Forschungsschwerpunkt
Algorithmen und mathematische Modellierung

Uniform-cost inverse absolute and vertex center location problems with edge length variations on trees

Behrooz Alizadeh and Rainer Ernst Burkard

Project Area(s):
Effizient lösbare kombinatorische Optimierungsprobleme

Institut für Optimierung und Diskrete Mathematik (Math B)

Report 2009-11, July 2009

Uniform-cost inverse absolute and vertex center location problems with edge length variations on trees

Behrooz Alizadeh*† Rainer E. Burkard *

24/7/09

Abstract. This article considers the inverse absolute and the inverse vertex 1-center location problems with uniform cost coefficients on a tree network T with $n+1$ vertices. The aim is to change (increase or reduce) the edge lengths at minimum total cost with respect to given modification bounds such that a prespecified vertex s becomes an absolute (or a vertex) 1-center under the new edge lengths. First an $O(n \log n)$ time method for solving the height balancing problem with uniform costs is described. In this problem the height of two given rooted trees is equalized by decreasing the height of one tree and increasing the height of the second rooted tree at minimum cost. Using this result a combinatorial $O(n \log n)$ time algorithm is designed for the uniform-cost inverse absolute 1-center location problem on tree T. Finally, the uniform-cost inverse vertex 1-center location problem on T is investigated. It is shown that the problem can be solved in $O(n \log n)$ time if all modified edge lengths remain positive. Dropping this condition, the general model can be solved in $O(r_v n \log n)$ time where the parameter r_v is bounded by $\lceil n/2 \rceil$. This corrects an earlier result of Yang and Zhang.

Keywords: facility location problem, inverse optimization, combinatorial optimization, absolute and vertex 1-centers

1 Introduction

Inverse location problems have become an important aspect of optimization in recent years due to their role in practice and theory. Whereas classical location problems deal with finding the optimal location of one or more new facilities on network systems or in the space in order to satisfy the demands of customers optimally (see e.g. Daskin [8], Francis, McGinnis and White [9], Love, Morris and Wesolowsky [14] and Mirchandani and Francis [15]), the goal of inverse location optimization is to modify specific parameters (like edge lengths or vertex weights of network, point weights or point coordinates on space) of a given location problem at minimum total

*Graz University of Technology, Institute of Optimization and Discrete Mathematics, Steyrergasse 30, 8010 Graz, Austria. {alizadeh, burkard}@opt.math.tugraz.at

†Sahand University of Technology, Faculty of Basic Sciences for Engineering, Department of Applied Mathematics, Tabriz, Iran. alizadeh@sut.ac.ir

cost within certain modification bounds such that a given feasible solution of the problems becomes an optimal solution with respect to the new parameters values.

One of the well-known models in inverse location optimization is the inverse version of the center problems. The earliest study in this direction was performed by Cai, Yang and Zhang [7] in 1999 in order to prove the $\mathcal{NP}$-hardness of the inverse vertex center problem with uniform cost coefficients on directed graphs. Later, Gassner [12] showed that the inverse unit-weight k-centrum problem with uniform cost coefficients on a tree can be solved in $O(n^3k^2)$ time. This implies an $O(n^3)$ time algorithm for the uniform-cost inverse absolute center location problem on trees. In 2008, Yang and Zhang [17] developed an $O(n^2 \log n)$ time solution method for the inverse vertex 1-center location problem on trees which is based on linear programming arguments. Furthermore, the authors proposed an $O(n)$ time approach for solving the uniform-cost case. We show in this article by a counter example that in general this solution approach does not work correctly. The inverse 1-center location problem with edge length augmentation was treated by Alizadeh, Burkard and Pferschy [2]. Using a sequence of self-defined AVL-search trees, they designed an exact combinatorial algorithm with time complexity of $O(n \log n)$ on tree networks. Moreover, it was shown that the problem can be solved in $O(n)$ time if all cost coefficients are uniform. Recently, Alizadeh and Burkard [1] investigated the inverse absolute (and vertex) 1-center location problems on trees with arbitrary nonnegative cost for increasing or reducing the edge lengths. They developed a combinatorial $O(n^2)$ time algorithm for the inverse absolute 1-center location problem in which no topology change occurs on the given tree. Dropping this condition, an exact $O(n^2r)$ time algorithm was proposed for the general model where r, $r < n$, is the compressed depth of the underlying tree. Moreover, the authors showed that the inverse *vertex* 1-center location problem can be solved by a new approach with improved $O(n^2)$ time complexity, if all modified edge lengths remain positive. Finally, it was also shown that in the general case one gets an improved $O(n^2r_v)$ time complexity where the parameter r_v is bounded by $\lceil n/2 \rceil$.

Concerning inverse median problems see Burkard, Pleschiutschnig and Zhang [5, 6], Gassner [11, 12], Burkard, Galavii and Gassner [4], Galavii [10], Baroughi, Burkard and Alizadeh [3], and Pleschiutschnig [16]).

In this article we consider the uniform-cost inverse absolute and vertex 1-center location problems with edge length variations on a tree network. We develop new solution methods with improved time complexities. The remainder of this article is outlined as follows: In the next section, we describe the underlying models. Then we recall some fundamental properties from the classical absolute (and vertex) center location problems in order to derive the essential solution ideas for solving the problems under investigation. In Subsection 3.1, an $O(n)$ time greedy algorithm is stated for solving the uniform-cost tree height reduction problem. The construction of a corresponding height-reduction cost function of a rooted tree is treated in Subsection 3.2. This function returns the minimum cost for reducing the height of the given tree by any feasible amount. The results of the Subsections are applied in Section 4 in order to balance the heights of two rooted trees optimally in $O(n \log n)$ time by increasing the height of one tree and reducing the height of the other one at minimum total (uniform) cost. A new combinatorial algorithm with improved $O(n \log n)$ time complexity for the uniform-cost inverse absolute 1-center location problem on trees is developed in Section 5. Finally, in Section 6, the uniform-cost inverse vertex 1-center location problem on a tree is solved in $O(n \log n)$ time, provided that the modified edge lengths remain positive. Dropping this condition, one can get an $O(r_v n \log n)$ time solution method for the general case where $r_v \leq \lceil n/2 \rceil$.

2 Problem definition and basic properties

Let an undirected tree $T = (V(T), E(T))$ with vertex set $V(T)$, $|V(T)| = n+1$, and edge set $E(T)$ be given. Every edge $e \in E(T)$ has a positive length $\ell(e)$. We say that a point p lies in T, $p \in T$, if p coincides with a vertex or lies on an edge of T. In the *classical absolute (or vertex)* 1*-center location problem* on the given tree T, the goal is to find a point $p \in T$ (or $p \in V(T)$ respectively,) such that the maximum distance from any vertex $v \in V(T)$ to point p becomes minimum. Define

$$f_\ell(p) = \max_{v \in V(T)} d_\ell(v, p).$$

where $d_\ell(v, p)$ denotes the shortest path distance from v to p with respect to edge lengths ℓ on T. Then the absolute (or vertex) 1-center problem can be stated as

$$\begin{aligned} &\text{minimize} \quad f_\ell(p) \\ &\text{subject to} \quad p \in T \text{ (or } p \in V(T)). \end{aligned} \tag{1}$$

A point p^* which solves problem (1), is said to be an *absolute* 1*-center* (or a *vertex* 1*-center*, respectively). In 1973, Handler [13] proposed linear solution algorithms for determining the absolute and the vertex 1-center of a tree network.

The *uniform-cost inverse absolute (or vertex)* 1*-center location problem with edge length variations on trees* can be stated as follows:

Let s be a prespecified vertex on the given tree network T. We want to modify the edge lengths ℓ to $\tilde{\ell}$ at minimum total cost so that the prespecified vertex s becomes an absolute (or a vertex) 1-center of tree T. Every edge length $\ell(e)$ can only be modified between a lower bound $\ell_{low}(e) \geq 0$ and a upper bound $\ell_{upp}(e)$. Moreover, we assume that the cost for modifying each length $\ell(e)$ by one unit is the same, say 1. Therefore, the total cost is measured by the following linear function

$$\sum_{e \in E(T)} (x(e) + y(e)),$$

where $x(e)$ is the amount by which the length $\ell(e)$ is increased and $y(e)$ is the amount by which the length $\ell(e)$ is reduced. A solution vector (x, y) with $x = \{x(e) : e \in E(T)\}$ and $y = \{y(e) : e \in E(T)\}$ is called feasible if it guarantees that the vertex s is an absolute (or a vertex) 1-center of T and all bounds for the edge lengths are met. Thus the uniform-cost inverse absolute (or vertex) 1-center location problem on T

can be written as the following *nonlinear semi-infinite (or nonlinear) optimization model*:

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{e \in E(T)} (x(e) + y(e)) & \\
\text{subject to} & f_{\tilde{\ell}}(s) \leq f_{\tilde{\ell}}(p) & \text{for all } p \in T \text{ (or } p \in V(T)), \\
& \tilde{\ell}(e) = \ell(e) + x(e) - y(e) & \text{for all } e \in E(T), \\
& 0 \leq \ x(e) \leq \ell^+(e) & \text{for all } e \in E(T), \\
& 0 \leq \ y(e) \leq \ell^-(e) & \text{for all } e \in E(T),
\end{array}
\tag{2}
$$

where we set $\ell^+(e) = \ell_{upp}(e) - \ell(e)$ and $\ell^-(e) = \ell(e) - \ell_{low}(e)$.

Problem (2) is a special case of the more general inverse absolute (or vertex, resp.) 1-center location problem whith arbitrary cost coefficients for the modification of edge lengths. Therefore, it can be solved by the methods developed in Alizadeh and Burkard [1]. In the case of uniform cost coefficients, however, different, simpler algorithms with improved time complexities can be developed. Like in the general case these algorithms are based on the following fundamental theorem by Handler [13]:

Theorem 2.1 *In an unweighted tree network T the midpoint of a longest path in T is an absolute* 1*-center. The absolute* 1*-center is unique. Moreover, the closest vertex to the absolute* 1*-center is a vertex* 1*-center of T.*

Let $Z(T)$ denote the set of all leaves of T rooted in s. For any two vertices $u, v \in V(T)$, let P_{uv} denote the unique path from u to v. The length of P_{uv} with respect to edge lengths ℓ is denoted by $\ell(P_{uv})$. Let P_{sz^*} be a longest path from s to the leaves $z \in Z(T)$ and let e^* be the first edge on this path. The subtree T^{e^*} of tree T is the subtree generated by edge e^* and all its descendants. For solving the inverse absolute 1-center location problem, we split T into two subtrees L and R such that

$$L \cap R = s,$$

with $L = T^{e^*}$ whereas subtree R contains all other edges, in particular all leaves in $Z(T) \backslash Z(L - s)$. Both, L and R are rooted in vertex s. Theorem 2.1 implies immediately the following lemma:

Lemma 2.2 *Given an unweighted tree network T, a vertex s is the unique absolute* 1*-center if and only if*

$$\ell(P_{sz^*}) = \max\{\ell(P_{sz}) : z \in Z(R)\}$$

holds.

If the prespecified vertex s is not the midpoint of a longest path on T, we have to reduce the length of some edges on tree L and have to increase the length of some edges on tree R at minimum cost subject to the given bounds until the maximum distance from s to the leaves in $Z(L)$ equals the maximum distance from s to the leaves in $Z(R)$.

Now let us turn to the uniform-cost inverse vertex 1-center location problem. Consider a longest path P_{sz^*}, $z^* \in Z(T)$, from the prespecified vertex s to the leaves. Let $a(s)$ be the unique adjacent vertex to s on this path P_{sz^*}, i.e., $e^* = sa(s)$. If we delete edge e^*, then the tree network T splits into two disjoint subtrees $\hat{L}$ and $\hat{R}$. We root subtree $\hat{L}$ in $a(s)$ and subtree $\hat{R}$ in vertex s. The solution algorithms for the inverse vertex 1-center location problem are based on the following lemma (see Yang and Zhang [17]) which is again an immediate consequence of Theorem 2.1:

Lemma 2.3 *Given an unweighted tree network T, a vertex s is a vertex 1-center location if and only if*

$$\ell(P_{a(s)z^*}) \leq \max\{\ell(P_{sz}) : z \in Z(\hat{R})\}. \tag{3}$$

Lemma 2.3 leads to the following *main idea* for solving the inverse vertex 1-center problem: split T into two subtrees $\hat{L}$ and $\hat{R}$ and root them in the vertices $a(s)$ and s, respectively. If the optimality condition (3) is not satisfied, then reduce the edge lengths on $\hat{L}$ and increase the edge lengths on $\hat{R}$ at minimum total cost with respect to the given modification bounds such that the maximum distance from $a(s)$ to the leaves in $Z(\hat{L})$ becomes equal to the maximum distance from s to the leaves in $Z(\hat{R})$.

In the next section, we discuss the problem of reducing the height of a rooted tree with uniform cost coefficients.

3 Uniform-cost tree height reduction problem

3.1 An algorithm for optimal height reduction

Let $T = (V(T), E(T))$ be a tree with vertex set $V(T)$ and edge set $E(T)$ which is rooted in a vertex s. Every edge $e \in E(T)$ has a positive length $\ell(e)$ which is allowed to be reduced at most by the amount $\ell^-(e)$. Moreover, let $h_\ell(T)$ denote the *height* of T which is equal to the length of a longest path from the root s to the leaves of T with respect to the edge lengths ℓ. The *uniform-cost tree height reduction problem* deals with reducing the height $h_\ell(T)$ by a given amount δ, the so-called *reduction argument*, at minimum total cost where it is assumed that the cost for reducing each length $\ell(e), e \in E(T)$, by one unit is the same, say 1. To reduce the height $h_\ell(T)$ to the value $h_\ell(T) - \delta$ we have to reduce the edge lengths $\ell(e)$ by amounts $y(e)$ with respect to the modification bounds

$$0 \leq y(e) \leq \ell^-(e) \quad \text{for all } e \in E(T)$$

so that the total cost

$$\sum_{e \in E(T)} y(e)$$

becomes minimum.

Every path P_{sz} from the root s to a leaf $z \in Z(T)$ can be reduced at most by the amount

$$\ell^-(P_{sz}) = \sum_{e \in E(P_{sz})} \ell^-(e).$$

Thus the shortest feasible height of the tree is given by

$$h_{\min} = \max_{z \in Z(T)} \{\ell(P_{sz}) - \ell^-(P_{sz})\}.$$

In other words, the tree T can be reduced by any amount $0 \leq \delta \leq \delta_{\max}$ with

$$\delta_{\max} = h_\ell(T) - h_{\min}. \tag{4}$$

Obviously, $\delta_{\max}$ can be computed in linear time.

In order to reduce the height of a tree T by δ, $0 < \delta \leq \delta_{\max}$, we proceed as follows. In a first step we compute for every edge e of T the height $h(e)$ of the subtree T^e starting with edge e. Beginning from the leaves this can be done in $O(n)$ time. Now we direct all edges from the root to the leaves. In the second step we scan all edges of the tree in a broad first way starting with the edges $e = sv$. Every edge $e = uv$ gets two labels (l_u, δ_e). The label l_u is the length of the path from root s to vertex u (with respect to the modified edge lengths) and δ_e is the amount by which the height of the subtree starting with edge e has to be reduced in order that tree T is reduced by δ. The root gets the label $l_s = 0$. For any edge $e = uv$ we do:

If $l_u + h(e) > h_\ell(T) - \delta$ do:

- $\mu = \min(\delta_e, \ell^-(e))$;
- $\ell(e) = \ell(e) - \mu$;
- $l_v = l_u + \ell(e)$;
- For all descendant edges $e' = vw$ set $\delta_{e'} = \min(0, \delta_e - \mu)$.

Altogether we get

Theorem 3.1 *The uniform-cost tree height reduction problem can be solved in $O(n)$ time on a tree with n edges.*

In the next subsection we construct a piecewise linear function $C(\delta)$ which returns the minimum cost for reducing the height of the rooted tree T for any feasible argument δ.

3.2 Construction of the height-reduction cost function $C(\delta)$

In order to solve the inverse 1-center problem we need the cost for reducing the tree L by any feasible amount δ. This cost is given by the *height-reduction cost function* $C(\delta)$. In this subsection we describe how this height-reduction cost function can be described in an efficient way.

For a given tree T we first construct an auxiliary *reduction tree* T_r. Tree T_r has the same vertices and edges as tree T, but its edge lengths are given by the reductions of the edge lengths in T, when tree T is reduced by $\delta_{\max}$. This means, when the length of edge e during the procedure outlined in Subsection 3.1 is reduced by $\tilde{y}(e)$ (in order to reduce the height of T by $\delta_{\max}$), then the length of edge e in T_r equals $\ell_r(e) = \tilde{y}(e)$. This implies that in tree T_r all edge lengths can be reduced to 0, i.e., $\ell_r^-(e) = \ell_r(e) = \tilde{y}(e)$.

Example. Let the left tree in Fig.1 be given. The labels $(\ell(e), \ell^-(e))$ describe the given length $\ell(e)$ of edge e and the amount $\ell^-(e)$ by which this length can be reduced. The tree contains three paths from s to the leaves, namely

- P_{sa} of length 11 which can be reduced by 5,
- P_{sb} of length 14 which can be reduced by 9,
- P_{sc} of length 9 which can be reduced by 5.

Thus $h(T) = 14$ and $\delta_{\max} = 8$ according to (4). The reduction tree becomes the right tree shown in Fig.1.

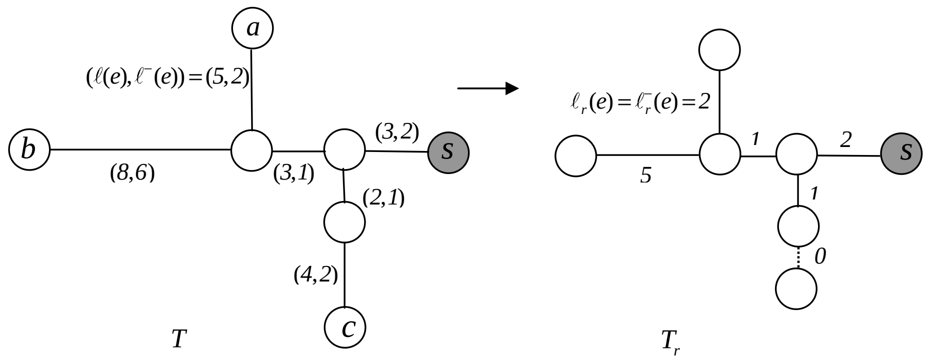

Figure 1: T_r is the auxiliary reduction tree of T.

The connection between T and T_r is given by the following lemma:

Lemma 3.2 *There exists a one-to-one correspondence between feasible solutions of the following two problems (a) and (b) and they yield the same objective function values:*

Problem (a): reduce the height $h_\ell(T)$ of tree T by the amount δ with respect to edge lengths ℓ and bounds ℓ^-.
Problem (b): reduce the height $h_{\ell_r}(T_r)$ of the reduction tree T_r by the amount δ with respect to the edge lengths ℓ_r and bounds ℓ_r^-.

Lemma 3.2 implies that the two trees T and T_r lead to the same height-reduction cost function $C(\delta)$. In order to compute this function we use tree T_r instead of tree T. Reducing the height $h_\ell(T)$ by the maximal amount $\delta_{\max}$ means that the height $h_{\ell_r}(T_r)$ is reduced to the value 0.

The height-reduction cost function $C(\delta)$ is a piecewise linear function which can be represented by the start point $\delta_0 = 0$, the end point $\delta_k = \delta_{\max}$ and $k-1$ breakpoints δ_i where the slopes of $C(\delta)$ change. The slopes a_i, $i = 1, ..., k$, equal the number of edges whose lengths are currently simultaneously reduced. So, if T_r has m vertex disjoint longest paths, then we get for the first slope $a_1 = m$. Let $\deg(v)$ denote the degree of vertex v in a tree. Then the number of breakpoints equals

$$B(T_r) = \begin{cases} \sum\limits_{v \in V(T_r)} \max(0, \deg(v) - 2) + 1 & \text{if } \deg(s) \geq 2, \\ \sum\limits_{v \in V(T_r)} \max(0, \deg(v) - 2) & \text{if } \deg(s) = 1. \end{cases}$$

The breakpoints of the function $C(\delta)$ can be computed by the following procedure.

HRCF determines $\delta_1, \ldots, \delta_{k-1}$ and the height-reduction cost function $C(\delta)$

Step 1. Associate with every edge $e = uv \in E(T_r)$ two labels $(h(e), z(e))$. The label $h(e)$ is the height of the subtree T_r^e beginning with edge $e = uv$. The label $z(e)$ contains the name of the leaf z such that P_{uz} is a path of maximal length in the subtree T_r^e. Starting from the leaves to the root of T_r in a breadth-first approach all labels $(h(e), z(e))$ are computed in $O(n)$ overall time.

Step 2. The labels of the edges incident with the root s allow to identify a longest path in T_r. All edges of this longest path are deleted. Thus we get a forest F of oriented subtrees of T_r with root set $R(F)$.

Step 3. For all $v \in R(F)$ and every edge $e = vw$ we get a new subtree T_r^e. We number these edges by $e_1, ..., e_t$ and get corresponding subtrees $T_r^1, \ldots, T_r^t$. For every $i = 1, ..., t$, a new breakpoint is found by

$$\delta = \delta_{\max} - h(e).$$

Step 4. For every $i = 1, \ldots, t$, execute Steps 2 and 3 on the subtrees T_r^i provided T_r^i is not just a vertex. The decomposition of the tree T_r to subtrees terminates when all subtrees are just single vertices.

Step 5. Order the breakpoints $0 < \delta_1 \leq \delta_2 \leq ... \leq \delta_{B(T_r)}$ increasingly. In any such breakpoint the slope of $C(\delta)$ increases by 1. Thus the final slope is $B(T_r)$.

Obviously, one can unite equal δ-values and increase the slope by their number. Thus we get

$$0 = \delta_0 < \delta_1 < ... < \delta_{k-1} < \delta_k = \delta_{\max},$$

and strictly increasing slopes $a_1, \ldots, a_k$ with $a_k = B(T_r)$. Let

$$\begin{aligned} &C_1(\delta) = a_1\delta, \\ &C_i(\delta) = C_{i-1}(\delta_{i-1}) + a_i(\delta - \delta_{i-1}) \quad \text{for } i = 2, \ldots, k. \end{aligned}$$

Then the function $C(\delta)$ is given by

$$C(\delta) = \begin{cases} C_1(\delta) & \text{if } \delta_0 \leq \delta \leq \delta_1, \\ C_2(\delta) & \text{if } \delta_1 \leq \delta \leq \delta_2, \\ \vdots & \\ C_k(\delta) & \text{if } \delta_{k-1} \leq \delta \leq \delta_k. \end{cases}$$

The following lemma shows the correctness of Procedure HRCF.

Lemma 3.3 *Procedure HRCF constructs and represents correctly the height-reduction cost function $C(\delta)$ of T_r.*

Proof. Every path P_i of length $\ell(P_i)$ in the decomposition above contributes the term

$$C_i(\delta) = \max(0, \delta - \ell(P_i))$$

to the function $C(\delta)$. Indeed, the reduction of the length of this path starts, when the height of the tree becomes smaller than $\ell(P_i)$. In this case the path must be shortened and contributes one unit to the cost until it reaches the length 0. Note that for this property it is essential to reduce in all subtrees a *longest* path first. The function $C(\delta)$ can be expressed as

$$C(\delta) = \sum_{i=1}^{B(T_r)} C_i(\delta).$$

■

Example (cont'd)

The tree T_r in Fig.1 has height 8. Thus the function $C(\delta)$ is defined for $0 \leq \delta \leq 8$. When the longest path in T_r is deleted, we are left with two disjoint paths of lengths 1 and 2, respectively. Thus we get breakpoints for $\delta_1 = 8 - 2 = 6$ and $\delta_2 = 8 - 1 = 7$. This yields the following function $C(\delta)$

$$C(\delta) = \begin{cases} \delta & \text{if } 0 \leq \delta \leq \delta_1 = 6, \\ 6 + 2(\delta - 6) & \text{if } 6 \leq \delta \leq \delta_2 = 7, \\ 8 + 3(\delta - 7) & \text{if } 7 \leq \delta \leq \delta_k = 8. \end{cases}$$

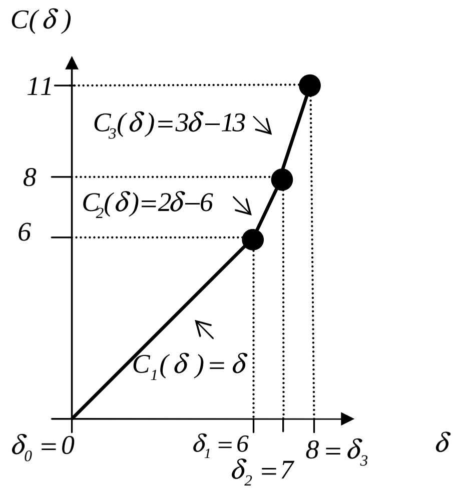

Figure 2: Graph of the height-reduction cost function $C(\delta)$ with height variable δ of tree T given in Figure 1.

This function is illustrated in Figure 2.

Let us turn to the complexity for the construction and representation of the function $C(\delta)$. We know that $|E(T_r)| = n$. The computation of all labels $(h(e), z(e))$ takes $O(n)$ time. Moreover, the decomposition of T_r into subtrees and therefore the determination of the values δ_i are done in $O(n)$ time. The slopes a_i are obtained in $O(n \log n)$, since we have to sort the corresponding δ_i values of the breakpoints. The terms $C_i(\delta)$, $i = 1, ..., k$, are computed in $O(n)$ time. Altogether, we get the following result.

Lemma 3.4 *Given a rooted tree T with n edges, the height-reduction cost function $C(\delta)$ can be constructed and represented in $O(n \log n)$ time.*

In the next section we use the height-reduction cost function in order to balance the heights of two rooted trees in an optimal way.

4 Height balancing of two rooted trees with uniform cost coefficients

Let two rooted trees T' and T'' with roots s' and s'', respectively, be given. W.l.o.g. let $h(T') > h(T'')$. The task of the *height balancing problem* is to reduce the edge lengths in T' and to increase the edge lengths in T'' such that both trees have the same height and the sum of length changes is minimum. Suppose that $\ell^+(e)$ and $\ell^-(e)$ are the maximum amounts by which the length $\ell(e)$ is allowed to be increased and reduced, respectively. Therefore we either increase $\ell(e)$ by an amount

$x(e)$ or reduce $\ell(e)$ by an amount $y(e)$ within the bounds $0 \leq x(e) \leq \ell^+(e)$ and $0 \leq y(e) \leq \ell^-(e)$ such that

$$h_{\tilde{\ell}}(T') = h_{\tilde{\ell}}(T'')$$

holds under the new edge lengths $\tilde{\ell}$ and the sum of changes

$$\sum_{e \in E(T' \cup T'')} (x(e) + y(e))$$

is minimum.

In order to increase the height of T'' to some height $h > h(T'')$ by minimum edge-length changes one has to increase the length of just one path $P_{s''z^0}$ from s'' to a leaf z^0 in T'' (cf. Lemmas 4.1 and 4.2 in [1]). We call this path *best candidate path*. Before discussing a solution method for the height balancing problem we observe that it is always not worse to increase the length of the best candidate path as far as possible than to reduce the height of tree T'. Indeed, increasing the length of path $P_{s''z^0}$ by δ yields a total change of just δ which is always less or equal to $C(\delta)$.

The maximum feasible amount by which the length of a path $P_{s''z}$ can be increased is given by

$$\ell^+(P_{s''z}) = \sum_{e \in E(P_{s''z})} \ell^+(e).$$

We define

$$Z_0 = \{z \in Z(T'') : \ell(P_{s''z}) + \ell^+(P_{s''z}) \geq h_\ell(T') - h_\ell(T'')\}.$$

If $Z_0 \neq \emptyset$, then an optimal solution can be found by increasing some edge lengths on T''. In this case we determine

$$z^0 \in \text{argmax}\{\ell(P_{s''z}) : z \in Z_0\}$$

and increase the length of $P_{s''z^0}$ up to the height $h_\ell(T')$.

If $Z_0 = \emptyset$, then we cannot balance the heights of the two trees by just increasing the length of a path in T''. In addition a height reduction of tree T' is needed. Observe that a path $P_{s''z}$ with $z \in Z(T'')$ can never be the best candidate path if

$$\ell(P_{s''z}) + \ell^+(P_{s''z}) < h_\ell(T'')$$

holds. Thus the best candidate path must be selected under the paths $P_{s''z}$ with

$$\ell(P_{s''z}) + \ell^+(P_{s''z}) \geq h_\ell(T'').$$

We define

$$\hat{Z}(T'') = \{z \in Z(T'') : \ \ell(P_{s''z}) + \ell^+(P_{s''z}) \geq h_\ell(T'')\}.$$

For every $z \in \hat{Z}(T'')$, we first increase the length of $P_{s''z}$ by the amount

$$\delta_z(T'') = \min\{\ell^+(P_{s''z}), h_\ell(T') - h_\ell(T'')\},$$

and then we reduce the height of T' by the amount

$$\delta_z(T') = h_\ell(T') - h_\ell(T'') - \delta_z(T'') \tag{5}$$

which means that the edges of T' have to be changed in total by $C(\delta_z(T'))$. In the case that the height of T' cannot be reduced by the amount $\delta_z(T')$, we set $C(\delta_z(T')) = +\infty$. Therefore, the minimum length change for fulfilling $h_{\tilde{\ell}}(T') = \tilde{\ell}(P_{s''z})$ is

$$C_z = C(\delta_z(T')) + \delta_z(T'').$$

A path $P_{s''z^0}$ with minimum value C_{z_0} is selected as best candidate path. For determining C_z we need to know $C(\delta_z(T'))$ for all $z \in \hat{Z}(T'')$. These *reduction costs* can be found in the following way:

1. Compute the height-reduction cost function $C(\delta)$ for tree T' by applying procedure HRCF of Subsection 3.2.
2. Compute the amounts $\delta_z(T')$ for all $z \in \hat{Z}(T'')$ according to (5) and sort these numbers non-decreasingly. Then compute $C(\delta_z(T'))$. In case that $\delta_z(T') > \delta_{\max}$, set $C(\delta_z(T')) = +\infty$ ($\delta_{\max}$ is the maximum feasible amount by which the height of T' can be reduced).

If $C_{z^0} = +\infty$ then two rooted trees T' and T'' cannot be balanced and the height balancing problem is infeasible. Otherwise an optimal solution of the problem is obtained by increasing the path length $\ell(P_{s''z^0})$ by the amount $\delta_{z^0}(T'')$ and by reducing the height of tree T' by the amount $\delta_{z^0}(T')$. The latter is performed by applying the tree height reduction method of Subsection 3.1. The edge length on the best candidate path $P_{s''z^0}$ can be increased in an arbitrary way subject to the modification bounds.

Assume that $n = |E(T')| + |E(T'')|$. We are now going to determine the time complexity of the solution method discussed above: The subset Z_0 can be found in $O(n)$ time if we traverse the tree T'' in a depth-first manner. The amounts $\delta_z(T')$, $z \in \hat{Z}(T'')$, are computed in $O(n)$ time. The height-reduction cost function $C(\delta)$ can be determined in at most $O(n \log n)$ time. Computing $C(\delta_z(T'))$, $z \in \hat{Z}(T'')$, takes at most $O(n \log n)$ time, since we sort the values $\delta_z(T')$ in advance. Increasing the length $\ell(P_{s''z^0})$ by the amount $\delta_{z^0}(T'')$ and reducing the height $h_\ell(T')$ by the amount $\delta_{z^0}(T')$ requires $O(n)$ time. Therefore, the overall time complexity is bounded by $O(n \log n)$. Thus we get the following theorem:

Theorem 4.1 *The problem of balancing heights of two rooted trees T' and T'' with uniform cost coefficients can be solved in $O(n \log n)$ time where n is the total number of edges in T' and T''.*

In the next two sections, we use the height balancing of two rooted trees as a subproblem for solving the uniform-cost inverse absolute and vertex 1-center problems.

5 Uniform-cost inverse absolute 1-center location problem

Consider the two subtrees L and R (introduced in Section 2) of the given tree T. We root L and R in vertex s. If $h_\ell(L) = h_\ell(R)$, then s is the midpoint of all longest paths in T and therefore the wanted absolute 1-center of tree T. Otherwise, we have to change some edge lengths on T in order to balance the heights of tree L and tree R.

If L is a path we can directly apply the height balancing procedure of Section 4 in order to balance the trees L and R. Now let us assume that L is not a path. Let $s_1 \in V(L)$ be the nearest vertex to s with $\deg(s_1) > 2$. If

$$\sum_{e \in E(P_{ss_1})} \ell^-(e) < \ell(P_{ss_1}) \tag{6}$$

or, if

$$h_\ell(L) - h_\ell(R) < \ell(P_{ss_1}) \tag{7}$$

holds, then any feasible edge length modification in tree L keeps vertex s_1 apart from vertex s. With other words, the root s remains always incident with just one edge in tree L. In this case we say that *no topology change* occurs on the given tree T. As an immediate consequence of Lemma 2.2 we get:

Lemma 5.1 *If at least one of the inequalities* (6) *and* (7) *holds, then it is sufficient to balance the heights of the subtrees L and R both rooted in the prespecified vertex s in order to solve the uniform-cost inverse absolute* 1*-center location problem on tree T.*

Moreover, we get:

Lemma 5.2 *In case that neither inequality* (6) *nor* (7) *holds the solution of the uniform-cost inverse absolute* 1*-center location problem on the tree T can be reduced to the solution of the same problem on tree T^1 which is obtained from T by reducing the length $\ell(P_{ss_1})$ to* 0*.*

Proof. Reducing $\ell(P_{ss_1})$ to 0 causes cost $\delta = \ell(P_{ss_1})$ and reduces the gap $h(L) - h(R)$ by δ. This is best possible. ■

Let P_{sz^*} be a longest path from root s to the leaves $z \in Z(L)$. If $L \neq P_{sz^*}$, let $s_1, \ldots, s_k$ be the set of all vertices on P_{sz^*} such that $\deg(s_i) > 2$, $i = 1, \ldots, k$, and $\ell(P_{ss_{i-1}}) < \ell(P_{ss_i})$ with $s_0 = s$. We denote $T^0 = T$ and define T^i as that tree which is obtained from T by reducing the path length $\ell(P_{ss_i})$ to zero. Let L^i and R^i be the two subtrees of T^i such that

$$L^i \cap R^i = s, \quad s = s_j, \; j = 0, ..., i,$$

and L^i contains the longest path P_{sz^*} and has at most one edge with endpoint s whereas subtree R^i contains all leaves in $Z(T^i) \backslash Z(L^i - s)$. Both subtrees L^i and R^i

are rooted in $s = s_0 = ... = s_i$. Based on Lemmas 5.1 and 5.2 we know that there exists an index

$$i_0 \in \{0, ..., k\}$$

such that an optimal solution of the uniform-cost inverse absolute 1-center location problem on tree T can be found by first reducing the length $\ell(P_{ss_{i_0}})$ to zero and then balancing the heights of the two subtrees L^{i_0} and R^{i_0}.

In order to solve the uniform-cost inverse absolute 1-center location problem on T, we have to determine the trees T^i, subtrees L^i, R^i and the heights $h_\ell(L^i)$, $h_\ell(R^i)$ for all $i = 0, \ldots, i_0$. We first determine the subtrees $L = L^0$ and $R = R^0$ of T and associate the label $h(e)$ to every edge $e \in E(L)$ ($h(e)$ denotes the height of subtree L^e beginning with edge e). These labels are computed in a breadth-first approach from the leaves to root s. The subtrees subtrees L^i, R^i and their heights for $i = 1, ..., i_0$ can be obtained by calling the following Procedure STH(i).

Procedure STH(i):

(i) Reduce the length of path $P_{s_{i-1}s_i}$ to zero in order to obtain tree T^i rooted in $s = s_j$, $j = 0, ..., i$, from T^{i-1}.

(ii) Let $e_1, ..., e_{t_i}$ be the set of edges leaving vertex s_i in the rooted tree L. W.l.o.g. assume that $P_{s_i z^*} \subseteq L^{e_1}$. Thus

$$L^i = L^{e_1} \quad \text{and} \quad R^i = \bigcup_{j=2}^{t_i} L^{e_j} \cup R^{i-1}.$$

(iii) The heights of L^i and R^i are given by

$$h_\ell(L^i) = h(e_1),$$
$$h_\ell(R^i) = \max\{h(e_j) \text{ for } j = 2, ..., t_i,\ h_\ell(R^{i-1})\}.$$

We immediately get

Lemma 5.3 *The trees T^i, L^i, R^i, $i = 0, \ldots, i_0$, and the corresponding heights $h_\ell(L^i)$, $h_\ell(R^i)$, $i = 0, \ldots, i_0$, are determined in $O(n)$ overall time.*

Based on the previous considerations we can state Algorithm 1 for solving the uniform-cost inverse absolute 1-center location problem.

We know that the partitioning of T into L and R takes $O(n)$ time. During the execution of Algorithm 1 we solve the height balancing problem in $O(n \log n)$ time according to Theorem 4.1. Moreover, based on Lemma 5.3 we can perform Procedure STH(i) for $i = 1, ..., k$, $k < n$, in $O(n)$ overall time. Thus the time complexity of Algorithm 1 is bounded by $O(n \log n)$. Altogether we get the following theorem:

Theorem 5.4 *The uniform-cost inverse absolute 1-center location problem with edge length variations can be solved in $O(n \log n)$ time on a tree network by applying Algorithm 1 where n is the number of edges on the given tree.*

Algorithm 1 determines an optimal solution of the uniform-cost inverse absolute 1-center location problem on a tree network T, vertex s being the absolute 1-center.

begin
partition T into subtrees L and R and root both in vertex s;
label a longest path P_{sz^*} from s to leaves of L;
if $L = P_{sz^*}$ **then**
balance the heights of L and R;
an optimal solution of the original problem is reached, stop;
else
let $s_1, \ldots, s_k \in V(P_{sz^*})$ such that $\deg(s_i) > 2$, $i = 1, \ldots, k$;
set $s_0 = s$, $L^0 = L$, $R^0 = R$ and $i = 0$;
while $h_\ell(L^i) - h_\ell(R^i) \geq \ell(P_{s_i s_{i+1}}) = \ell^-(s_i s_{i+1})$ **do**
set

$$y'(e) = \ell(e) \quad \text{for all } e \in E(P_{s_i s_{i+1}});$$

recall Procedure STH$(i+1)$ to determine subtrees L^{i+1}, R^{i+1} and heights $h_\ell(L^{i+1})$, $h_\ell(R^{i+1})$;
update $i = i + 1$;
end while
find an optimal solution vector (x_i, y_i) of the height balancing problem on the subtrees L^i and R^i; an optimal solution of the original problem is given by

$$x^*(e) = \begin{cases} 0 & \text{for all } \ e \in E(P_{ss_i}), \\ x_i(e) & \text{otherwise,} \end{cases} \quad , \quad y^*(e) = \begin{cases} y'(e) & \text{for all } \ e \in E(P_{ss_i}), \\ y_i(e) & \text{otherwise.} \end{cases}$$

end if
end

6 Uniform-cost inverse vertex 1-center location problem

This section considers the uniform-cost inverse *vertex* 1-center location problem on the given tree network T. Recently, Yang and Zhang [17] suggested a solution method for solving this problem with uniform cost coefficients in $O(n)$ time in the case that the modified edge lengths $\tilde{\ell}$ all remain positive, i.e., when

$$\ell^-(e) < \ell(e) \quad \text{for all } e \in E(T). \tag{8}$$

Below we show by a counter example that their solution approach does not work correctly in general. Then we propose a novel solution method which relies on balancing the heights of two subtrees $\hat{L}$ and $\hat{R}$ of T.

Counter example to Yang and Zhang's solution method:

Consider the tree network T given in Figure 3. Assume that T has the edge lengths $\ell(e_i)$ and the modification bounds $\ell^-(e_i)$ and $\ell^+(e_i)$ as given in Table 1.

If we apply Yang and Zhang's solution method described in Section 4 of [17], then we get P_{sz_2} as the best candidate path for increasing edge lengths. This leads

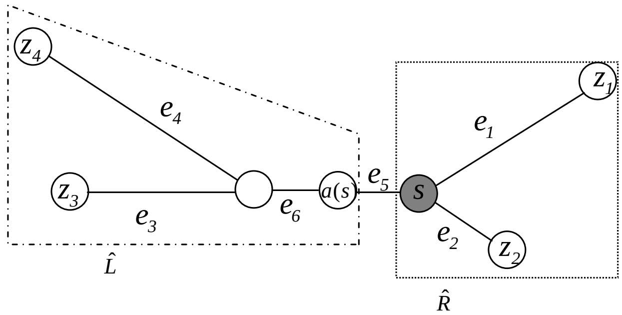

Figure 3: Illustration of the tree T applied in the counter example

i	$\ell(e_i)$	$\ell^-(e_i)$	$\ell^+(e_i)$
1	8	3	1
2	3	2	9
3	6	4	5
4	11	7	2
5	2	1	8
6	2	1	4

Table 1: edge lengths and modification bounds assigned to tree T of Figure 3

to a feasible solution with objective value $C_{z_2} = 10$. But one can easily observe that by increasing the length $\ell(e_1)$ by the amount 1 and reducing the lengths $\ell(e_4)$ and $\ell(e_6)$ by the amounts 3 and 1, respectively, vertex s becomes a vertex 1-center of T with respect to the new edge lengths. This solution incurs the total cost $C_{z_1} = 5$. Thus we have found a better feasible solution! This implies that Yang and Zhang's method developed for the uniform-cost inverse vertex 1-center location problem is not CORRECT in general.

In order to solve the inverse vertex 1-center location problem on T with uniform cost correctly we proceed as follows. Let us first assume that inequalities (8) hold in the given tree network T. Consider the two subtrees $\hat{L}$ and $\hat{R}$ of T (introduced in Section 2) which are rooted in the vertices $a(s)$ and s, respectively. Lemma 2.3 immediately implies

Lemma 6.1 *In order to solve the uniform-cost inverse vertex* 1*-center location problem on an unweighted tree T, it is sufficient to balance the heights of the subtrees $\hat{L}$ and $\hat{R}$ provided that the condition* (8) *is satisfied.*

By applying the solution method developed in Section 4 we balance the heights of the subtrees $\hat{L}$ and $\hat{R}$. If these two subtrees cannot be balanced, then the original problem is infeasible. Otherwise an optimal solution is obtained.

The subtrees $\hat{L}$ and $\hat{R}$ can be obtained in $O(n)$ time if tree T is traversed in a depth-first manner. On the other hand, according to Theorem 4.1 the balancing the heights of $\hat{L}$ and $\hat{R}$ (with uniform modification costs) is done in $O(n \log n)$ time. Therefore, we conclude

Theorem 6.2 *The uniform-cost inverse vertex* 1*-center location problem with edge length variations can be solved in* $O(n \log n)$ *time on a tree network (with* n *edges) provided that the modified edge lengths remain positive.*

If we drop condition (8), then the general model (with *uniform cost coefficients*) can be solved in $O(r_v n \log n)$ time by applying the solution method developed in [1] for the general inverse vertex 1-center location problem (with *arbitrary cost coefficients*), provided that we balance the heights of the trees with an $O(n \log n)$-time solution method as introduced in this article. Note that the parameter r_v, the so called *minimal repeat number* of T, is bounded by $\lceil n/2 \rceil$ and has explicity been defined in [1].

Altogether we get

Theorem 6.3 *The general uniform-cost inverse vertex* 1*-center location problem with edge length variations is solvable in* $O(r_v n \log n)$ *time on a tree network where* n *is the number of edges and* r_v *is the minimal repeat number (defined in [1]) of the given tree.*

7 Conclusions

In this article we investigated the uniform-cost inverse absolute and vertex 1-center location problems with edge length variations on a tree network. We first developed a new combinatorial $O(n \log n)$ time solution method for the height balancing problem with uniform cost coefficients. Applying this method we get algorithms for the uniform-cost inverse absolute (and vertex) 1-center location problems with fast time complexities.

In the case that the edge lengths of the given tree T are only permitted to be reduced the uniform-cost inverse absolute and vertex 1-center location problems are solved in $O(n)$ and $O(nr_v)$ times, respectively. For balancing two subtrees it suffices in these special models to reduce the height of the highest subtree to the height of the second highest by applying the tree height reduction method of Subsection 3.1.

Acknowledgments

The first author gratefully acknowledges financial support by the NAWI-Project under the grant F-NW-MATH-05. This work was also supported by the Austrian Science Fund (FWF) Project P18918-N18.

References

[1] B. Alizadeh and R.E. Burkard, Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees, *Technical Report 2009-09*, Graz University of Technology, 2009.

[2] B. Alizadeh , R.E. Burkard and U. Pferschy, Inverse 1-center location problems with edge length augmentation on trees, *Technical Report 2009-07*, Graz University of Technology, 2009.

[3] F. Baroughi-Bonab, R.E. Burkard and B. Alizadeh, Inverse median location problems with variable coordinates, *Technical Report 2009-08*, Graz University of Technology, 2009.

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

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

[6] R.E. Burkard, C. Pleschiutschnig and J. Zhang, The inverse 1-median problem on a cycle, *Discrete Optimization*, **5** (2007) 242–253.

[7] 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.

[8] M.S. Daskin, *Network and Discrete Location: Modeles, Algorithms and Applications*, John Wiley, New York, 1995.

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

[10] M. Galavii, *Inverse* 1-*Median Problems*, Ph.D. Thesis, Institute of Optimization and Discrete Mathematics, Graz University of Technology, Graz, Austria, 2008.

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

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

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

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

[15] B.P. Mirchandani, R.L. Francis, *Discrete Location Theory*, John Wiley, New York, 1990.

[16] C. Pleschiutschnig, *Inverse Median Problems*, Ph.D. Thesis, Institute of Optimization and Discrete Mathematics, Graz University of Technology, Graz, Austria, 2005.

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