The Inverse $p$-Maxian Problem on Trees with Variable Edge Lengths

Kien Trung Nguyen
Mathematics Department, Teacher College, Cantho University, Vietnam.
Email: trungkien@ctu.edu.vn
April 14, 2015

Abstract

We concern the problem of modifying the edge lengths of a tree in minimum total cost so that the prespecified $p$ vertices become the $p$-maxian with respect to the new edge lengths. This problem is called the inverse $p$-maxian problem on trees. Gassner proposed efficient combinatorial algorithm to solve the the inverse 1-maxian problem on trees in 2008. For the problem with $p \geq 2$, we claim that the problem can be reduced to finitely many inverse 2-maxian problem. We then develop algorithms to solve the inverse 2-maxian problem for various objective functions. The problem under $l_1$-norm can be formulated as a linear program and thus can be solved in polynomial time. Particularly, if the underlying tree is a star, then the problem can be solved in linear time. We also devised $O(n \log n)$ algorithms to solve the problems under Chebyshev norm and bottleneck Hamming distance, where $n$ is the number of vertices of the tree. Finally, the problem under weighted sum Hamming distance is NP-hard.

keywords: Location problem; $p$-maxian; Inverse problem; Tree.

1 Introduction

Location theory plays an important role in Operations Research due to its numerous applications. Here, we want to find optimal locations of new facilities. Location theory was intensive investigated and solved, see Kariv and Hakimi [19, 20], Hamacher [17], Eiselt [12]. Recently, a new approach of location theory, the so-called inverse location problem, has been focused by many researchers. In the inverse setting, we aim to modify the parameters in minimum cost so that the prespecified locations become optimal in the perturbed problem.

For the inverse 1-median problems, Burkard et al. [7] solved the inverse 1-median problem on trees and the inverse 1-median problem on the plane with Manhattan norm in $O(n \log n)$ time. Then Galavii [15] proposed a linear time algorithm for the inverse 1-median problem on trees. Burkard et al. [6] investigated the inverse Fermat-Weber problem and solved the problem in $O(n \log n)$ time if the given points are not colinear. Otherwise, the problem can be formulated as a convex program. For the inverse 1-median problem
on a cycle, Burkard et al. [8] developed an \(O(n^2)\) algorithm based on the concavity of the corresponding linear program constraints. Additionally, the inverse \(p\)-median problem on networks with variable edge lengths is \(NP\)-hard, see Bonab et al. [5]. However, the inverse 2-median problem on a tree can be solved in polynomial time. More particularly, if the underlying tree is a star, the corresponding problem is solvable in linear time. Sepasian and Rahbarnia [22] investigated the inverse 1-median problem on trees with both vertex weights and edge lengths variations. They proposed an \(O(n \log n)\) algorithm to solve that problem. While most recently papers concerned the inverse 1-median problem under linear cost functions, Guan and Zhang [16] solved the inverse 1-median problem on trees under Chebyshev norm and Hamming distance by binary search algorithm in linear time.

Cai et al. [11] was the first who showed that although the 1-center problem on directed networks can be solved in polynomial time, the inverse problem is \(NP\)-hard. Hence, it is interesting to study some special situations of inverse 1-center problem on networks which are polynomially solvable. Alizadeh and Burkard [1] developed a combinatorial algorithm with complexity of \(O(n^2)\) to solve the inverse 1-center problem on unweighted trees with variable edge lengths, provided that the edge lengths remain positive throughout the modification. Dropping this condition, the problem can be solved in \(O(n^2c)\) time where \(c\) is the compressed depth of the tree. For the corresponding uniform-cost problem, Alizadeh and Burkard [2] devised improved algorithms with running time \(O(M \log M)\) and \(O(cn \log n)\). Then Alizadeh et al. [3] use the AVL-tree structure to develop an \(O(n \log n)\) algorithm for solving the inverse 1-center problem on trees with edge length augmentation. Especially, the uniform-cost problem can be solved in linear time. For the inverse 1-center problem on the simple generalization of tree graphs, the so-called cactus graphs, Nguyen and Chassein [21] showed the \(NP\)-hardness.

Although the inverse location problem was intensively studied, there is limited papers related to the inverse obnoxious location problem. Alizadeh and Burkard [4] developed linear time algorithm to solve the inverse obnoxious 1-center problem. Moreover, Gassner [13] investigated the inverse 1-maxian problem on trees with variable edge lengths and reduced the problem to a minimum cost circulation problem which can be solved in \(O(n \log n)\) time.

We focus in this paper the inverse \(p\)-maxian problem on trees with \(p \geq 2\). This paper is organized as follows. We briefly introduced the optimality criterion of the \(p\)-maxian problem on trees as well as formulate the problem under arbitrary cost function in Section 2. Section 3 concentrates on the problem under \(l_1\)-norm. We show that the problem can be formulated as finitely many linear programs. If the underlying tree is a star, we can solve the problem in linear time. We focus on Section 4 and Section 5 the problems under Chebyshev norm and Hamming distance, respectively. We show that these problem can be solved in \(O(n \log n)\) time.

2 The \(p\)-maxian problems in trees

Given a graph \(G = (V, E)\), each vertex \(v \in V\) associates with a non-negative weight \(w_v\) and each edge has a non-negative length \(\ell_e\). The length of the shortest path connecting two vertices \(u\) and \(v\) in \(G\) is the distance \(d(u, v)\) between these two vertices. A point on \(G\) is either a vertex or lies on an edge of the graph. In a
p-maxian problem on $G$ we aim to identify a set of $p$ points on $G$, say $X = \{x_1, x_2, \ldots, x_p\}$, to maximize the maxian objective function

$$F(X) = \sum_{i=1}^{M} w_i \max_{1 \leq j \leq p} d(v_i, x_j)$$

As proposed by Burkard et al. [9], the $p$-maxian problem on a tree is to find the subset of points $X \subset T$ with $|X| = p$ to maximize the objective function

$$F(X) = \sum_{i=1}^{M} w_i \max_{1 \leq j \leq p} d(v_i, x_j)$$

By the vertex domination property, there exists a $p$-maxian of $G$ that is the set of $p$ vertices. Furthermore, Burkard et al. [9] showed that $\{a, b\}$ is a 2-maxian of a tree $T$ if $P(a, b)$ is its longest path. We further show that this condition is also the necessary condition as follows.

**Theorem 1.** (2-Maxian Criterion)

Given two vertices $a, b$ on the tree $T$, then $\{a, b\}$ is a 2-maxian of $T$ if and only if $P(a, b)$ is the longest path in the tree.

**Proof.** For the sufficient condition, see [9]. Assume that $P(a, b)$ is not the longest path of the tree, we will prove that $\{a, b\}$ is not its 2-maxian as well. Let $m_{ab}$ be the midpoint of path $P(a, b)$. By deleting an edge that contains $m_{ab}$, we can deduce two parts $L$ and $R$ of $T$ which contain $a$ and $b$, respectively. Then we get

$$F(\{a, b\}) = \sum_{v \in L} d(v, b) + \sum_{v \in R} d(v, a).$$

Moreover, let $P(s, t)$ be the longest path of the tree. We trivially get $d(m_{ab}, a) \leq \max\{d(m_{ab}, s), d(m_{ab}, t)\}$ and $d(m_{ab}, b) \leq \max\{d(m_{ab}, s), d(m_{ab}, t)\}$, and at least one of the two inequalities does not hold with equality. Otherwise, it contradicts the assumption that $P(u, v)$ is the longest path. Therefore, we obtain $F(\{a, b\}) < F(u, v)$.

We now reformulate the optimality criterion in Theorem 1. For a leaf $v$ in $T$, we consider two paths $P(a, v)$ and $P(b, v)$. Let $v_{ab}$ be the common vertex of paths $P(a, v)$, $P(b, v)$, $P(a, b)$. Then we get the following result that states the conditions for $\{a, b\}$ to be the longest path in $T$.

**Lemma 1.** $P(a, b)$ is the longest path of $T$ if and only if $d(v, v_{ab}) \leq d(a, v_{ab})$ or $d(v, v_{ab}) \leq d(b, v_{ab})$ for all leaves $v$ in $T$.

Next we state the inverse version of the $p$-maxian problem on trees. Given a tree $T = (V, E)$ and a prespecified $p$-vertex. We can assume without loss of generality that the prespecified $p$ vertices are the leaves of $T$. The length of each edge $e$ in $E$ can be increased or decreased by an amount $p_e$ or $q_e$, i.e. the new length of $e$ is $\hat{\ell}_e = \ell_e + p_e - q_e$ and is assumed to be non-negative. We can formulate the inverse $p$-maxian on $T$ as follows.

1. The $p$-vertex becomes a $p$-maxian of the tree with respect to new edge lengths $\hat{\ell}$.
2. The cost function $F(p, q)$ is minimized.

3. Modifications are limited within the upper bounds, i.e. $0 \leq p_e \leq \bar{p}_e$ and $0 \leq q_e \leq \bar{q}_e$.

Burkard et al. [9] states that the set $S$ of $p$ vertices is a $p$-maxian of $T$ if and only if it contains a pair of vertices $\{a, b\}$ such that $P(a, b)$ is the longest path of $T$. Therefore, the inverse $p$-maxian problem on a tree can be reduced to $p^2$ 2-maxian problem on this tree. From now on, we will consider the 2-maxian problem on the tree $T$.

Consider a monotone cost function $C$ and $\{a, b\}$ is a pair of leaves which are the prespecified vertices, we get the following property.

**Proposition 1.** In the optimal solution of the inverse 2-maxian problem on $T$, it suffices to increase the lengths of edges in $P(a, b)$ and reduce the lengths of edges in $T \setminus P(a, b)$.

By Proposition 1, we can set $x_e := p_e$ and $\bar{x}_e := \bar{p}_e$ for $e \in P(a, b)$, and $x_e := q_e$ and $\bar{x}_e := \bar{q}_e$ for $e \in T \setminus P(a, b)$. An edge $e$ is said to be modified by an amount $x_e$ if its modified length is set to $\hat{\ell}_e := \ell_e + \text{sign}(e)p_e$, where $\text{sign}(e) = 1$ if $e \in P(a, b)$ and $\text{sign}(e) = -1$ if $e \notin P(a, b)$.

In summarize, we can formulate the problem as below.

$$
\begin{align*}
\min & \quad C(x) \\
\text{s.t.} & \quad \sum_{e \in P(a, v_{ab})} x_e + \sum_{e \in P(v, v_{ab})} \geq G(a, v), \forall v \in \mathcal{L} \\
& \quad \sum_{e \in P(b, v_{ab})} x_e + \sum_{e \in P(v, v_{ab})} \geq G(b, v), \forall v \in \mathcal{L} \\
& \quad 0 \leq x_e \leq \bar{x}_e, \forall e \in E.
\end{align*}
$$

(1)

3 The problem under $l_1$-norm

Assume that modifying an edge $e$ by a unit amount yields a cost $c_e$, then the objective function in (1) under $l_1$-norm can be written as

$$
C(x) = \sum_{e \in E} c_e x_e.
$$

The inverse 2-maxian problem under $l_1$-norm can be formulated as a linear program. It is therefore solvable in polynomial time. However, an efficient combinatorial algorithm is still unknown.

We now consider the case where the underlying tree is a star graph with center vertex $v_0$. We can directly deliver the following property from Lemma.

**Corollary 1.** (Optimality criterion)

Given two leaf nodes $a, b$ in a star graph $S$. Then, $\{a, b\}$ is a 2-maxian of the star graph if and only if $\ell_{(v_0, a)}$ and $\ell_{(v_0, b)}$ are two largest edges of the star graph.

For simplicity, we denote $\ell_{(v_0, a)}$ and $\ell_{(v_0, b)}$ by $\ell_a$ and $\ell_b$. By Lemma we consider how to modify the length of edges with a minimal cost such that $\hat{\ell}_a, \hat{\ell}_b$ becomes the two largest edge in the star. Asume that $\ell_a < \ell_b$, we analyze these two cases.
1. If $\tilde{l}_a \in [l_a, l_b]$, we do not increase the length of $l_b$ but decrease the length of $e \neq (v_0, b)$ with $l_e > \tilde{l}_a$ to $\tilde{l}_a$. We first presolve the problem as follows:

- If $\xi = \min_{e \neq (v_0, b): l_e > l_a} \{l_e - \bar{x}_e\} > l_a$, then increase $l_a$ by an amount $\xi - l_a$.
- If $\theta = l_a + \bar{x}_{(v_0, a)} < l_e$ for $e \neq (v_0, b)$, then we decrease the length of $l_e$ by an amount $l_e - \theta$.

If the presolution is not possible, then the problem is infeasible. Otherwise, we can formulate the problem as follows.

$$\min f(z) = c_a(z - l_a) + \sum_{e \neq (v_0, b): l_e > z} c_e(l_e - z)$$

where $z = \tilde{l}_a$.

2. If $l_a > l_b$, then the length of $e$ where $l_e > \tilde{l}_a$ must be decrease to $\tilde{l}_a$ and the length of $l_b$ must be increase to $\tilde{l}_a$. First of all we have to increase $l_a$ an amount $l_b - l_a$ and reprocess the problem as in the first case. If the preprocess is not possible then the problem is infeasible. Otherwise, we can formulate the problem as follows.

$$\min f(z) = (c_a + c_b)(z - l_b) + \sum_{e: l_e > z} c_e(l_e - z)$$

where $z = \tilde{l}_a = \tilde{l}_b$.

In both cases, the function $f(z)$ is a convex function. According to Alizadeh and Burkard [4], the problem can be solve in linear time. To get the optimal solution of the inverse 2-maxian problem on $S$, we have to solve two problems and get the best one. Therefore, we attain the following result.

**Theorem 2.** The inverse 2-maxian problem on a star graph can be solved in linear time.

## 4 Problem under Chebyshev norm

We also assume that modifying an edge $e$ by a unit amount yields a cost $c_e$, the objective function in [1] under Chebyshev norm can be written as

$$C(x) = \max_{e \in E} \{c_e x_e\}.$$

The solution method of the problem is based on the greedy modification. We define a maximum modification of edge lengths of $T$ with cost $C$ and follows.

**Definition 1.** (Maximum modification)

In a maximum modification with cost $C$ we modify each edge of $T$ as much as possible such that the cost is limited within $C$. It means we set
\[ x_e := \begin{cases} \frac{c_e}{\bar{x}_e}, & \text{if } c_e \bar{x}_e > C \\ \bar{x}_e, & \text{if } c_e \bar{x}_e \leq C. \end{cases} \]

We can solve the problem by the following two phases.

**Phase 1:** Find the interval that contains the optimal cost.

We sort the costs \( \{c_e x_e\}_{e \in E} \) in nondecreasing order and unite the similar values. Then we obtain a sequence of costs

\[ c_1 \bar{x}_1 < c_2 \bar{x}_2 < \ldots < c_m \bar{x}_m. \]

Here, \( m = O(n) \).

Now we aim to find the smallest index \( i_0 \in \{1, \ldots, m\} \) such that the gap \( G \) becomes nonpositive by applying the maximum modification with cost \( c_{i_0} \bar{x}_{i_0} \). We can find such an index by applying a binary search algorithm. If the application of the maximum modification with cost \( c_i \bar{x}_i \) results in a gap \( G > 0 \), we know that \( i_0 > k \). In the case where \( G \leq 0 \), we know that \( i_0 \leq k \).

We aim to find the smallest index \( i_0 \) such that \( P(a, b) \) become the longest path by applying the maximum modification with cost \( c_{i_0} \bar{x}_{i_0} \). The index \( i_0 \) can be sound by applying a binary search algorithm. If we apply the maximum modification with cost \( c_k \bar{x}_k \) but the path \( P(a, b) \) does not become the longest path then \( i_0 > k \). Otherwise, we know \( i_0 \leq k \).

Let us analyze the complexity to find \( i_0 \). In each iteration we modify the edge lengths of the tree in linear time. Then we find the longest path of the tree in linear time (see Handler [18]), and compare the length of \( P(a, b) \) with the longest one in order to decide if Lemma 1 holds or not in linear time. As the binary search stops in \( O(\log n) \) time, the procedure above runs in \( O(n \log n) \) time.

Assume that we have found the interval \( [c_{i_0-1} \bar{x}_{i_0-1}, c_{i_0} \bar{x}_{i_0}] \). We apply the maximum modification with cost \( c_{i_0-1} \bar{x}_{i_0-1} \) and get the modified tree \( \hat{T} \). Also, we update the upper bounds of modifications. The next step is to define a parameter \( t \in [0, c_{i_0} \bar{x}_{i_0}] \) and find the smallest value \( t \) such that \( P(a, b) \) becomes the longest path of \( \hat{T} \) with maximum modification with cost \( c_{i_0} t \).

**Phase 2:** Find the minimizer \( t \) with respect to optimal objective value in \( [0, c_{i_0} \bar{x}_{i_0}] \).

Consider a vertex \( v \) in the path \( P(a, b) \). For each vertex \( u \neq v \) and \( u \in P(a, b) \), we delete the branch containing \( u \) and its edges do not pass the path \( P(a, b) \). Then we get a tree \( T_v \) rooted at \( v \). Denote by \( \mathcal{L}(T_v) \) the set of leaves in \( T_v \) except \( a \) and \( b \). Now we try to modify the edge lengths of \( T \) so that there is no path \( P(v, v') \), for \( v' \in \mathcal{L}(T_v) \), is longer than \( \min \{d(v, a), d(v, b)\} \). We call this problem \((P_v)\). For solving this problem, we consider the modifying length of these following paths.

- For a path \( P(v, v') \) with \( v' \in \mathcal{L}(T_v) \) and \( d(v, v') > \min \{d(v, a), d(v, b)\} \), its modifying length is
  \[ d(v, v') - \sum_{e \in P(v, v')} \frac{e}{c_e}. \]
• For the paths \( P(v, a) \) and \( P(v, b) \), we obtain the modifying lengths

\[
d(v, a) + \sum_{e \in P(v, a)} \frac{t_e}{c_e},
\]

and

\[
d(v, b) + \sum_{e \in P(v, b)} \frac{t_e}{c_e}.
\]

The smallest value \( t \), such that the lengths of all paths \( P(v, v') \) for \( v' \in \mathcal{L}(T_v) \) are not larger than \( \min\{d(v, a), d(v, b)\} \), is

\[
\arg \min \max\{d(v, v') - \sum_{e \in P(v, v')} \frac{t_e}{c_e}, d(v, a) + \sum_{e \in P(v, a)} \frac{t_e}{c_e}, d(v, b) + \sum_{e \in P(v, b)} \frac{t_e}{c_e}\},
\]

where \( v' \in \mathcal{L}(T_v) \), \( d(v, v') > \min\{d(v, a), d(v, b)\} \) and \( t \in [0, \bar{x}_{i0}] \). This minimizer can be found in linear time, see Gassner [14].

To find the minimizer \( t^* \) in Phase 2, we have to solve all subproblems \( (P_v) \) for \( v \in (a, b) \) and get the largest one. It ensure that \( P(a, b) \) becomes the longest path of the tree \( \tilde{T} \) with maximum modification with cost \( c_{i0} t^* \).

Now let us analyze the complexity of the algorithm. Consider the problem \( (P_v) \), we use breath-first-search to find the modified length of paths \( P(v, v') \), \( P(v, a) \), \( P(v, b) \), then solve the subproblem in in \( O(|\mathcal{L}(T_v)|) \) time. Therefore, it yields a total time \( O(\sum_{v \in (a, b)} |\mathcal{L}(T_v)|) = O(n) \) time.

**Theorem 3.** The inverse 2-maxian problem on a tree can be solved in \( O(n \log n) \) time.

## 5 Problem under Hamming distance

First, let us focus on the problem under bottleneck Hamming distance. Assume that modifying one unit length of edge \( e \) costs \( c_e \), we aim to minimize the following objective function

\[
\max_{e \in E} \{c_e H(x_e)\}.
\]

Here, \( H \) is a Hamming distance and defined by

\[
H(\theta) := \begin{cases} 0, & \text{if } \theta = 0 \\ 1, & \text{otherwise} \end{cases}
\]

By the special structure of the Hamming distance, the objective function receives finitely many values, say \( \{c_e : e \in E\} \). Therefore, we can solve the problem by finding the smallest value in \( \{c_e : e \in E\} \) such that the optimality criterion in Lemma holds.

Number the edges in \( T \) by \( 1, \ldots, n \), and the corresponding costs are \( c_1, \ldots, c_m \), for \( m = n - 1 \). Let us first sort the costs \( \{c_e : e \in E\} \) increasingly and get without loss of generality a sequence

\[
c_1 < c_2 < \ldots < c_m.
\]
Now we apply a binary search algorithm to find the optimal cost. We start with cost \( c_k, k = \lfloor \frac{m+1}{2} \rfloor \). We modify all the edges with cost at less than or equal to \( c_k \). If the optimality criterion holds, we know that the optimal value is less than or equal to \( c_k \). Otherwise, it is larger than \( c_k \). In each iteration we recompute the length of the longest path in linear time (see Handler [18]) and compare the length of \( P(a, b) \) with that of the path. Moreover, as the binary search stops after \( O(\log m) \) iterations, Phase 1 runs in \( O(m \log m) \) time.

**Theorem 4.** The inverse 2-maxian problem on trees under bottleneck Hamming distance can be solved in \( O(n \log n) \) time.

For the problem under weighted sum Hamming distance, the objective function can be written as \( \sum_{e \in E} c_e H(x_e) \). We can easily reduce the Knapsack problem into an inverse \( p \)-maxian problem under weighted sum Hamming distance. Therefore, we get the following result.

**Theorem 5.** The inverse \( p \)-maxian problem on tree under weighted sum Hamming distance is \( NP \)-hard.

### 6 Conclusion

We have addressed the inverse \( p \)-maxian problem, \( p \geq 2 \), under various objective functions. It is shown that the problem can be reduced to \( p^2 \) many 2-maxian problems. Then we have formulated the inverse 2-maxian problem on trees under \( l_1 \)-norm as a linear program and solved the problem on star graphs in linear time. Furthermore, the inverse 2-maxian problem under Chebyshev norm and Hamming distance is solvable in \( O(n \log n) \) time, where \( n \) is the number of vertices in the tree. For future research topics, we will consider the inverse maxian problem on other classes of graphs, e.g., cacti, interval graphs, block graphs, etc.

### References

[1] B. Alizadeh and R.E. Burkard, Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees, Networks, Volume 58, Issue 3, pages 190-200, October 2011.

[2] B. Alizadeh and R.E. Burkard, Uniform-cost inverse absolute and vertex center location problems with edge length variations on trees, Discrete Applied Mathematics Vol. 159, pages 706-716, 2011.

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

[4] B. Alizadeh and R.E. Burkard, A linear time algorithm for inverse obnoxious center location problems on networks, CEJOR DOI 10.1007/s10100-012-0248-5, 2012.

[5] F.B. Bonab, R.E. Burkard, and E. Gassner, Inverse \( p \)-median problems with variable edge lengths, Math Meth Oper Res 73:263-280. DOI 10.1007/s00186-011-0346-5, 2011.

[6] R. E. Burkard, M. Galavii, E. Gassner. The inverse Fermat-Weber problem. European Journal of Operational Research 206 pp. 11-17, 2010.
[7] R.E. Burkard, C. Pleschiutschnig, and J.Z. Zhang, Inverse median problems, Discrete Optimization 1 23-39, 2004.

[8] R.E. Burkard, C. Pleschiutschnig, and J.Z. Zhang, The inverse 1-median problem on a cycle, Discrete Optimization 5 242253, 2008.

[9] R.E. Burkard, J. Fathali, and H. T. Kakhki, The p-maxian problem on a tree, Operations Research Letters, Vol. 35, 331-335, 2007.

[10] R.E. Burkard and J. Hatzl, Median problems with positive and negative weights on cycles and cacti, Journal of Combinatorial Optimization, Vol. 20, 27-46, 2010.

[11] M.C. Cai, X.G. Yang, and J.Z. Zhang, The complexity analysis of the inverse center location problem, J. Global Optimization, 15(2):213-218, 1999.

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

[13] E. Gassner, The inverse 1-maxian problem with edge length modification. J. Comb. Optimization, 16(1):5067, 2008.

[14] E. Gassner, Up- and downgrading the 1-center in a network, Eur. Jour. Oper. Res., Vol. 198, 70-377, 2009.

[15] M. Galavii, The inverse 1-median problem on a tree and on a path, Electronic Notes in Disrete Mathematics 36 1241-1248, 2010.

[16] X. Guan and B. Zhang, Inverse 1-median problem on trees under weighted Hamming distance, Journal of Global Optimization, Vol. 54, pp 75-82, 2011.

[17] H. W. Hamacher. Mathematische Lösungsverfahren für planare Standortprobleme. Vieweg and Teubner, 1995.

[18] G.Y. Handler, Minimax location of a facility in an undirected tree networks, Transp. Sci. 7 287-293, 1973.

[19] O. Kariv and S.L. Hakimi, An algorithmic approach to network location problems, I. The p-centers, SIAM Journal on applied Mathematics, vol. 37, no. 3, pp. 513-538, 1979.

[20] O. Kariv and S.L. Hakimi, An algorithmic approach to network location problems, II. The p-medians, SIAM Journal on applied Mathematics, vol. 37, no. 3, pp. 536-560, 1979.

[21] K.T. Nguyen and A. Chassein, Inverse eccentric vertex problem on networks, CEJOR, Doi: 10.1007/s10100-014-0367-2, 2014.

[22] A.R. Sepasian and F. Rahbarnia, An $O(n \log n)$ 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, 2013.