Computation of Inverse 1-Center Location Problem on the Weighted Trees

Biswanath Jana, Sukumar Mondal and Dr. Madhumangal Pal

***Abstract*---Let T be a tree with $(n+1)$ vertices and n edges with positive edge weights. The inverse 1-center problem on an edge weighted tree consists in changing edge weights at minimum cost so that a pre-specified vertex becomes the 1-center. In the context of location problems Cai et al. [9] proved that the inverse 1-center location problem with edge length modification on general un-weighted directed graphs is NP-hard, while the underlying center location problem is solvable in polynomial time. Alizadeh et al. [1] have designed an algorithm for inverse 1-center location problem with edge length augmentation on trees in $O(n \log n)$ time, using a set of suitably extended AVL-search trees. In [2], Alizadeh et al. have designed a combinatorial algorithm for inverse absolute on trees in $O(n^2)$ time when topology not allowed and $O(n^2 r)$ time when topology allowed. In this paper, we present an optimal algorithm to find an inverse 1-center location on the weighted trees with $(n+1)$ vertices and n edges, where the edge weights can be changed within certain bounds. The time complexity of our proposed algorithm is $O(n)$, if T is traversed in a depth-first-search manner.

***Keywords*--**Tree-Networks, Center Location, 1-Center Location, Inverse 1-Center Location, Inverse Optimization, Tree.

I. Introduction

LET $G = (V, E)$ be a connected graph with vertex set V and edge set E, such that $|V| = n+1$, $|E| = n$. Every edge $(u,v) \in E$ has different weight w_i. In G, a walk is defined as a finite alternating sequence of vertices and edges beginning and ending with vertices such that each edge is incident with the vertices preceding and following it. No edge appears more than once in a walk. A vertex, however, may appear more than once. An open walk in which no vertex appears more than once is called a path. A closed walk in which no vertex (except the initial and the final vertex) appears more than once is called a circuit. A tree T is a connected graph without any circuits. i.e., a tree is a connected acyclic graph. Clearly there is one and only one path between every pair of vertices of T. A tree T is weighted if there is a non-negative real number associated with each edge of T. In an un-weighted tree $T = (V, E)$, where $|E| = |V| - 1$, the eccentricity e(v) of the vertex v is defined as the distance from v to the vertex farthest from $v \in T$, i.e.,

$$e(v) = \max\{d(v, v_i), \text{ for all } v_i \in T\},$$

where $d(v, v_i)$ is the sum of the weights of the edges on the path between v and v_i.

In weighted tree $T = (V, E)$, the eccentricity e(v) of the vertex v is defined as the sum of the weights of the edges from v to the vertex farthest from $v \in T$, i.e.,

$$e(v) = \max\{d(v, v_i), \text{ for all } v_i \in T\},$$

where $d(v, v_i)$ is the sum of the weights of the edges on the path between v and v_i.

A vertex with minimum eccentricity in the tree T is called a center of that tree T, i.e., if $e(s) = \min\{e(v)$, for all $v \in V\}$, then s is the 1-center. It is clear that every tree has either one or two centers.

The eccentricity of a center in a tree is defined as the radius of the tree and is denoted by $\rho(T)$, i.e.,

$$\rho(T) = \{\min_{v \in T} e(v)\}.$$

The diameter of a tree T is defined as the length of the longest path in T i.e., the maximum eccentricity is the diameter. Let the weighted tree T with (n+1) vertices and n edges. The inverse 1-center problem on a tree consists in changing edge weights within certain bounds so that a pre-specified vertex becomes 1-center.

A. *Survey of Relevant Literature*

As shown in [8, 31, 32], the inverse problems of many combinatorial/network optimization problems can be solved by strongly or weakly polynomial algorithms. In fact in [29], it is shown that for a large class of combinatorial/network optimization problems, if the original problem can be solved in polynomial times, then its inverse problem can be solved in polynomial time by a quite uniform methodology. A detailed survey on inverse optimization problems has been compiled by Heuberger [21]. In the context of location problems Cai et al. [9] proved that the inverse 1-center location problem with edge length modification on general un-weighted directed graphs is NP-hard, while the underlying center location

Manuscript received on January 18, 2012, review completed on January 25, 2012 and revised on February 02, 2012.

Biswanath Jana, Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore -721 102, India.

Sukumar Mondal, Department of Mathematics, Raja N. L. Khan Women's College, Gope Palace, Paschim Medinipur-721 102, India.
E-Mail: sm5971@rediffmail.com, smnlkhan@gmail.com.

Dr. Madhumangal Pal, Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore -721 102, India. E-Mail: mmpalvu@gmail.com

Digital Object Identifier No: NCE022012007.

problem is solvable in polynomial time. In 2004, Burkard et al. [4] consider inverse p-median location problems can be solved in polynomial time, when p is fixed and not an input parameter. They proposed a greedy like $O(n \log n)$ time algorithm for the inverse 1-median problem with vertex weight modifications on tree networks. Galavi [17] showed later that this problem can actually be solved in $O(n)$ time. Moreover, Burkard et al. [4] proved that the inverse 1-median problem on the plane under Manhattan (or Chebyshev) norm can be solved in $O(n \log n)$ time. Later the same authors [5] investigated the inverse 1-median problem with vertex weight modification and unit cost on a cycle. They showed that this problem can be solved in $O(n^2)$ time by using methods from computational geometry. In 2007, Gassner [16] suggested an efficient $O(n \log n)$ time solution method for the inverse 1-maxian problem with edge length modifications on tree networks. The inverse Fermat-Weber problem was studied by Burkard et al. [6, 7]. The authors derived a combinatorial approach which solves the problem in $O(n \log n)$ time for unit cost and under the assumption that the pre-specified point that should become a 1-median does not coincide with a given point in the plane. Galavii [17] showed that the 1-median on a path with positve/negative weights lies in one of the vertices with positive weights or lies in one of the end points of the path. This property allows to solve the inverse 1-median problem on a path with negative weights in $O(n)$ time. Gassner [18] consider an inverse version of the convex ordered median problem and showed that this problem is NP-hard on general graphs, even on trees. Further, it was shown that the problem remains NP-hard for unit weights or if the underlying problem is a K-centrum problem (but not, if both of these conditions hold). The inverse unit-weight K-centrum problem with unit cost coefficients on a tree can be solved in $O(n^3 k^2)$ time. Recently, Yang and Zhang [30] proposed an $O(n^2 \log n)$ time solution method for the inverse vertex center problem on a tree provided that the modified edge lengths always remain positive. Dropping this condition, they mention that the general problem can be solved in $O(n^3 \log n)$ time. Recently, Alizadeh et al. [1] have designed an algorithm for inverse 1-center location problem with edge length augmentation on trees in $O(n \log n)$ time, using a set of suitably extended AVL-search trees. In [2], Alizadeh et al. have designed a combinatorial algorithm for inverse absolute on trees in $O(n^2)$ time when topology not allowed and $O(n^2 r)$ time when topology allowed.

Inverse optimization problems have recently attained significant theoretical interest due to their relevance in practice and for a comprehensive survey on inverse optimization problems see [13, 19, 21, 23].

Network location problems belong to basic optimization models which are concerned with finding the "best" location of single or multiple new facilities in a network of demand points such that a given function which depends on the distance between the facilities and clients becomes minimum. Depending on the the model under investigation, facilities or clients may either be placed only at vertices or may also lie on edges of the network. For further details on these problems the reader is referred to the books of Daskin [10], Drezner et al. [12], Francis et al. [15], Mirchandani et al. [26] and Nickel et al. [28].

Burton and Toint [3] first investigated an inverse shortest paths problem in 1992. Since then, many problems have been considered by various authors, working at least partly independently. The notation of 'inverse optimization' is always similar, but not the same. Handler [20] showed minimax location of a facility in an undirected tree graph. His paper is addressed to the problem of locating the absolute and vertex centers (minimax criterion) of an undirected tree graph. Hochbaum [22] introduced the pseudoflow algorithm for the maximum-flow problem that employs only pseudoflows and does not generate flows explicitly. The complexities of the pseudoflow algorithm, the pseudoflow-simplex, and the parametric variants of pseudoflow and pseudoflow-simplex algorithms are all O(mnlogn) on a graph with n nodes and m arcs. Zhang et al. [33] presented a strongly polynomial algorithm for a reverse location problem in tree networks. Megiddo [25] designed linear-time algorithms for linear programming in R^3.

In this paper, we propose an algorithm to compute inverse 1-center location problem on edge weighted trees in $O(n)$ time, where n is number of vertices of the tree.

B. Applications of the Problem

For instance, an important application comes from geophysical sciences and concerns predicting the movements of earthquakes. To achieve this aim, geologic zones are discretized into a number of cells. Adjacency relations can be modeled by arcs in a corresponding network (Moser [27]). Although some estimates for the transmission times are known, precise values are hard to obtain. By observing an earthquake and the arrival times of the resulting seismic perturbations at various points and assuming that earthquakes travel along shortest paths, the problem is to refine the estimates of the transmission times between the cells. This is just an inverse shortest path problem.

Another possible application actually changes the real costs: Assume that we are given a road network and some facility in it. The aim is to place the facility in such a way that the maximum distance to the customers is minimum. However we are often faced with the situation that the fac ility already exists and can not be relocated with reasonable costs. In such a situation, we may want to modify the network as little as possible (improving roads costs), such that the location of the facility becomes optimum (or such that the distances to the customers do not exceed some given bounds). This is an example of the inverse center location problem. When modeling traffic networks, a further option is to impose tolls in order to enforce an efficient use of the network (Dial [11]).

The choice of the word 'inverse optimization' was motivated in part by the widespread use of inverse methods in other fields, for instance Marlow [24] and Engl et al. [14].

C. Organization of the Paper

In the next section we shall discuss about preliminaries i.e., the formulation of inverse 1-center problem of the edge weighted tree. In Section 3, we present an algorithm to get inverse 1-center of the modified edge weighted tree. Some notations have also presented in this section. The time complexity is also calculated in this section. In Section 4 we give a conclusion.

II. PRELIMINARIES

Let $v_0, v_1, v_2, \cdots, v_{(n-1)}, v_n$ be an unweighted path between v_0 and v_n of the tree T such that $(v_k, v_{k+1}) \in E$ for $k = 0,1,2,3,\cdots,(n-1)$ and this path is denoted by $P(v_0, v_n)$. Clearly in a path $P(v_0, v_n)$, the length of the path, denoted by $\delta(P)$, is $d(v_0, v_n)$, i.e., $d(v_0, v_n) = \delta(P) = e(v_0) = e(v_n)$ and if this length of the path $P(v_0, v_n)$ is even, i.e., n is even, then radius of the path P is given by $\rho(P) = \delta/2 = d(v_0, v_n)/2$ and this is the eccentricity of the vertex $v_{\frac{n}{2}}$. So $v_{\frac{n}{2}}$ is the center of the path $P(v_0, v_n)$ and is at odd location when n is odd, then radius of the path P is given by $\rho(P) = (\delta + 1)/2$ and this is the eccentricity of each of the vertices $v_{\frac{n-1}{2}}$ and $v_{\frac{n+1}{2}}$, i.e., $v_{\frac{n-1}{2}}$ and $v_{\frac{n+1}{2}}$ are the two centers of the path $P(v_0, v_n)$ if n is odd.

Now we can introduce dummy vertex v_c in between $v_{\frac{n-1}{2}}$ and $v_{\frac{n+1}{2}}$ such that $(v_{\frac{n-1}{2}}, v_c) \in E$ and $(v_c, v_{\frac{n+1}{2}}) \in E$ so that v_c becomes the one center of the path $p(v_0, v_n)$.

Let v_j be the pre-specified vertex which is to be the center of the edge weighted path $P(v_0, v_n)$. Our problem is to minimize the cost of changing the weights of the edges in order to v_j to become the center of the path $P(v_0, v_n)$. Now in the following, we give the steps to show the inverse 1-center problem on the edge weighted path $P(v_0, v_n)$.

1) Find $d_1 = d(v_j, v_0)$ and $d_2 = d(v_j, v_n)$.

2) If $d_1 = d_2$, then v_j is 1-center.

3) If $d_1 < d_2$ we increase the edge weights of the edges $(v_j, v_{(j-1)}), (v_{(j-1)}, v_{(j-2)}), \cdots, (v_1, v_0)$ and decrease the edge weights of the edges $(v_j, v_{(j+1)}), (v_{(j+1)}, v_{(j+2)}), \cdots, (v_{(n-1)}, v_n)$ maintaining the following three conditions:

a) The vertex v_j becomes inverse 1-center of P with respect to $\overline{w}(e)$, where $e \in E(P)$ i.e., for all $p \in V(P)$,

$$max_{v \in V(P)} d_{\overline{w}}(v, v_j) \le max_{v \in V(P)} d_{\overline{w}}(v, p),$$

where $w = w(e)$ is the weight of the edge (positive real number) e and $\overline{w}(e)$ are the modified edge weights,

b) The linear cost function

$$\sum_{e \in E(P)} \{c^+(w)x(w) + c^-(w)y(w)\}$$

becomes minimum, where $x(w)$ and $y(w)$ are the maximum amounts by which the edge weight $w(e)$ is increased and reduced respectively. $c^+(w)$ is the non-negative cost if $w(e)$ is increased by one unit and $c^-(w)$ is the non-negative cost if $w(e)$ is reduced by one unit,

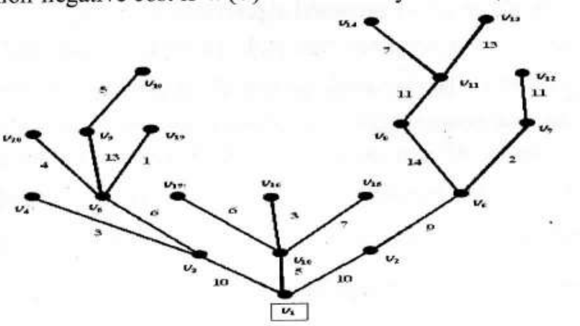

Fig.1 A Tree T

The new edge weights lie within given modification bounds

$$w_{low}(e) \le \overline{w}(e) \le w_{upp}(e)$$

for all edge in $E(P)$.

c) If $d_1 > d_2$ we increase the edge weights of the edges $(v_j, v_{j+1}), (v_{j+1}, v_{j+2}), \cdots, (v_{n-1}, v_n)$ and decrease the edge weights of the edges $(v_j, v_{j-1}), (v_{j-1}, v_{j-2}), \cdots, (v_1, v_0)$ maintaining the above three conditions.

Hence, based on the above conditions, the inverse 1-center location problem on the edge weighted tree T can be formulated as the following non-linear semi-infinite (or nonlinear) optimization model:

$$Minimize \sum_{e \in E(T)} \{c^+(w(e))x(w(e)) + c^-(w(e))y(w(e))\}$$

subject to

$$max_{v\in\Gamma(T)} d_{\bar{w}}(v,s) \le max_{v\in\Gamma(T)} d_{\bar{w}}(v,p)\text{, for all } p \in T (or\ p \in V(T)),$$

$$\overline{w}(e) = w(e) + x\{w(e)\} - y\{w(e)\}\text{, for all } e \in E(T),$$

$$x\{w(e)\} \le w^{+}\{w(e)\}\text{, for all } w \in E(T),$$

$$y\{w(e)\} \le w^{-}\{w(e)\}\text{, for all } w \in E(T),$$

$$x\{w(e)\}, y\{w(e)\} \ge 0\text{, for all } w \in E(T),$$

Where $w^{+}\{w(e)\} = w_{upp}(e) - w(e)$ and $w^{-}\{w(e)\} = w(e) - w_{low}(e)$ are the maximum feasible amounts by which $w(e)$ can be increased and reduced, respectively. Every feasible solution (x, y) with $x = \{x(w(e)) : e \in E(T)\}$ and $y = \{y(w(e)) : e \in E(T)\}$ is also called a feasible modification of the inverse 1-center location problem.

III. Algorithm and its Complexity

In this section we propose a combinatorial algorithm for the inverse 1-center location problem on the edge weighted tree T. The main idea of our proposed algorithm is as follows:

Let T be a weighted tree with $(n+1)$ vertices and n edges. Let V be the vertex set and E be the edge set. Let s be any non-pendant specified vertex in the tree T which is to be 1-center. At first we calculate the longest weighted path from s to any pendant vertex of T. Let sv_i be the longest weighted path from s to v_i and sv_j be the next longest path from s to v_j such that there is no common vertex except s. Now calculate the weights of two paths sv_i and sv_j. If weight of sv_i and sv_j are equal then s is the vertex 1-center as well as inverse 1-center of T. But our concentration is on unequal weights. If the weight of sv_i is greater than sv_j then we add the maximum weight with the pre weighted edge from s to v_j consecutively such that $w_{low}(e) \le \overline{w}(e) \le w_{upp}(e)$, for all $e \in E(T)$, where $w_{low}(e)$ and $w_{upp}(e)$ are the smallest and highest edge weights in T and $\overline{w}(e)$ be the modified edge weight. In this way if we seen the weight of sv_i and weight of sv_j are equal, then s be the inverse 1-center. But, if the weights of sv_i and sv_j are not equal, then we subtract the maximum weight from the pre weighted edge consecutively from the vertex s to v_i such that $w_{low}(e) \le \overline{w}(e) \le w_{upp}(e)$, for all $e \in E(T)$ and in this way the weights of sv_i and sv_j must become equal. Therefore, the pre-specified vertex s be the inverse 1-center of the edge weighted tree T.

Now, we introduce some notations for our algorithmic purpose.

R : Longest edge weighted path from s.
L : Another longest edge weighted path from s does not contain any vertex of the path R except s.
$w(R)$: Weight of the path R.
$w(L)$: Weight of the path L.
$w^{*}(R)$: Modified weight of the path R.
$w^{*}(L)$: Modified weight of the path L.

Our proposed algorithm is as follows:

A. *Algorithm 1-INV-LOC-TREE*

Input: Tree T with edge weight and specified vertex s.

Output: Vertex s as inverse 1-center of the tree T and modified tree T'.

- *Step 1.* Set s as a pre-specified vertex in T.
- *Step 2.* Compute the longest edge weighted path (only one) $R = sv_i$ from s to other vertex v_i on the given tree.
- *Step 3.* Next compute another longest edge weighted path (only one) $L = sv_j$ from s to the vertex v_j does not contain any vertex of the path R except s.
- *Step 4.* Calculate difference of the weights of two paths R and L i.e., $|w(R) - w(L)|$.
- *Step 5.*
  - *Step 5.1.* If $w(sv_i) = w(sv_j)$, $i \ne j$, then s is the vertex one center as well as inverse 1-Center of T.
  - *Step 5.2.* If $w(sv_i) > w(sv_j)$, then distribute the weight $w(sv_i) - w(sv_j)$ on the path sv_j, i.e., L, from s such that the bounds rule holds good.
  - *Step 5.2.1.* If $w^{*}(L) = w(R)$, then s is the vertex 1-center.
  - *Step 5.2.2.* If $w^{*}(L) < w(R)$, then we decrease the excess weight of the path R from s with bounds rule until $w^{*}(L) = w^{*}(R)$.
- *Step 5.3.* If $w(sv_i) < w(sv_j)$, then distribute the weight $w(sv_j) - w(sv_i)$ on the path sv_i, i.e., R, from s such that the bounds rule holds good.
  - *Step 5.3.1.* If $w^{*}(R) = w(L)$, then s is the vertex 1-center.

- *Step 5.3.2.* If $w^*(R) < w(L)$, then we decrease the excess weight of the path L from s with bounds rule until $w^*(R) = w^*(L)$.

- *end 1-INV-LOC-TREE.*

Using above *algorithm 1-INV-LOC-TREE* we can find out the inverse 1-center location problem on any edge weighted tree. Justification of this statement follows the following illustration.

Illustration of the algorithm to the tree T in Figure 1:

Let $s = v_1$ be the pre-specified vertex of the tree T which is to be one center. Next we find the longest path $R = v_1v_{13} = sv_{13}$ from s to other vertex v_{13} on the given tree and find next longest path $L = v_1v_{10} = sv_{10}$ from s to the vertex v_{10} does not contain any vertex of the path R except s.

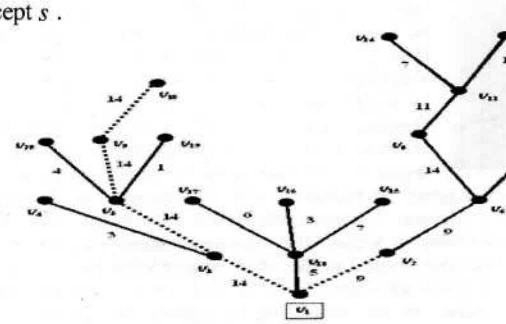

Fig. 2Modified Tree T' of the Tree T

Next calculate the weights of the paths R and L. Let $w(R)$ and $w(L)$ be the weights of the paths R and L respectively. Here $w(R) = 57$ and $w(L) = 34$. Next calculate the difference of weights of two paths i.e., calculate $w(R) - w(L)$. Therefore $w(R) - w(L) = 57 - 34 = 23$. To get equal weights of $w(R)$ and $w(L)$ we add the weight 4 to the edge (v_1, v_3), 8 to the edge (v_3, v_5), 1 to the edge (v_5, v_9), 9 to the edge (v_9, v_{10}) and decrease the weight 1 from the weight of the edge (v_1, v_2). After modification we get $w(L) = \{(10+4)+(6+8)+(13+1)+(5+9)\}$ $= 56$ and $w(R) = \{(10-1)+9+14+11+13\} = 56$ i.e., $w(L) = w(R)$. Therefore the vertex $s = v_1$ is the inverse one center.

Now, we have the modified tree T' with modified edge weights.

Next we shall prove the following important result.

Lemma 1 The algorithm 1-INV-LOC-TREE correctly computes the inverse 1-center location on the edge weighted tree.

Proof. Let s be the pre-specified vertex in T. We have to prove that s is the inverse 1-center. At first, by Step 2 we have calculated the weight of the path $R = sv_i$ and by Step 3, the weight of the path $L = sv_j$. If $w(R) = w(L)$, then s is the vertex 1-center as well as inverse 1-center (Step 5.1). But if $w(R) > w(L)$, then by Step 5.2 we have distributed the excess weight $w(R) - w(L)$ on the path L from s obeying the bounds conditions $w_{low}(e) \le \overline{w}(e) \le w_{upp}(e)$, for all $e \in E(T)$ and if $w(R) < w(L)$, then we have distributed the excess weight $w(L) - w(R)$ on the path R obeying the same bounds rule (Step 5.3). By this process we get $w^*(R) = w^*(L)$, which is the condition of inverse 1-center. Therefore s is the inverse 1-center. Hence *algorithm 1-INV-LOC-TREE* correctly computes the inverse 1-center for any edge weighted tree.

We have another important observation in the tree T' given by the *algorithm 1-INV-LOC-TREE.*

Lemma 2 The specified vertex s in the modified tree T' is the 1-center.

Proof. This result directly follows from Lemma 1. By *algorithm 1-INV-LOC-TREE,* finally we get $w^*(R) = w^*(L)$ in the modified tree T'. Therefore the specified vertex s in the modified tree T' is the 1-center.

The following describe the total time complexity of the algorithm to compute inverse 1-center problem on the edge weighted tree.

Theorem 1 The time complexity to find inverse 1-center problem on a given edge weighted tree T is $O(n)$, where n is the number of vertices of the tree.

Proof. Step 1 takes $O(1)$ time. Step 2 i.e., longest edge weighted path from s to v_i can be computed in $O(n)$ time if T is traversed in a depth-first-search manner. Similarly Step 3 can be computed in $O(n)$ time. Step 4 takes $O(1)$ time as comparing two numbers takes $O(1)$ time. Also Step 5 takes $O(1)$ time. Comparing two numbers and distribution of the excess weight takes $O(n)$ time. So, Step 5.2 and Step 5.3 can be computed $O(n)$ time. Hence overall time complexity of our proposed *algorithm 1-INV-LOC-TREE* is $O(n)$ time.

IV. Concluding Remarks

In this article, we investigated the inverse 1-center location problem with different edge weights on the tree. We developed exact combinatorial solution algorithm for the tree with fast running time $O(n)$.

ACKNOWLEDGMENTS

The authors thanks the anonymous referees for their thoughtful and valuable remarks which led to improvement of this paper.

REFERENCES

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

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

[3] Burton, D. and Toint, Ph. L., On an instance of the inverse shortest paths problem, Math. Programming, 53 (1992) 45-61.

[4] Burkard, R. E., Pleschiutschnig, C. and Zhang, J., Inverse median problems, Discrete Optimization, 1 (2004) 23-39.

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

[6] Burkard, R. E., Galavii, M. and Gassner, E., The inverse Fermat-Weber problem, Technical Report 2008-14, Graz University of Technology, Graz, 2008.

[7] Burkard, R. E., Alizadeh, B., Inverse center location problems, International Symposium on Combinatorial Optimization, 36 (2010) 105-110.

[8] Cai, M. C. and Li, Y. J., Inverse matroid intersection problem, ZOR Math. Meth. Oper. Res. 45 (1997) 235-243.

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

[10] Daskin, M. S., Network and discrete location: modeles, algorithms and applications, Wiley, New York, 1995.

[11] Dial, R. B., Minimal-revenue congestion pricing part-I: A fast algorithm for the single-origin case, Transportation Res. Part B, 33 (1999) 189-202.

[12] Drezner, Z., Hamacher, H. W., Facility location, applications and theory, Springer, Berlin, 2004.

[13] Duin, C. W. and Volgenant, A., Some inverse optimization problems under the Hamming distance, European Journal of Operational Research, 170 (2006) 887-899.

[14] Engl, H. W., Hanke, M. and Neubauer, A., Regularization of inverse problems, Kluwer Academic Publishers Group: Dordrecht, 1996.

[15]

[16] Francis, R. L., McGinnis, L. F. and White, J. A., Facility layout and location, an analytical approach. Prentice Hall, Englewood Cliffs, 1992.

[17] Gassner, E., The inverse 1-maxian problem with edge length modification, J. Combinatorial Optimization, 16 (2007) 50-67.

[18] Galavi, M., Inverse 1-median problems, Ph. D. thesis, Institue of Optimization and Discrete Mathematics, Graz University of Technology, Graz, 2008.

[19] Gassner, E., An inverse approach to convex ordered median problems in trees, Technical Report 2008-16, Graz University of Technology, Graz, 2008.

[20] Guan, X. and Zhang, J., Inverse bottleneck optimization problems under weighted Hamming Distance, Lecture Notes in Computer Science, 4041 (2006) 220-230.

[21] Handler, G. Y., Minimax location of a facility in an undirected tree graph, Transportation Science, 7 (1973) 287–293.

[22] Heuberger, C., Inverse combinatorial optimization: A survey on problems, methods, and results, Journal of Combinatorial Optimization, 8 (2004) 329-361.

[23] Hochbaum, D. S., The pseudoflow algorithm: A new algorithm for the maximum flow problem, Operations Research, 56 (2008) 992–1009.

[24] Kellerer, H., Pferschy, U. and Pisinger, D., Knapsack problems, Springer, Berlin, 2004.

[25] Marlow, B., Inverse problems, http://www.inverse-problems.com/. Megiddo, N., Linear-time algorithms for linear programming in R3 and related problems, SIAM J. Comput., 12 (1983) 759–776.

[26] Mirchandani, B. P. and Francis, R. L., Discrete location theory, Wiley, New York, 1990.

[27] Moser, T. J., Shortest paths calculation of seismic rays, Geophysics, 56 (1991) 59-67.

[28] Nickel, S., Puerto, J., Location theory, a unified approach, Springer, Berlin, 2005.

[29] Yang, C. and Zhang, J. Z., Two general methods for inverse optimization problems, Appl.Math. Lett. 12 (1999) 69-72.

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

[31] Zhang, J. Z. and Liu, Z. H., Calculating some inverse linear programming problems, J. Computational and Applied Mathematics, 72 (1996) 261-273.

[32] Zhang, J. Z. and Ma, Z. F., A network flow method for solving some inverse combinatorial optimization problems, Optimization, 37 (1996) 59-72.

[33] Zhang, J., Liu, Z. and Ma, Z., Some reverse location problems, European Journal of Operational Research, 124 (2000) 77–88.

Biswanath Jana was born in Paschim Medinipur, West Bengal, India on 8th day of January, 1980. He received his Bachelor of Science degree with Honours in Mathematics from Vidyasagar University, West Bengal, India in 2002 and he has passed Master of Science (Applied Mathematics) in Mathematics with first class from same University in 2004. He received his Bachelor of Education degree in 2006 from same University. Currently he is an Assistant Teacher in Mathematics of Jhargram Bikash Bharati Sikshayatan, Jhargram, Paschim Medinipur, West Bengal, India.

Sukumar Mondal was born in Paschim Medinipur, West Bengal, India on the fifth day of September, 1971. He received his Bachelor of Science degree With Honours in Mathematics (rank second) from Vidyasagar University, West Bengal, India in 1992 and he stood first class second in Master of Science (Applied Mathematics) degree in 1994 from same University. He received his Bachelor of Education degree in 1996 from same University. He joined the department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore, West Bengal, India, as a Junior Research Fellow on August 1997 and started research work for his Ph. D.degree. He was also awarded an individual researchfellowship by Council of Scientific and IndustrialResearch, Govt. Of India in 1997. He is currently a Ph. D. candidate in the department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, West Bengal, India. He was a Reader of department of Mathematics of Yogada Satsangha Palpara Mahavidyalaya, Palpara, Purba Medinipur, West Bengal, India. Currently he is Associate Professor of Dept. of Mathematics of Raja N. L. Khan Women's College, Medinipur, West Bengal, India.

Dr. Madhumangal Pal currently Professor of Applied Mathematics, Vidyasagar University, India. He has received Gold and Silver medals from Vidyasagar University for rank first and second in M.Sc. and B.Sc. examinations respectively. Also he received jointly with Prof. G.P.Bhattacherjee, "Computer Division Medal" from Institute of Engineers (India) in 1996 for best research work. Prof. Pal has successfully guided 11 research scholars for Ph.D. degrees and has published more than 82 articles in international journals, 20 articles in national journals, 11 articles in edited book and 26 articles in conference proceedings. His specializations include Computational Graph Theory, Fuzzy Correlation & Regression, Fuzzy Game Theory, Fuzzy Matrices, Genetic Algorithms and Parallel Algorithms.

Prof. Pal is the Editor-in-Chief of "Journal of Physical Sciences", and member of the editorial Boards of the journals "International Journal of Fuzzy Systems & Rough Systems", "International Journal of Computer Science, Systems Engineering & Information Technology", "Advanced Modelling and Optimization", "International Journal of Logic and Computation", "ISRN Discrete Mathematics" and "International Journal of Engineering Science, Advanced Computing and Bio-Technology". He is also a reviewer of several international journals.

Prof. Pal is the author of the books "Fortran 77 with Numerical and Statistical Analysis" published by Asian Books, New Delhi, "Numerical Analysis for Scientists and Engineers" and "Classical Mechanics" published by Narosa New Delhi and Alpha Sciences, Oxford, U.K., "Engineering Mathematics – Series", PHI Learning, New Delhi, "C Programs including Numerical and Statistical Methods", Narosa.

He organized several national seminars/ conferences/ workshop. Also, participated, delivered invited talks and chaired in national and international seminars/ conferences/ winter school/ refresher course in Indian and Abroad.