	2730000	12143000
Brazil	v_{23}	9427000	4150000	15639000
Argentina	v_{24}	6286000	2850000	11043000

The median of this network is the vertex in Belarus. But, the owners are like to Qatar be median. To this aim using our proposed method we should move some users that assigned to the servers in the set A to the servers in the set B, where

$$A = \{v_{15}, v_{16}, v_{17}, v_{18}, v_{19},\ v_{20}, v_{21}, v_{22}, v_{23}, v_{24}\},$$

$$B = \{v_1, v_2, v_3, v_4, v_8,\ v_9, v_{10}, v_{11}, v_{12}\}.$$

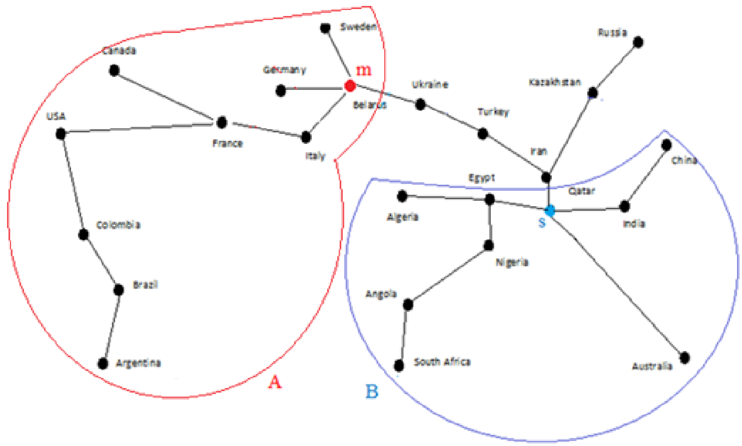

FIGURE 3. The sets A and B for the social network SN.

Figure 3 shows the tree of SN and the sets A and B. Table 2 contains the cost of transferring users from the servers in the set A to the set B. The total weight of vertices in this network is $W = \sum_{i=1}^{24} w_i = 108667000$.
Thus $H = W/2 = 54333500$. Also

$$W(T_1) = \sum_{v_j \in A} w_i = 48750000, W(T_2) = \sum_{v_j \in B} w_i = 36276000.$$

Therefore, the optimality gap is

$$D = H - W(T_2) = 54333500 - 36276000 = 18057500.$$

TABLE 2. The cost of transferring weight from the servers in A to servers in B.

	v_1	v_2	v_3	v_4	v_8	v_9	v_{10}	v_{11}	v_{12}
v_{15}	18.4	14.7	16.2	12.6	13.9	16.3	15.2	18.3	12.4
v_{16}	19.1	15.4	16.5	14.3	16.9	16.4	17.8	18.3	14.7
v_{17}	17.2	16.7	15.5	12.2	15.9	15.8	16.3	19.2	15.8
v_{18}	22.4	18.4	18.3	13.5	16.2	16.7	19.4	21.6	18.2
v_{19}	25.5	18.3	23.4	14.1	17.5	23.7	28.2	30.3	29.1
v_{20}	31.3	25.4	35.5	21.6	30.2	34.6	30.9	35.9	26.8
v_{21}	34.5	31.6	34.9	26.4	28.8	31.7	32.3	35.1	30.9
v_{22}	38.8	31.9	37.4	28.3	31.2	33.4	33.8	36.7	32.5
v_{23}	39.1	35.7	38.4	28.9	32.5	32.7	36.2	37.2	33.1
v_{24}	41.9	37.4	39.5	33.4	34.9	36.6	37.9	39.4	34.4

The conditions in Lemma 2.7 hold. The optimal solution of problem (P_3) for this example is reported in Table 3. The total transferring cost is 247599000. The new weights of vertices (which are denoted by $\hat{w}_i$) are reported in Table 4.

TABLE 3. The amount of weight transferred from the vertices in A to the vertices in B.

	v_1	v_2	v_3	v_4	v_8	v_9	v_{10}	v_{11}	v_{12}
v_{15}	0	374000	0	0	1768000	0	0	0	2037000
v_{16}	0	2938000	0	0	0	0	0	0	0
v_{17}	0	0	2647000	7224000	0	0	0	0	0
v_{18}	0	0	0	0	0	1069500	0	0	0
v_{19}	0	0	0	0	0	0	0	0	0
v_{20}	0	0	0	0	0	0	0	0	0
v_{21}	0	0	0	0	0	0	0	0	0
v_{22}	0	0	0	0	0	0	0	0	0
v_{23}	0	0	0	0	0	0	0	0	0
v_{24}	0	0	0	0	0	0	0	0	0

With considering the results in Table 4, one can observe that in this example the weights of vertices with lowest cost are transferred.

4. **Computational results**

In this section we examine some test problems for our proposed algorithm. We used 10 instances from 41 to 534 vertices that have been generated in [8]. For each instance, three given points, denoted by s, are considered. The weights of vertices and their lower and upper bounds are generated randomly in [80,100], [0,10] and [200,400], respectively. For each two vertices v_i and v_j, the length of shortest path between them is considered as the cost of transferring each unit of weight of v_i to v_j. All problems have been solved by IBM ILOG CPLEX Optimization Studio 12.1. The computational experiments were conducted on a Laptop AMD Ryzen 5, 3500U with 2.10 GHz processor and 8 Gb of RAM.

The results are given in Table 5. In this table the columns with the heading $nv(P_1)$ and $nc\,(P_1)$ indicate the number of variables and constraints in model (P_1), respectively. Also, the columns with the heading $nv(P_3)$ and $nc\,(P_3)$ indicate the same notations for model (P_3).

The results indicate decreasing the numbers of variables and constraints dramatically from model (P_1) to (P_3). The average of $\frac{nv(P_3)}{nv(P_1)}$ and $\frac{nc(P_3)}{nc(P_1)}$ is 0.1369 and 0.2774, respectively.

http://dx.doi.org/10.22108/toc.2023.136964.2047

TABLE 4. The weights of servers after transferring.

Country	Vertex No.	w_i	$\hat{w}_i$
Australia	v_1	3191000	3191000
India	v_2	12086000	**15398000**
China	v_3	11129000	**13776000**
Qatar	v_4	1086000	**8310000**
Iran	v_5	1827000	1827000
Kazakhstan	v_6	747000	747000
Russia	v_7	4138000	4138000
Egypt	v_8	1522000	**3290000**
Nigeria	v_9	812000	812000
Angola	v_{10}	3672000	3672000
South Africa	v_{11}	2355000	2355000
Algeria	v_{12}	423000	**2460000**
Turkey	v_{13}	3230000	3230000
Ukraine	v_{14}	2972000	2972000
Belarus	v_{15}	8365000	**4186000**
Sweden	v_{16}	3476000	**538000**
Germany	v_{17}	13745000	**3874000**
Italy	v_{18}	12821000	**11751500**
France	v_{19}	8782000	8782000
Canada	v_{20}	4236000	4236000
USA	v_{21}	11265000	11265000
Colombia	v_{22}	8763000	8763000
Brazil	v_{23}	9427000	9427000
Argentina	v_{24}	6286000	6286000

In Figure 4, the tree of instance tree2 and the sets A and B for the case $s = v_8$ are depicted. In this case, the number of vertices is 127, $|A| = 71$ and $|B| = 41$. Therefore, 15 vertices is not considered in model (P_3).

Figure 5 shows the percentage of the sum of weights of vertices in the sets A and B before and after transferring weights. As one can see after transferring (case b), the sum of weights of vertices in B is exactly 50% of sum of all weights.

TABLE 5. The results of solving model (P_3).

Tree#	n	$nv(P_1)$	$nc(P_1)$	m	s	$\lvert A\rvert$	$\lvert B\rvert$	$nv(P_3)$	$nc(P_3)$	$\frac{nv(P_3)}{nv(P_1)}$	$\frac{nc(P_3)}{nc(P_1)}$	Obj.func.	CPU(in sec)
Tree1	41	1681	123	v_4	v_3	27	12	324	40	0.193	0.325	11693	0.15
					v_6	28	13	364	42	0.217	0.341	4659	0.14
					v_7	28	13	364	42	0.217	0.341	5341	0.14
Tree2	127	16129	381	v_3	v_{10}	81	34	2754	116	0.171	0.304	76348	0.21
					v_8	71	41	2911	113	0.180	0.297	49582	0.51
					v_{13}	71	22	1562	94	0.097	0.247	143651	0.17
Tree3	159	25281	477	v_6	v_{10}	112	47	5264	160	0.208	0.334	36333.5	0.39
					v_{15}	112	18	2016	131	0.080	0.275	121686	0.51
					v_2	96	52	4992	149	0.197	0.312	41615.5	0.56
Tree4	186	34596	558	v_4	v_{20}	96	21	2016	118	0.058	0.211	163235	0.16
					v_{10}	94	32	3008	127	0.087	0.228	123487	0.21
					v_{15}	94	27	2538	122	0.073	0.219	129718	0.26
Tree5	243	59049	729	v_6	v_{10}	166	71	11786	238	0.200	0.326	119683	0.44
					v_2	166	76	12616	243	0.214	0.333	88876.5	0.55
					v_{20}	166	29	4814	196	0.082	0.268	307411.5	0.34
Tree6	301	90601	903	v_2	v_{20}	159	38	6042	198	0.067	0.219	390326	0.43
					v_{15}	234	52	12168	287	0.134	0.318	301433	0.85
					v_{10}	234	64	14976	299	0.165	0.331	224713	0.48
Tree7	344	118336	1032	v_4	v_{10}	191	88	16808	280	0.142	0.271	172662.5	0.52
					v_{20}	191	48	9168	240	0.077	0.233	387234.5	0.39
					v_{15}	191	55	10505	247	0.089	0.240	300480.5	0.70
Tree8	439	192721	1317	v_3	v_{10}	217	53	11501	271	0.060	0.206	479748	1.07
					v_{25}	217	51	11067	269	0.057	0.204	493634	0.49
					v_5	225	214	48150	440	0.250	0.334	686	1.39
Tree9	463	214369	1389	v_3	v_{15}	270	68	18360	339	0.086	0.244	468789	0.95
					v_5	270	193	52110	464	0.243	0.334	50529	1.32
					v_{10}	288	134	38592	423	0.180	0.305	191023	1.10
Tree10	534	285156	1902	v_3	v_5	408	84	34272	493	0.120	0.259	445392	1.20
					v_{30}	344	63	21672	408	0.076	0.215	550774	1.06
					v_{10}	408	61	24888	470	0.087	0.247	540647	1.25

http://dx.doi.org/10.22108/toc.2023.136964.2047

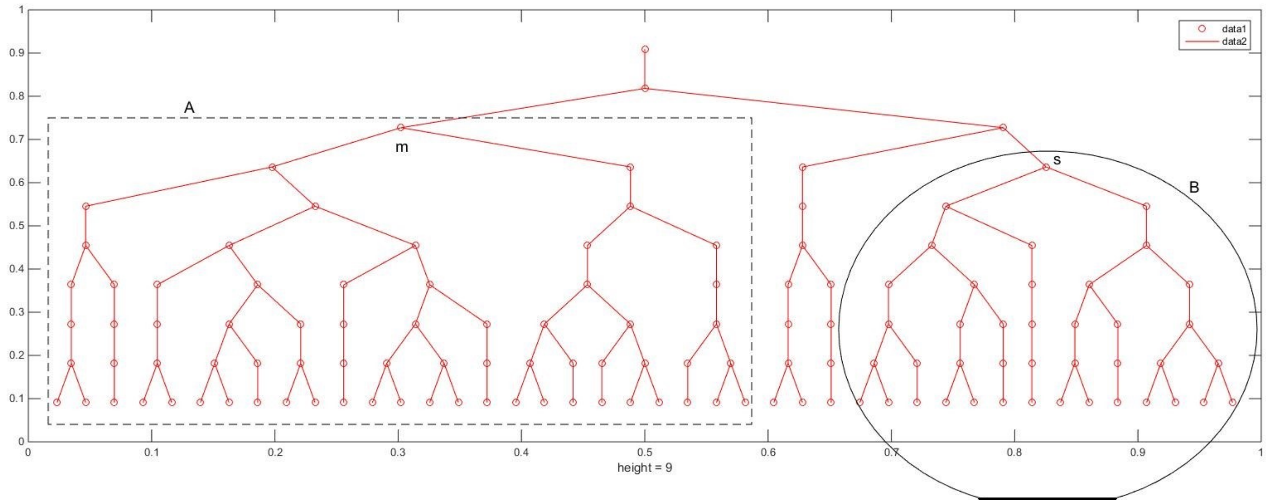

FIGURE 4. The sets A and B in the case $s = v_8$ for instance 2.

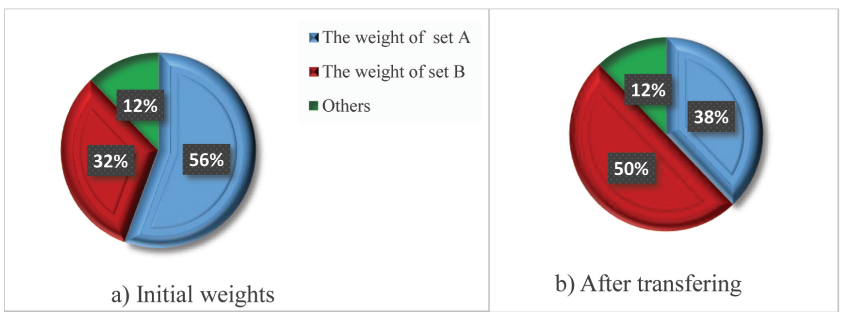

FIGURE 5. Percentage of the sum of the weight of vertices in sets A and B for instance tree2 with $s = v_8$.

5. Summary and conclusion

In this paper we consider a new case of inverse 1-median problem with variable vertex weights, in which the weights of vertices should be transferred. An efficient linear programming model was investigated. Then a real word application and numerical examples are presented to show the effectivity of the proposed model.

Note that in the previous versions of the inverse location models with variable vertex weights, modifying the weights with minimum cost has been considered. In this paper, we investigate the weight transferring instead of modifying. This strategy can also be applied for other kind of inverse location models such as inverse center problem and inverse continuous location models in the future works.

Acknowledgments

The authors wish to thank Professor Shahin Gelareh for his critical comments and suggestions on the paper. The authors would also like to thank the anonymous referee for his/her helpful comments.

References

[1] B. Alizadeh, R. E. Burkard and U. Pferschy, Inverse 1-center location problems with edge length augmentation on trees, *Computing*, **86** (2009) 331–343.

[2] B. Alizadeh and R. E. Burkard, Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees, *Networks*, **58** (2011) 190–200.

[3] R. E. Burkard, C. Pleschiutschnig and J. Z. Zhang, Inverse median problems, *Discrete Optim.*, **1** (2004) 23–39.

[4] R. E. Burkard, C. Pleschiutschnig and J. Z. Zhang, The inverse 1-median problem on a cycle, *Discrete Optim.*, **5** (2008) 242–253.

[5] M. C. Cai, X. G. Yang and J. Zhang, The complexity analysis of the inverse center location problem, *J. Global Optim.*, **15** (1999) 213–218.

[6] M. S. Daskin, *Network and Discrete Location: Models, Algorithms and Applications*, ohn Wiley & Sons, Inc., New York, 1995.

[7] H. A. Eiselt and V. Marianov, *Foundations of location analysis*, International Series in Operations Research and Management Science, Springer, 2011.

[8] J. Fathali and M. Zaferanieh, The balanced 2-median and 2-maxian problems on a tree, *J. Comb. Optim.*, **45** (2023) 16 pp.

[9] M. Galavii, The inverse 1-median problem on a tree and on a path, *Electron. Notes Discret. Math.*, **36** (2010) 1241–1248.

[10] A. J. Goldman, Optimal center location in simple networks, *Transportation Sci.*, **5** (1971) 212–221.

[11] X. Guan and B. Zhang, Inverse 1-median problem on trees under weighted Hamming distance, *J. Global Optim.*, **54** (2012) 75–82.

[12] S. L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, *Operations Research*, **12** (1964) 450–459.

[13] M. Nazari and J. Fathali, Inverse and reverse 2-facility location problems with equality measures on a network, *Iran. J. Math. Sci. Inform.*, **18** (2023) 211–225.

[14] K. T. Nguyen, Inverse 1-median problem on block graphs with variable vertex weights, *J. Optim. Theory Appl.*, **168** (2016) 944–957.

[15] K. T. Nguyen, T. H. Nguyen, H. Nguyen-Thu and T. T. Le, Pham V. H., On some inverse 1-center location problems, *Optimization*, **68** (2019) 999–1015.

[16] K.T. Nguyen and A. R. Sepasian, The inverse 1-center problem on trees with variable edge lengths under Chebyshev norm and Hamming distance, *J. Comb. Optim.*, **32** (2016) 872–884.

[17] S. Omidi and J. Fathali, Inverse single facility location problem on a tree with balancing on the distance of server to clients, *J. Ind. Manag. Optim.*, **18** (2022) 1247–1259.

[18] S. Omidi, J. Fathali and M. Nazari, Inverse and reverse balanced facility location problems with variable edge lengths on trees, *Opsearch*, **57** (2020) 261–273.

[19] A. R. Sepasian and F. Rahbarnia, An $O(nlogn)$ algorithm for the Inverse 1-median problem on trees with variable vertex weights and edge reductions, *Optimization*, **64** (2015) 595–602.

[20] A. Weber, *Uber den Standort der Industrial*, Tubingen Theory of Location of Industries, University of Chicago Press, (1909).

Tahere Sayar

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

Email:t.sayar.t@gmail.com

Jafar Fathali

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

Email: fathali@shahroodut.ac.ir

Mojtaba Ghiyasi

Department of Management and Accounting, Faculty of Industrial Engineering and Management Sciences, Shahrood University of Technology, Shahrood, Iran.

Email: mgt.ghiasi@gmail.com

http://dx.doi.org/10.22108/toc.2023.136964.2047