

AN ADJUSTABLE ROBUST APPROACH FOR A 1-MEDIAN LOCATION PROBLEM ON A TREE

Maiko Shigeno
University of Tsukuba

(Received November 19, 2006; Revised September 21, 2007)

Abstract A robust approach to solving discrete optimization problems with uncertain data has recently been extensively studied and extended. This paper presents a new robust optimization model, called the adjustable robust optimization model, where a small perturbation for a solution is permitted after the uncertain parameters are realized. We apply this model to a 1-median location problem under uncertainty and give a polynomial time algorithm to determine robust solutions in the case of a tree graph.

Keywords: Discrete optimization, robust optimization, location problem

1. Introduction

Mathematical programming problems with uncertainty in input data have attracted significant attention, because uncertainty affects a wide range of decisions that managers, engineers, and other decision makers must make. One of the principal methods that have been proposed to address data uncertainty is a worst-case approach. The worst-case approach is looking for a solution that performs reasonably well for all possible input data. During the last decade, this approach has been developed as a *robust optimization*.

For discrete optimization, Kouvelis and Yu [14] proposed a framework for robust optimization. Their robust optimization uses a scenario based approach to represent the input data uncertainty. Two natural methods for describing the set of all possible scenarios have been considered: the discrete-scenario case and the interval data case. In the discrete-scenario case, all possible scenarios are assumed to be finite and are described by explicitly listing all of the scenarios. In the interval data case, it is assumed that for each numerical parameter, only the lower and upper bounds for the value of this parameter are known, and the parameter can take on any value between these bounds, regardless of the value taken by other parameters. That is to say, the set of scenarios is given by the Cartesian product of the intervals of uncertainty for the parameters.

One of the main themes of robust discrete optimization is the treatment of well-known discrete optimization with uncertainty in input data [1–6, 8, 9, 11, 15]. Even simple discrete optimization problems tend to become intractable in robust optimization. Indeed, it has been observed that many polynomially solvable discrete optimization problems such as minimum spanning tree problems and shortest path problems become NP-hard [6, 14]. This observation is not surprising because the existence of several scenarios is deeply related to multi-criteria optimization. Hence, from an algorithmic point of view, only extremely simple problems such as 1-median/center location and economic order quantity have been treated on robust discrete optimization.

The purpose of the present paper is to introduce adjustable robustness on discrete op-

timization and to investigate its applicability within the context of a well-known discrete optimization problem, that is, a 1-median location problem on a tree.

2. Robust Optimization

Let S be a set of all potentially realizable input data scenarios, and X be a set of all solutions. We use the notations D^s and X_s to denote the instance of the input data and the set of all feasible solutions, respectively, corresponding to a scenario $s \in S$. Suppose that the cost of a solution $x \in X_s$ with a scenario $s \in S$ is evaluated by a function $f(x, D^s)$. For a scenario $s \in S$, the deterministic optimization problem can be written as $\min_{x \in X_s} f(x, D^s)$.

There are two most widely considered measures of robustness. The first is *absolute robustness*, which minimizes the cost among all feasible solutions over all realizable input data scenarios, i.e.,

$$\min_{x \in \bigcap_{s \in S} X_s} \max_{s \in S} f(x, D^s).$$

The second is *deviation robustness*, which exhibits the best worst case deviation from optimality among all feasible solutions over all realizable input, i.e.,

$$\min_{x \in \bigcap_{s \in S} X_s} \max_{s \in S} (f(x, D^s) - f(x_s^*, D^s)),$$

where x_s^* is an optimal solution for $\min_{x \in X_s} f(x, D^s)$. This problem is also called the *minmax regret optimization problem*.

In the above robustness, the obtained solution is fixed before an actual scenario realization is known. Recently, Ben-Tal et al.[7] developed linear programs with uncertain parameters, where some of the variables must be determined before the realization of the uncertain parameter, while the remainder of the variables can be chosen after the realization. Although this adjustable solution is realistic, all of the variables are usually treated uniformly in discrete optimization. If anything, several practical occasions allow a small perturbation for a solution after a scenario is realized. So we propose a new adjustable robust optimization model. Let us introduce an exchange cost $c^s(x, y)$ from $x \in X$ to $y \in X$ according to each scenario $s \in S$. The proposed robustnesses are formulated as follows:

$$(\text{ARO})_A \quad \min_{x \in X} \max_{s \in S} \min_{y \in X_s} (f(y, D^s) + c^s(x, y))$$

$$(\text{ARO})_D \quad \min_{x \in X} \max_{s \in S} \min_{y \in X_s} (f(y, D^s) - f(x_s^*, D^s) + c^s(x, y)).$$

If exchange costs $c^s(x, x)$ for all $s \in S$ and $x \in X$ are zero and $c^s(x, y)$ are sufficiently large for all $s \in S$ and $x, y \in X$ with $x \neq y$, the problems $(\text{ARO})_A$ and $(\text{ARO})_D$ coincide with the original absolute robust optimization and deviation robust optimization, respectively. Note that the exchange cost can also represent adjustable/non-adjustable variables in the model due to Ben-Tal et al. [7].

3. Adjustable Robust 1-median Location on a Tree

Robust simple facility location problems have recently received increasing interest [1, 3–5, 8, 9, 15]. In this section, we apply our adjustable robust model to a simple location problem, namely, a 1-median location problem on a tree, which is one of a few polynomially solvable robust optimization problems.

3.1. Problem formulation

Let $G = (V, T)$ be a tree graph with vertex set V and edge set T . Each scenario $s \in S$ is given by a vertex weight $w^s : V \rightarrow \mathbb{R}$ and a positive edge length $d^s : T \rightarrow \mathbb{R}_{>0}$. Some of the vertices may have negative weights, as in Burkard and Krarup's model [10], which locates "friendly" as well as "obnoxious" facilities. A *point* of the graph corresponds either to a vertex or to any inner point of an edge. With a scenario $s \in S$, the cost of serving all of the vertices of the tree from a point y is given by

$$f(y, D^s) = \sum_{v \in V} w^s(v) d^s(v, y),$$

where $d^s(v, y)$ is the distance between points v and y , that is, the length of the unique path from v to y in G under a scenario s . A point minimizing the cost $f(y, D^s)$ is called *1-median* with respect to a scenario $s \in S$. For an exchange cost, an edge cost $c^s : T \rightarrow \mathbb{R}$ is also given for each scenario $s \in S$. The exchange cost $c^s(x, y)$ from a point x to a point y is defined by the length of the unique path between x and y according to the cost c^s . We assume that for any inner point x of an edge (u, v) , the ratios of the length of the segment between u and x to the length of edge (u, v) , that is,

$$\frac{d^s(u, x)}{d^s(u, v)} \quad \text{and} \quad \frac{c^s(u, x)}{c^s(u, v)}$$

remain constant for every scenario. Then, $(\text{ARO})_A$ for a 1-median location on a tree is formulated as

$$\min_{x \in G} \max_{s \in S} \min_{y \in G} (f(y, D^s) + c^s(x, y)),$$

where the notation $x \in G$ indicates that x is a point of G .

3.2. Algorithm

According to a location point x and a scenario $s \in S$, an *adjustment location point* y , denoted by y_x^s , is a point attaining $\min_{y \in G} (f(y, D^s) + c^s(x, y))$. We first establish the properties of y_x^s . For an edge $(p, q) \in T$, let $G_{p \leftarrow q}$ and $G_{q \leftarrow p}$ be the two trees containing vertices p and q , respectively, of the forest $G \setminus \{(p, q)\}$. Notations $\tilde{y}_{p \leftarrow q}^s$ and $\tilde{y}_{q \leftarrow p}^s$ stand for location points attaining $\min_{y \in G_{p \leftarrow q}} (f(y, D^s) + c^s(p, y))$ and $\min_{y \in G_{q \leftarrow p}} (f(y, D^s) + c^s(q, y))$, respectively.

Lemma 3.1 *For a location point x on an edge $(p, q) \in T$ and a scenario $s \in S$, an adjustment location point y_x^s accords to $\tilde{y}_{p \leftarrow q}^s$, $\tilde{y}_{q \leftarrow p}^s$ or x itself.*

Proof. When $y_x^s \in G_{p \leftarrow q}$, we have

$$\begin{aligned} f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(x, \tilde{y}_{p \leftarrow q}^s) &\geq f(y_x^s, D^s) + c^s(x, y_x^s) \\ &= f(y_x^s, D^s) + c^s(p, y_x^s) + c^s(x, p) \\ &\geq \min_{y \in G_{p \leftarrow q}} (f(y, D^s) + c^s(p, y)) + c^s(x, p) \\ &= f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(p, \tilde{y}_{p \leftarrow q}^s) + c^s(x, p) = f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(x, \tilde{y}_{p \leftarrow q}^s). \end{aligned}$$

Thus, $f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(x, \tilde{y}_{p \leftarrow q}^s) = f(y_x^s, D^s) + c^s(x, y_x^s)$ holds. Analogously, we have $f(\tilde{y}_{q \leftarrow p}^s, D^s) + c^s(x, \tilde{y}_{q \leftarrow p}^s) = f(y_x^s, D^s) + c^s(x, y_x^s)$ when $y_x^s \in G_{q \leftarrow p}$. Finally, we consider the case in which y_x^s is an inner point of the edge (p, q) . If we locate a point y between x and q , then

$$f(y, D^s) + c^s(x, y) = f(x, D^s) + \left(\sum_{v \in V(G_{p \leftarrow q})} w^s(v) - \sum_{v \in V(G_{q \leftarrow p})} w^s(v) \right) d^s(x, y) + c^s(p, q) \frac{d^s(x, y)}{d^s(p, q)},$$

where $V(G_{p \leftarrow q})$ (resp. $V(G_{q \leftarrow p})$) is the vertex set of $G_{p \leftarrow q}$ (resp. $G_{q \leftarrow p}$). Hence, if $\sum_{v \in V(G_{p \leftarrow q})} w^s(v) - \sum_{v \in V(G_{q \leftarrow p})} w^s(v) < -c^s(p, q)/d^s(p, q)$, then y is moved to the vertex q . Similarly, if $\sum_{v \in V(G_{q \leftarrow p})} w^s(v) - \sum_{v \in V(G_{p \leftarrow q})} w^s(v) < -c^s(p, q)/d^s(p, q)$, then y is moved to the vertex p . Otherwise, y does not move from x . Thus, in this case, y_x^s is x itself. Combining the above results, we verify that y_x^s coincides with $\tilde{y}_{p \leftarrow q}^s$, $\tilde{y}_{q \leftarrow p}^s$, or x itself. \square

The following fact is derived from the proof of Lemma 3.1.

Corollary 3.1 For any $(p, q) \in T$ and $s \in S$, a location point $\tilde{y}_{p \leftarrow q}^s$ can be chosen from $\{\tilde{y}_{v \leftarrow p}^s \mid (v, p) \in T, v \neq q\} \cup \{p\}$. \square

Let us define $Z^s(x) = \min_{y \in G} (f(y, D^s) + c^s(x, y))$ and $Z_{pq}^s(\cdot)$ as the restriction of $Z^s(\cdot)$ on an edge $(p, q) \in T$, that is to say, it is the function $Z^s(\cdot)$ whose argument x can be taken only on the edge (p, q) . It follows from Lemma 3.1 that

$$Z_{pq}^s(x) = \min \left\{ f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(x, \tilde{y}_{p \leftarrow q}^s), f(\tilde{y}_{q \leftarrow p}^s, D^s) + c^s(x, \tilde{y}_{q \leftarrow p}^s), f(x, D^s) \right\} \quad (3.1)$$

holds for each x on an edge (p, q) and $s \in S$. By defining $\tilde{F}_{p \leftarrow q}^s = f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(p, \tilde{y}_{p \leftarrow q}^s)$, the function $Z_{pq}^s(x)$ can be rewritten by

$$\min \left\{ \tilde{F}_{p \leftarrow q}^s + c^s(x, p), \tilde{F}_{q \leftarrow p}^s + c^s(x, q), f(p, D^s) + \left(\sum_{v \in V(G_{p \leftarrow q})} w^s(v) - \sum_{v \in V(G_{q \leftarrow p})} w^s(v) \right) d^s(p, x) \right\}.$$

Let $Z(x) = \max_{s \in S} Z^s(x)$. The proposed algorithm, described below, tries to place x on each edge $(p, q) \in T$ such that $Z(x)$ is minimized, and finally outputs x minimizing the total cost.

- A1: compute $f(v, D^s)$ for all $v \in V$ and $s \in S$, and keep $\sum_{z \in V(G_{p \leftarrow q})} w^s(z) - \sum_{z \in V(G_{q \leftarrow p})} w^s(z)$ for all $(p, q) \in T$ and $s \in S$.
A2: find $\tilde{y}_{p \leftarrow q}^s$, $\tilde{y}_{q \leftarrow p}^s$, $\tilde{F}_{p \leftarrow q}^s$ and $\tilde{F}_{q \leftarrow p}^s$ for each $(p, q) \in T$ and $s \in S$.
A3: value = ∞ , $x^* = \emptyset$
A4: **for** every edge $(p, q) \in T$ **do**
A5: solve $\min\{Z(x) \mid x \text{ is on } (p, q)\}$.
A6: **if** value $> \min\{Z(x) \mid x \text{ is on } (p, q)\}$ **then**
A7: update value = $\min\{Z(x) \mid x \text{ is on } (p, q)\}$ and x^* to its optimal x .
A8: **end**

Lines A1 and A2 are a preprocessor to establish $Z^s(x)$. By maintaining the results of the preprocessor, we can evaluate $Z^s(x)$ for each $x \in G$ in $O(1)$. Burkard and Dollani [8] described an $O(|V|)$ algorithm for computing $f(y, D^s)$ for all $y \in V$ and for obtaining values $\sum_{z \in V(G_{p \leftarrow q})} w^s(z)$ and $\sum_{z \in V(G_{q \leftarrow p})} w^s(z)$ for all $(p, q) \in T$. Hence, Line A1 can be performed in $O(|S||V|)$ time. We next show that Line A2 is also performed in the same time. Intuitively, the following property is obvious because both $f(\cdot, D^s)$ and $c^s(p, \cdot)$ (resp. $c^s(q, \cdot)$) are linear functions on each edge.

Lemma 3.2 For any $(p, q) \in T$ and $s \in S$, a point $\tilde{y}_{p \leftarrow q}^s$ (resp. $\tilde{y}_{q \leftarrow p}^s$) can be located at a vertex.

Proof. Suppose that a location point $\tilde{y}_{p \leftarrow q}^s$ is on an edge (p', q') , where the path between $\tilde{y}_{p \leftarrow q}^s$ and p contains the vertex q' . Then we have

$$\begin{aligned} & f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(p, \tilde{y}_{p \leftarrow q}^s) \\ &= f(q', D^s) + c^s(p, q') + \left(\sum_{v \in V(G_{q' \leftarrow p'})} w^s(v) - \sum_{v \in V(G_{p' \leftarrow q'})} w^s(v) + \frac{c^s(p', q')}{d^s(p', q')} \right) d^s(q', \tilde{y}_{p \leftarrow q}^s). \end{aligned}$$

If $\sum_{v \in V(G_{q' \leftarrow p'})} w^s(v) - \sum_{v \in V(G_{p' \leftarrow q'})} w^s(v) + c^s(p', q')/d^s(p', q') > 0$, then we obtain $f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(p, \tilde{y}_{p \leftarrow q}^s) \geq f(q', D^s) + c^s(p, q')$. Otherwise, it holds that $f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(p, \tilde{y}_{p \leftarrow q}^s) \geq f(p', D^s) + c^s(p, p')$. Hence, we can choose a location point $\tilde{y}_{p \leftarrow q}^s$ at p' or q' . \square

The following procedure computes location points $\tilde{y}_{p \leftarrow q}^s$ and $\tilde{y}_{q \leftarrow p}^s$ and values $\tilde{F}_{p \leftarrow q}^s$ and $\tilde{F}_{q \leftarrow p}^s$ for all $(p, q) \in T$ with respect to a scenario $s \in S$. The parent and the set of all children of the vertex v in the tree G rooted at a specified vertex r are denoted by $P(v)$ and $C(v)$, respectively.

(Phase I)

- 1: $\tilde{F}_{v \leftarrow P(v)}^s = f(v, D^s)$ and $\tilde{y}_{v \leftarrow P(v)}^s = v$ for all leaves v
- 2: add all leaves to V^{up}
- 3: **while** $V^{\text{up}} \neq V \setminus \{r\}$ **do**
- 4: choose $p \notin V^{\text{up}}$, where $C(p) \subseteq V^{\text{up}}$
- 5: $\hat{v} = \arg \min \{ \tilde{F}_{v \leftarrow p}^s + c^s(v, p) \mid v \in C(p) \}$
- 6: **if** $\tilde{F}_{\hat{v} \leftarrow p}^s + c^s(\hat{v}, p) < f(p, D^s)$
- 7: **then** $\tilde{F}_{p \leftarrow P(p)}^s = \tilde{F}_{\hat{v} \leftarrow p}^s + c^s(\hat{v}, p)$ and $\tilde{y}_{p \leftarrow P(p)}^s = \tilde{y}_{\hat{v} \leftarrow p}^s$
- 8: **else** $\tilde{F}_{p \leftarrow P(p)}^s = f(p, D^s)$ and $\tilde{y}_{p \leftarrow P(p)}^s = p$
- 9: add p to V^{up}
- 10: **end**

(Phase II)

- 11: set $p = r$, $F(p) = f(p, D^s)$ and $L(p) = p$.
- 12: **if** $C(p) = \emptyset$ **then** go to Line 22
- 13: $\hat{v} = \arg \min \{ \tilde{F}_{v \leftarrow p}^s + c^s(v, p) \mid v \in C(p) \}$
- 14: **if** $\tilde{F}_{\hat{v} \leftarrow p}^s + c^s(\hat{v}, p) < F(p)$
- 15: **then** $\tilde{F}_{p \leftarrow v}^s = \tilde{F}_{\hat{v} \leftarrow p}^s + c^s(\hat{v}, p)$ and $\tilde{y}_{p \leftarrow v}^s = \tilde{y}_{\hat{v} \leftarrow p}^s$ for $v \in C(p) \setminus \{\hat{v}\}$
- 16: **else** $\tilde{F}_{p \leftarrow v}^s = F(p)$ and $\tilde{y}_{p \leftarrow v}^s = L(p)$ for $v \in C(p) \setminus \{\hat{v}\}$
- 17: **if** $C(p) \setminus \{\hat{v}\} = \emptyset$ **then** go to Line 22
- 18: $\check{v} = \arg \min \{ \tilde{F}_{v \leftarrow p}^s + c^s(v, p) \mid v \in C(p) \setminus \{\hat{v}\} \}$
- 19: **if** $\tilde{F}_{\check{v} \leftarrow p}^s + c^s(\check{v}, p) < F(p)$
- 20: **then** $\tilde{F}_{p \leftarrow \hat{v}}^s = \tilde{F}_{\check{v} \leftarrow p}^s + c^s(\check{v}, p)$ and $\tilde{y}_{p \leftarrow \hat{v}}^s = \tilde{y}_{\check{v} \leftarrow p}^s$
- 21: **else** $\tilde{F}_{p \leftarrow \hat{v}}^s = F(p)$ and $\tilde{y}_{p \leftarrow \hat{v}}^s = L(p)$
- 22: add p to V^{down}
- 23: **while** $V^{\text{down}} \neq V$ **do**
- 24: choose $(p, q) \in T$, where $p \notin V^{\text{down}}$ and $q \in V^{\text{down}}$
- 25: **if** $\tilde{F}_{q \leftarrow p}^s + c^s(p, q) < f(p, D^s)$
- 26: **then** $F(p) = \tilde{F}_{q \leftarrow p}^s + c^s(p, q)$ and $L(p) = \tilde{y}_{q \leftarrow p}^s$
- 27: **else** $F(p) = f(p, D^s)$ and $L(p) = p$.
- 28: perform Lines 12–22.
- 29: **end**

The correctness of this procedure is based on Corollary 3.1 and Lemma 3.2. Phase I finds a location point for each vertex in its descendants. Phase II returns a location point, which does not contain its descendants. During each phase of the above procedure, each vertex is visited once and constant time is spent for the computations. We now obtain the following result.

Lemma 3.3 *Location points $\tilde{y}_{p \leftarrow q}^s$ and $\tilde{y}_{q \leftarrow p}^s$ and location costs $\tilde{F}_{p \leftarrow q}^s$ and $\tilde{F}_{q \leftarrow p}^s$ for all edges $(p, q) \in T$ with respect to a scenario $s \in S$ can be found in $O(|V|)$ time. \square*

Thus, the preprocess in Lines A1 and A2 can be performed in $O(|S||V|)$ time.

Finally, we discuss how to solve $\min\{Z(x) \mid x \text{ is on } (p, q)\}$. Note that the function $Z(\cdot)$ restriction to an edge $(p, q) \in T$, denoted by $Z_{pq}(\cdot)$, is a piecewise linear, but not convex function. Thus, we must check all breakpoints of $Z_{pq}(\cdot)$ in order to find the minimizer. Let $[a, b]$ be a subinterval of edge (p, q) such that a and b are two distinct breakpoints of $Z_{pq}^s(\cdot)$ for some $s \in S$ and there is no breakpoint between a and b . Since the function $Z_{pq}^s(\cdot)$ has at most two breakpoints, we have at most $2|S| + 1$ subintervals, and we can enumerate all subintervals in an edge in $O(|S| \log |S|)$ time by sorting all of the breakpoints. In order to determine a minimizer of $Z(\cdot)$ in an subinterval, we can use the procedure developed by Kouvelis and Yu in the case of the robust 1-median problem on a tree [14], because $Z^s(\cdot)$ is a linear function in each subinterval. Since their procedure runs in $O(|S|)$ time, we can solve $\min\{Z(x) \mid x \text{ is on } (p, q)\}$ in $O(|S|^2)$ time. Thus, we obtain the following conclusion.

Theorem 3.1 $(\text{ARO})_A$ *for the 1-median location problem on a tree can be solved in $O(|S|^2|V|)$ time. \square*

We can apply this result to $(\text{ARO})_D$ for a 1-median location on a tree, which can be rewritten by $\min_{x \in G} \max_{s \in S} (Z^s(x) - f(x_s^*, D^s))$. A 1-median x_s^* for each scenario $s \in S$ can be found in $O(|V|)$ time [10]. The above-mentioned procedure to solve $\min\{Z(x) \mid x \text{ is on } (p, q)\}$ works well when a constant term is subtracted from a function $Z^s(\cdot)$ for each scenario. So, we obtain the following result.

Corollary 3.2 $(\text{ARO})_D$ *for the 1-median location problem on a tree is also solved in the same running time of $(\text{ARO})_A$. \square*

Moreover, the algorithm also works correctly when the exchange cost is not symmetric, that is, $c^s(x, y) \neq c^s(y, x)$.

3.3. Speeding up for solving a special case

Lastly, we discuss the case in which the weights w^s and exchange costs c^s are restricted to nonnegative values and positive values, respectively. In this case, we can find an optimal location point efficiently.

Lemma 3.4 *For any scenario $s \in S$, a 1-median for s attains $\min_{x \in G} Z^s(x)$. Moreover, $Z^s(\cdot)$ is strictly quasi-convex on any path in G .*

Proof. Let x^s be a 1-median for a scenario s . Then we have, for any $x \in G$,

$$Z^s(x) = f(y_x^s, D^s) + c^s(x, y_x^s) \geq f(y_x^s, D^s) \geq f(x^s, D^s) \geq Z^s(x^s).$$

Hence x^s attains $\min_{x \in G} Z^s(x)$.

We next show that for a point x lying in the path between x_1 and x_2 with $Z^s(x_1) \neq Z^s(x_2)$, $Z^s(x) < \max\{Z^s(x_1), Z^s(x_2)\}$ holds. Without loss of generality, assume that x_2 is in the descendants of x in the tree G rooted at a 1-median x^s . Note that $f(\cdot, D^s)$ is convex on any path in G . Hence, if $f(x_2, D^s) > f(x^s, D^s)$, then $f(x_2, D^s) > f(x, D^s)$ holds. At first, we consider the case of $y_{x_2}^s = x_2$. If $f(x_2, D^s) > f(x^s, D^s)$, then we have

$$Z^s(x_2) = f(x_2, D^s) > f(x, D^s) \geq Z^s(x).$$

Otherwise, since both of x_2 and x are 1-median, we have $Z^s(x) = \min_{x' \in G} Z^s(x') = Z^s(x_2) < Z^s(x_1)$. In the case of $y_{x_2}^s \neq x_2$, if $y_{x_2}^s$ is in the descendants of x in the tree G rooted at a 1-median x^s , then $f(y_{x_2}^s, D^s) \geq f(x, D^s) \geq f(x^s, D^s)$ hold. Thus we obtain

$$Z^s(x_2) = f(y_{x_2}^s, D^s) + c^s(x_2, y_{x_2}^s) > f(y_{x_2}^s, D^s) \geq f(x, D^s) \geq Z^s(x).$$

Otherwise, since $y_{x_2}^s$ is nearer to x than x_2 , we have

$$Z^s(x_2) = f(y_{x_2}^s, D^s) + c^s(x_2, y_{x_2}^s) > f(y_{x_2}^s, D^s) + c^s(x, y_{x_2}^s) \geq Z^s(x).$$

□

From the above lemma and the properties of strictly quasi-convex functions, we obtain the following.

Corollary 3.3 *The function $Z(\cdot)$ is strictly quasi-convex on any path in G . Thus, a local minimum of $Z(\cdot)$ is a global minimum.*

Corollary 3.4 *Assume that s_x is a worst case scenario at the point x , that is, $Z(x) = Z^{s_x}(x)$. For any point y in the descendants of x in the tree G rooted at a 1-median for the scenario s_x , $Z(y) \geq Z(x)$ holds.*

Proof. Since x lies in the path between a 1-median for s_x and y , we have

$$Z(y) \geq Z^{s_x}(y) \geq Z^{s_x}(x) = Z(x).$$

□

These properties imply that we can adopt the procedure developed by Averbakh and Berman for the minmax regret 1-median problem on a tree [5]. A centroid of a tree is a vertex x that minimizes over all vertices the largest number of vertices in subtrees obtained by deleting x from the tree. The procedure initializes $G' = G$ and repeats the following steps until G' contains only one edge or we find an optimal point:

- (1) find a centroid x in G' ;
- (2) determine a worst case scenario s_x in x ;
- (3) update G' by deleting the descendant of x in G' rooted at a 1-median for s_x .

Steps (1) and (3) can be performed in $O(|V'|)$ time, where V' is the vertex set of G' [12, 13]. Step (2) runs in $O(|S|)$ time if we have information obtained at preprocessor A2. It is known that for a centroid x , the number of vertices in any subtree obtained by deleting x from the tree, does not exceed $3/4$ of the number of all vertices in the tree. Therefore, the above procedure runs in $O(|V| + |S| \log |V|)$ time.

We now show an efficient way of finding an optimal location point after the above procedure outputs an edge (p, q) . From Corollary 3.3, we only have to find a local optimal point. So, we adapt Kouvelis and Yu's procedure with some modification, because functions $Z_{pq}^s(\cdot)$ may not be linear. From Equation (3.1), if a 1-median for s is nearer to p than q , $Z_{pq}^s(x) = \min \{f(\tilde{y}_{p \leftarrow q}^s, D^s) + c^s(x, \tilde{y}_{p \leftarrow q}^s), f(x, D^s)\}$ holds, since $f(x, D^s) \leq f(\tilde{y}_{q \leftarrow p}^s, D^s) \leq f(\tilde{y}_{q \leftarrow p}^s, D^s) + c^s(x, \tilde{y}_{q \leftarrow p}^s)$. Hence, $Z_{pq}^s(\cdot)$ has at most one break-point for each $s \in S$. Therefore, for any pair of functions $Z_{pq}^s(\cdot)$ and $Z_{pq}^{s'}(\cdot)$, there are at most three intersection points. The procedure starts by grouping the functions $Z_{pq}^s(\cdot)$ in pairs in an arbitrary fashion. If one of the functions in a pair is below the other function in the pair, the former function does not contribute to $Z_{pq}(\cdot)$, and so can be eliminated. If this is not the case, then two lines in the pair intersect at at most three points. Let x^1, \dots, x^ℓ be all of the intersection points for the various pairs. We can determine the $\frac{1}{4}\ell$ -th nearest

point $x^{(1)}$ to p , the $\frac{1}{2}\ell$ -th point $x^{(2)}$, and the $\frac{3}{4}\ell$ -th point $x^{(3)}$ in linear time. Let $\lambda^{(i)}$ and $\rho^{(i)}$ be the left and right gradient values of $Z(x^{(i)})$ for $i = 1, 2, 3$. If $\lambda^{(i)} \leq 0 \leq \rho^{(i)}$, then $x^{(i)}$ is an optimal location point. Otherwise, if $\lambda^{(1)} > 0$, then an optimal location point is in the interval $[p, x^{(1)}]$. Thus, we can eliminate one function from each pair for which all of the intersection points are greater than $x^{(1)}$. The eliminated function is the function for which the value at $x^{(1)}$ is lower in the pair. If the values at $x^{(1)}$ are equivalent, then the function of which the slope at $x^{(1)}$ is higher can be eliminated. If $\rho^{(1)} < 0$ and $\lambda^{(2)} > 0$, we can eliminate one function from each pair in which all intersection points are not in the interval $[x^{(1)}, x^{(2)}]$ in a manner similar to that described above. A symmetric procedure can be developed for the case of $\rho^{(3)} < 0$ and the case of $\rho^{(3)} > 0$ and $\lambda^{(2)} < 0$. Each iteration reduces the number of functions by a factor of 8. Repeated application of this procedure will finally result in two functions. By evaluating $Z(\cdot)$ at intersections of these functions and at vertices p and q , we obtain an optimal location point. This procedure finds an optimal point in the edge (p, q) in $O(|S|)$ time.

Theorem 3.2 *When the weights w^s and exchange costs c^s are nonnegative and positive, respectively, both $(\text{ARO})_A$ and $(\text{ARO})_D$ for the 1-median location problem on a tree can be solved in $O(|S||V|)$ time.*

Proof. We can obtain an edge containing an optimal location point in $O(|V| + |S| \log |V|)$ if we have obtained information of preprocessor A2, which is performed in $O(|S||V|)$ time. Since an optimal point on the edge can be found in $O(|S|)$ time, we obtain the desired complexity. \square

4. Conclusion

The present paper developed a new adjustable robust optimization model and applied it to a 1-median location problem on a tree. We showed a polynomial time algorithm for the problem. The application of our adjustable robust model to other problems such as the 1-center location problem will be investigated in the future.

Acknowledgements

I'm grateful for helpful comments of the referees. This research was partially supported by the MEXT Grant-in-Aid for Scientific Research (C) No. 19510137.

References

- [1] M.A. Aloulou, R. Kalai and D. Vanderpooten: Minmax regret 1-center problem on a network with a discrete set of scenarios. *Cahiers de recherche en ligne du LAMSADE - Document no 132* (2005).
- [2] I. Averbakh: On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, **90** (2001), 263–272.
- [3] I. Averbakh: Complexity of robust single facility location problems on networks with uncertain edge length. *Discrete Applied Mathematics*, **127** (2003), 505–522.
- [4] I. Averbakh: The minmax relative regret median problem on networks. *INFORMS Journal on Computing*, **17** (2005), 451–461.
- [5] I. Averbakh and O. Berman: An improved algorithm for the minmax regret median problem on a tree. *Networks*, **41** (2003), 97–103.
- [6] I. Averbakh and V. Lebedev: Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, **138** (2004), 289–301.

- [7] A. Ben-Tal, A. Goryashko, E. Guslitzer and A. Nemirovski: Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, **99** (2004), 351–376.
- [8] R.E. Burkard and H. Dollani: Robust location problems with pos/neg weights on a tree. *Networks*, **38** (2001), 102–113.
- [9] R.E. Burkard and H. Dollani: A note on the robust 1-center problem on trees. *Annals of Operations Research*, **110** (2002), 69–82.
- [10] R.E. Burkard and J. Krarup: A linear algorithm for the pos/neg-weighted 1-median problem on a cactus. *Computing*, **60** (1998), 193–215.
- [11] E. Conde: An improved algorithm for selecting p items with uncertain returns according to the minmax-regret criterion. *Mathematical Programming*, **100** (2004), 345–353.
- [12] A.J. Goldman: Optimal center location in simple networks. *Transportation Science*, **5** (1971), 212–221.
- [13] O. Kariv and S.L. Hakimi: An algorithmic approach to network location problems: Part 2. The p -medians. *SIAM Journal on Applied Mathematics*, **37** (1979), 539–560.
- [14] P. Kouvelis and G. Yu: *Robust Discrete Optimization and its Applications* (Kluwer Academic Publishers, Norewell, MA, 1997).
- [15] H.I. Yu, T.C. Kin and B.F. Wang: Improved algorithms for the minmax regret 1-median problem. *Lecture Notes in Computer Science*, **4112** (2006), 52–62.

Maiko Shigeno
Graduate School of Systems and Information Engineering
University of Tsukuba
Tsukuba, Ibaraki 305-8573, Japan
E-mail: maiko@sk.tsukuba.ac.jp