



# Robust vertex centdian facility location problem on tree networks

Soudabeh Seyyedi Ghomi<sup>1</sup> · Fahimeh Baroughi<sup>1</sup>

Received: 17 September 2023 / Accepted: 3 June 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

## Abstract

This paper studies the robust vertex centdian location problem on tree networks with interval edge lengths. To obtain a robust solution, we use the minmax regret criterion. First, we obtain at most  $n$  worst case scenario for each pair of vertices on tree networks. Then we reduce this number to  $\log n$ . Finally, the total time for obtaining a robust solution is equal to  $O(n^4)$ . This time is reduced to  $O(n^3 \log n)$  by using the property of binary search tree and weighted centroid.

**Keywords** Minmax regret · Robust optimization · Minmax criterion · Centdian location problem

## 1 Introduction

The centdian location problem is the combination of  $p$ -center and  $p$ -median location problems which has been defined by Hakimi (1965), Kariv and Hakimi (1979). In a given system with specified set of clients and candidate locations, the aim of the median problem is to find the location of points on system which minimize the total weighted distances from each client to its closest established facility. Also, on this system, the center problem wants to find locations among candidate locations which minimize the maximum weighted distances of each client to its closest established facility.

If  $f_c(x) = \max_{1 \leq i \leq n} w_i d(X, v_i)$  presents the objective function of the  $p$ -center problem and  $f_m(x) = \sum_{1 \leq i \leq n} w_i d(X, v_i)$  denotes the objective function of the  $p$ -median problem, then the centdian problem is considered as a bi-objective model as  $\min_x \{f_c(x), f_m(x)\}$ . One of the methods to solve a multi-objective problem in the literature is to convert the multi-objective model into a one-objective model. This is done in several ways. The two most important methods are the  $\epsilon$ -constrained method and the weighted sum method (Ehrgott, 2005). The weighted sum method is also known as the combination method.

In  $\epsilon$ -constrained method, one of the  $p$ -center and  $p$ -median objective functions must be minimized, provided that the other objective function is not greater than  $\epsilon$ . The weighted sum method considers the convex combination of  $p$ -median and  $p$ -center objective functions as objective function. Revelle and Swain (1970) and Halpern (1978) introduced the first

✉ Fahimeh Baroughi  
baroughi@sut.ac.ir

<sup>1</sup> Department of Applied Mathematics, Faculty of Basic Sciences, Sahand University of Technology, Tabriz, Iran

$\epsilon$ -constrained method and combination method which are also called the central facilities location problem and centdian location problem, respectively. Berman et. al. described a  $\epsilon$ -constrained median-center problem called the modified  $p$ -median problem (Toregas et al., 1971). In this model,  $p$ -medians are to be found such that the center objective is less than  $\epsilon$ . For more information about this method, see (Halpern, 1979; Tamir et al., 1998). In these articles, instead of the distance function in the median objective function, they considered a non-decreasing convex function.

In the weighted sum method, the convex combination of the median and center objective functions are considered as objective function of centdian problem. In combination method, instead of convex combination, the simple sum of the median and center functions are considered. Also, in some methods, the weight of all clients are the same in both center and median problems and in some cases are different. The last case is called the double weighted centdian problem.

Halpern studied the 1-centdian location problem on general graphs and trees, in which the weight of all clients in center problem are equal to one (Halpern, 1980). Tamir et al. considered the  $p$ -centdian problem on tree networks with different weights for any client in both center and median problems, and presented an algorithm with  $O(pn^6)$  time complexity for  $p \geq 4$  and  $O(n^p)$  time complexity for  $p < 4$  (Tamir et al., 1998).

Assume now that the input data of a problem have not an certain value. For example, consider a network with uncertain vertex weights or uncertain edge lengths or both. Then the presented algorithms in above can not solve the centdian problems on such networks. If the value of input data is specified by a distribution function, then we can use the probably optimization methods. If the value of these data is characterized by a membership function, then we can use the fuzzy optimization approaches. Assume now that the input data of a problem have not an certain value. For example, consider a network with uncertain vertex weights or uncertain edge lengths or both. Then the above mentioned algorithms can not solve the centdian problems on such networks. If the value of input data is specified by a distribution function, then we can use the probably optimization methods. If the value of these data are characterized by a membership function, then we can use the fuzzy optimization approaches. Suppose that the value of input data of a problem do not be specified by a distribution or membership function. Such problems are belong to the robust optimization models and they can be solved by existing approaches in the robust optimization. The first approach in robust optimization presented by Soyster. He considered a linear programming model in 1973 and tried to counteract to the uncertainty of data, by a conservative approach (Soyster, 1973). The final solution of his approach was far away from optimal solution, because he considered the worst cases. Mulvey et al. investigated Soyster's approach in discrete optimization (Mulvey et al., 1995). Ben Tal et al. presented new algorithm for robust optimization, which is not efficient (Ben-Tal & Nemirovski, 2000). Bertsimas et al. introduced an algorithm, counteract the uncertainty of data by controlling the level of conservative. By using this method, they obtained the new linear optimization model, and called the robust counterpart of first model (Bertsimas & Sim, 2004). One of the important approaches in robust optimization is minmax regret criterion, which is proposed by Kouvelis and Ye (1993). In this method, the maximum difference between the cost (profit) of decision of decision maker and the cost (profit) of optimal decision is minimized.

In the robust facility location problems, the minmax regret criterion use for obtaining the robust solution by using the combinatorial approaches. Averbakh (2003) showed that the minmax regret 1-center and 1-median problems on general graphs with uncertain edge lengths are NP-hard. These problems on special graphs, such path, tree, cycle, cactus and etc can be solved in polynomial time. Also, these problems on general graphs with uncertain

vertex weights are solvable in polynomial time. Averbakh et al. considered the minmax regret 1-center problem on weighted trees with interval edge lengths and vertex weights and on un-weighted trees with interval edge lengths and presented solution methods with  $O(n^6)$  and  $O(n^2 \log n)$  time complexity, respectively (Averbakh & Berman, 2000a). Burkard et al. considered the above problems and improved their time complexity to  $O(n^3 \log n)$  and  $O(n \log n)$ , respectively (Burkard & Dollani, 2002). Also, Aloulou et al. obtained a combinatorial algorithm for minmax regret 1-center problem on general graphs and trees with time complexity  $O(mn(\log n + |S|)|S|)$  and  $O(n|S|(n + \log n))$ , respectively (Aloulou et al., 2005). In these cases, the vertex weights of general graph and tree are described by a discrete set of scenarios such  $S$ . Kouvelis considered the minmax regret 1-median problem on trees and proposed an algorithm with  $O(n^4)$  time complexity (Kouvelis et al., 1993). Chen et al. improved this time to  $O(n|S|)$  (Chen & Lin, 1998). Also, Averbakh et al. obtained two algorithms for the robust 1–median problem on general graphs and on tree networks with time complexity  $O(mn^2 \log n)$  and  $O(n \log^2 n)$ , respectively (Averbakh & Berman, 2000b, 2003).

Conde considered the robust double weighted centdian problem on tree networks and presented an algorithm with  $O(n^3 \log n)$  time (Conde, 2008). Li et al. studied the robust centdian problem with interval vertex weights on dynamic graphs and proposed an algorithm with  $O(n^3 \log n)$  time complexity (Li et al., 2018). In this case, the weight of each node at center and median problems are the same. Up to now, the robust centdian problem with uncertainty edge lengths not investigated.

In this paper, we consider the robust vertex centdian problem on trees with interval edge lengths. We organize the paper in several sections. Section 2 defines the classic and robust vertex centdian problem on networks. We obtain a finite subset of worst case scenarios for the robust vertex centdian problem in Sect. 3 and solve the robust vertex centdian problem under these scenarios. Finally, conclusion is given in Sect. 4.

## 2 Problem statement

Let  $T = (V, E)$  be a tree network with vertex set  $V = \{v_1, \dots, v_n\}$  and edge set  $E$  such that each vertex  $v_i \in V$  is associated with a non-negative weight  $w_i$  and each edge  $e \in E$  is associated with a non-negative length  $l(e)$ . For each pair of vertices  $x, y \in V$ , let  $d(x, y)$  be the length of shortest path between  $x$  and  $y$ . In the classical vertex centdian problem the aim is to find a vertex  $v^* \in V$  such that it minimizes the following objective function:

$$f_c(x) + f_m(x), \quad \forall x \in V \quad (1)$$

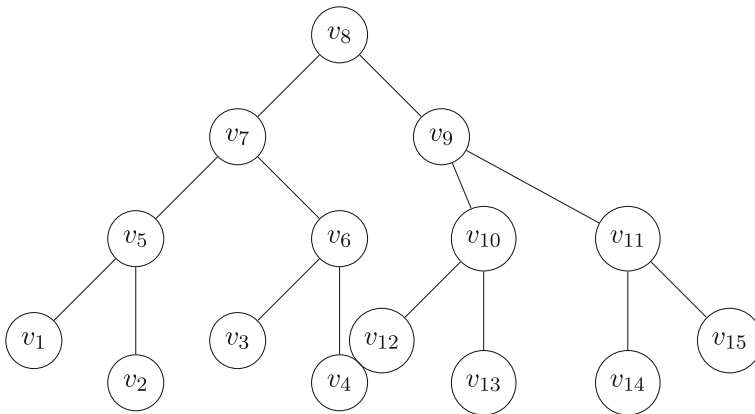
where

$$f_c(x) = \max_{i=1, \dots, n} w_i d(x, v_i)$$

and

$$f_m(x) = \sum_{i=1}^n w_i d(x, v_i).$$

Now assume that the length of edges of  $T$  do not have an exact and specified value. Let every edge  $e \in E$  is indicated by an interval  $[l^-(e), l^+(e)]$  such that  $l^+(e) \geq l^-(e) \geq 0$  and  $l(e) \in [l^-(e), l^+(e)]$ . Consider the set  $S$  as the cartesian product of these intervals. The set  $S$  is called the set of all happens of uncertain parameters. Any  $s \in S$  is a scenario and  $l_e^s$  is



**Fig. 1** An example

the length of edge  $e$  under this scenario. In this paper, we use the minmax regret criterion for obtaining the robust solution of uncertain centdian problem. Therefore, the robust vertex centdian problem is formulated using this criterion as follows:

$$\min_{x \in V} \max_{s \in S} [f_c(x, s) + f_m(x, s) - f_c(x^*, s) - f_m(x^*, s)] \quad (2)$$

where  $x^*$  is a vertex centdian on  $T$  under scenario  $s \in S$  and

$$f_c(x, s) = \max_{i=1, \dots, n} w_i d^s(x, v_i), \quad f_m(x, s) = \sum_{i=1}^n w_i d^s(x, v_i).$$

Note that  $d^s(x, v_i)$  represents the length of the shortest path between  $x$  and  $v_i$  under scenario  $s \in S$ . Obviously,  $|S| = \infty$ . Thus it is not easy to find the optimal solution of (2). Therefore, for obtaining the optimal solution of these problems, we must search the finite subset of  $S$  and solve these problems on the obtained subset. This subset of scenarios are selected so that the value of  $R^s(x, y) = f_c(x, s) + f_m(x, s) - f_c(y, s) - f_m(y, s)$  is maximized for all pair of vertices  $x, y$  on  $T$ . Scenarios that maximize the value of  $R^s(x, y)$  are called the worst case scenarios.

To find these scenarios, first select an arbitrary vertex on  $T$  and root the tree on this vertex. Call this vertex  $r$ . Convert this tree into a binary tree by adding at most  $O(n)$  vertices. Set the corresponding vertex weights and edge lengths equal to zero (Tamir, 1996). Let  $u_L$  and  $u_R$  be the left and right children of vertex  $r$ . Call the rooted subtrees in these two vertices  $u_L$  and  $u_R$ ,  $T_L$  and  $T_R$ , respectively.

Reorder the indices of the nodes of binary tree  $T$  such that the indices of all vertices on subtree  $T_L$  are smaller than the index of  $u_L$ . In other words, if  $v_i$  and  $v_j$  are the left and right descendants of vertex  $v_t$ , respectively, then  $i < t$ ,  $j < t$  and  $i < j$ . Also, the indices of all vertices on subtree  $T_R$  are greater than the index of  $u_R$ . That is, if  $v_i$  and  $v_j$  are the left and right descendants of vertex  $v_t$ , respectively, then  $i > t$ ,  $j > t$  and  $i < j$  (see Fig. 1). This reordering is performed in linear time (Korte & Vygen, 2012).

Let  $r = v_k$  and  $v_1, \dots, v_{k-1}$  be the nodes of subtree  $T_L$  and  $v_{k+1}, \dots, v_n$  be the nodes of subtree  $T_R$ . Also, assume that the weight of vertices of tree  $T$  are satisfied in the following inequality:

$$0 \leq w_1 \leq w_2 \leq \dots \leq w_n. \tag{3}$$

Now set all the leaves of subtree  $T_R$  in the set  $Leav(R)$ . Then for each  $x \in V$  and  $s \in S$ , we have

$$\begin{aligned} f_c(x, s) &= \max \left\{ \max_{1 \leq i \leq k} w_i d^s(x, v_i), \max_{k+1 \leq i \leq n} w_i d^s(x, v_i) \right\} \\ &= \max \left\{ \max_{1 \leq i \leq k} w_i d^s(x, v_i), \max_{v_i \in Leav(R)} w_i d^s(x, v_i) \right\}. \end{aligned} \tag{4}$$

We know that for each vertex  $v_i \in Leav(R)$ , there exists at least an arbitrary vertex  $v_j$  on unique path between  $x$  and  $v_i$  such that

$$w_i d^s(x, v_i) = w_i d^s(x, v_j) + w_i d^s(v_j, v_i).$$

On the other hand, we know that  $w_i d^s(v_j, v_i) \geq 0$ . Then we can conclude the following statement:

$$\begin{aligned} w_i d^s(x, v_i) &\geq w_i d^s(x, v_j) \\ &\geq w_j d^s(x, v_j). \end{aligned}$$

The last inequality follows from this assumption that for all  $i > j$ , we have  $w_i \geq w_j$ . As mentioned earlier, solving the problem (2) requires finding a finite number of worst-case scenarios. So the first step is to find these scenarios. For finding the worst-case scenarios, we should have information about worst case scenarios corresponding to median and center problems.

### 3 Worst case scenarios of the robust centdian problem

For obtaining the worst case scenarios of the robust centdian problem, first we must know about the worst case scenarios of the center and median problems. Then by using these scenarios, we can obtain the worst case scenarios of the robust centdian problem.

#### 3.1 Worst case scenario of the robust center problem

Let  $x$  be an arbitrary vertex on tree  $T$ ,  $s_x$  be a worst case scenario according to  $x$  and  $v_x$  be the farthest vertex from  $x$  under scenario  $s_x$ . i.e, the weighted distance of this vertex from  $x$  is maximum ( $v_x$  is called the critical vertex of  $x$ ). Let  $x, y, z$  be the vertices of a tree network with weights 2, 5, 1 and  $d(x, y) = 2, d(x, z) = 4$ . Then the vertex  $y$  is the critical vertex of  $x$ . Because

$$w(y) * d(x, y) > w(z) * d(x, z).$$

Averbakh and Berman (2000a) showed that  $s_x$  is satisfied in the following condition:

1. The edge lengths belonging to the path between  $x$  and  $v_x$ , are equal to upper bounds of the associated intervals.
2. The other edge lengths are equal to lower bounds of according intervals.

Define  $S(x) = \bigcup_{i=1}^n s_i(x)$ . It is obvious that  $s_x \in S(x)$ . We use (4) and conclude that  $s_x$  belongs to  $\tilde{S}(x) = (\bigcup_{i=1}^k s_i(x)) \cup (\bigcup_{v_i \in Leav(R)} s_i(x))$ .

### 3.2 Worst case scenario of the robust median problem

Let  $x$  be a desired vertex on the tree  $T$ ,  $s_x$  be a worst case scenario for  $x$  and  $y$  be a 1–median on the tree  $T$  under scenario  $s_x$ . Suppose  $T_u^v$  and  $T_v^u$  are the rooted subtrees of tree  $T$  in  $u$  and  $v$  respectively and obtained by removing edge  $e = (u, v)$ . Also,  $T_u^v$  contains the vertex  $u$  but does not contain the vertex  $v$  and  $T_v^u$  contains the vertex  $v$  but does not contain the vertex  $u$ . Assume that the total weight of the subtrees  $T_u^v$  and  $T_v^u$  are denoted by  $W(T_u^v)$  and  $W(T_v^u)$ , respectively. Averbakh and Berman (2000b) proved that  $s_x$  have the following properties:

1. The length of edge  $e = (u, v)$  that belongs to path between  $x$  and  $y$  and  $y \in T_v^u$ , is computed as follows:

$$l^{s_x}(e) = \begin{cases} l^+(e) & \text{if } W(T_u^v) < W(T_v^u), \\ l^-(e) & \text{if } W(T_u^v) > W(T_v^u). \end{cases} \quad (5)$$

If  $W(T_u^v) = W(T_v^u)$ , then the length of edge  $e$  can be considered any value in the associated interval.

2. The length of edges not belong to path between  $x$  and  $y$ , can be any arbitrary value on the associated intervals.

Therefore, for every pair of vertices  $x$  and  $y$  on tree  $T$  a scenario can be calculated that holds in the above conditions.

### 3.3 Worst case scenarios of the robust centdian problems

Now consider the minmax regret vertex centdian problem presented in (2). This problem can be rewrite as follows:

$$\min_{x \in V} \max_{y \in V} \max_{s \in S} R^s(x, y). \quad (6)$$

In this case, for finding a finite subset of  $S$ , we must obtain a scenario  $s(x, y)$  for all pair of nodes  $x$  and  $y$  such that it maximizes  $R^s(x, y)$ . In other words, we must compute  $s(x, y)$  satisfies in the following relation:

$$R^{s(x,y)}(x, y) = \max_{s \in S} R^s(x, y).$$

**Lemma 3.1** *Assume that  $x$  and  $y$  are two arbitrary vertices on  $T$  and  $s \in S$ . Therefore, we can conclude the following statements:*

1. Let  $T_x^y$  be a subtree on  $T_R$ . If the critical vertex of  $x$  belongs to  $T_x^y \cap \text{Leav}(R)$ , then the critical vertex of  $y$  is one of the vertices in  $\text{Leav}(R)$ , in which its index is equal or greater than the index of critical vertex of  $x$ .
2. Let  $T_x^y$  be a subtree of  $T_L$ . If the critical vertex of  $x$  belongs to  $T_x^y$ , then the critical vertex of  $y$  is either one of the vertices in  $\text{Leav}(R)$  or one of the vertices with index greater than index of the critical vertex of  $x$ .

**Proof** 1. Let  $v^*$  be the critical vertex of  $x$  under scenario  $s$  and  $v^*$  be in the subtree  $T_R$ . Therefore, we have

$$w(v^*)d^s(x, v^*) \geq w_i d^s(x, v_i), \quad \forall v_i \in V(T).$$

In the other hand,  $\forall v_i \in V(T) \setminus V(T_x^y)$ ,  $w_i \leq w(v^*)$ . Then

$$w(v^*)d^s(y, v^*) = w(v^*)d^s(y, x) + w(v^*)d^s(x, v^*)$$

$$\begin{aligned} &\geq w(v^*)d^s(y, x) + w_i d^s(x, v_i) \\ &\geq w_i d^s(y, x) + w_i d^s(x, v_i) \end{aligned}$$

2. The proof of this part is the similar to the previous part. □

**Lemma 3.2** *Let  $e = (u, v)$  be an edge on path between  $x$  and  $y$ , which is closest to  $x$  and  $W(T_u^v) > W(T_v^u)$ . Therefore, for all edge  $(a, b)$  on described path,  $W(T_a^b) > W(T_b^a)$ .*

**Proof** Assume that  $(a, b)$  is an arbitrary edge except  $e$  on path between  $x$  and  $y$  such that  $y \in T_b^a$ . Then

$$W(T_a^b) = W(T_u^v) + \sum_{v_i \in P[v, a]} w_i, \quad W(T_b^a) = W(T_v^u) - \sum_{v_i \in P[v, a]} w_i.$$

Consequently, we have

$$W(T_b^a) \leq W(T_v^u) < W(T_u^v) \leq W(T_a^b).$$

The first and last inequalities are inferred from the non-negativity of the weights. □

**Corollary 3.3** *Suppose that  $e = (u, v)$  is the adjacent edge of  $y$  on the path between  $x$  and  $y$ . Also,  $W(T_u^v) < W(T_v^u)$ . Then for all edge  $(a, b)$  on this path, we have*

$$W(T_a^b) < W(T_b^a).$$

In the following, we want to find the worst case scenario for all vertices  $x$  and  $y$  on the tree  $T$ . For this purpose, we must compute the values  $W(T_u^v)$  and  $W(T_v^u)$  for all edge  $e = (u, v)$  on  $T$ . Let  $v_i$  be an arbitrary vertex in  $T_L \cup Leav(R)$ . Therefore, scenario  $s_i(x, y)$  is computed as follows:

**Case 1.** Let  $\tilde{T}_x$  be a rooted subtree in vertex  $x \in T$ ,  $y \notin \tilde{T}_x$  and  $v_i$  be a vertex on subtree  $\tilde{T}_x$ . In other words, there is no edge on intersection of the paths  $P[x, v_i]$  and  $P[x, y]$ . In this case, the length of all edges on  $T$ , under scenario  $s_i(x, y)$  is characterized in the following way:

1. The length of all edges on path  $P[x, v_i]$  equals to upper bounds of corresponding intervals.
2. The length of all edges on subgraph  $T \setminus (P[x, y] \cup P[x, v_i])$  equals to lower bounds of corresponding intervals.
3. The length of edge  $e = (u, v)$  on the path  $P[x, y]$  is calculated according to the following procedure:

If  $u = x$  and  $W(T_u^v) \geq W(T_v^u)$ , then the lengths of all edges on the path  $P[x, y]$  are equal to lower bounds of corresponding intervals. Because the length of these edges are equal to lower bound of intervals in both center and median problems. If  $W(T_u^v) < W(T_v^u)$ , then we must find the closest edge  $(a, b)$  to  $x$  on the path  $P[x, y]$ , which satisfies in the condition  $W(T_a^b) \geq W(T_b^a)$ . Set the edge lengths on the path between  $a$  and the vertex  $y$  equal to lower bounds of corresponding intervals.

Assume  $A_i$  contains all edges  $e = (u, v)$  on the path  $P[x, y]$  such that they satisfy in the condition  $W(T_u^v) < W(T_v^u)$ . In other words

$$A_i = \{e = (u, v) \in P[x, y] : W(T_u^v) < W(T_v^u)\}.$$

The length of edge  $e \in A_i$  in the center problem is equal to lower bound and in the median problem is equal to upper bound of associated interval. Then the length of these edges must be computed in centdian problem, separately.

Compute the weighted distance of all vertices of  $\bar{T}_x$  from  $x$ . Suppose  $v^*$  is a vertex on  $\bar{T}_x$  with maximum weighted distance from  $x$ . Set the edge lengths on  $A_i$  equals to upper bound of intervals and consider this scenario as  $s_i^+(x, y)$ . Compute the weighted distance of all vertices of  $T \setminus \bar{T}_x$  from  $x$ . Assume that  $v'$  has the maximum weighted distance from  $x$  in  $T \setminus \bar{T}_x$ .

**Subcase1:** Let  $w(v^*)d^{s_i^+(x,y)}(x, v^*) \geq w(v')d^{s_i^+(x,y)}(x, v')$ . Therefore,  $v^*$  is a critical vertex for  $x$  under scenario  $s_i^+(x, y)$  and consequently, on scenario  $s_i(x, y)$ . According to Lemma 3.1, the critical vertex of  $y$  is on  $T^* = \bar{T}_x \cup Leav(R)$ . Therefore, the critical vertex of  $y$  is either  $v^*$  or the index of this vertex is greater than the index of vertex  $v^*$ . Let  $v''$  be the critical vertex of  $y$  under scenario  $s_i^+(x, y)$ .

**Remark 3.4** Let  $e = (u, v)$  be an edge of  $A_i$ . Then for some  $v_j$  with index greater than the index of vertex  $v^*$ , we have

$$-w(v^*) + (-W(T_u^v) + W(T_v^u)) \geq -w_j + (-W(T_u^v) + W(T_v^u)).$$

If we use Remark 3.4 and define  $\Delta_e^* = -w(v^*) + (-W(T_u^v) + W(T_v^u))$ , then we conclude the following Lemma:

**Lemma 3.5** If  $v'' = v^*$  and  $\Delta_e^* \geq 0$ , then the length of edge  $e$  on  $A_i$  under scenario  $s_i(x, y)$  is equal to its length under scenario  $s_i^+(x, y)$ .

**Proof** Suppose that  $\bar{s}_i$  is a obtained scenario from  $s_i^+(x, y)$  by decreasing the length of at least one edge of  $A_i$ . Also, let  $\bar{v}$  be the critical vertex of  $y$  under scenario  $\bar{s}_i$ . Then  $\bar{v} = v^*$  or the index of  $\bar{v}$  is exceed from index of  $v^*$ . If  $\bar{v} = v^*$ , then for all edge  $e = (u, v) \in A_i$  we have

$$\begin{aligned} f(x, \bar{s}_i) - f(y, \bar{s}_i) &= (-w(v^*) - W(T_u^v) + W(T_v^u))d^{\bar{s}_i}(x, y) \\ &= \Delta_e^* d^{\bar{s}_i}(x, y) \\ &\leq \Delta_e^* d^{s_i^+(x,y)}(x, y) \\ &= f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)). \end{aligned}$$

If  $v'' \neq v^*$ , then

$$\begin{aligned} f(x, \bar{s}_i) - f(y, \bar{s}_i) &\leq w(v^*)d^{s_i^+(x,y)}(x, v^*) - w(v^*)d^{s_i^+(x,y)}(y, v^*) - \alpha \\ &\quad + (-W(T_u^v) + W(T_v^u))(d^{s_i^+(x,y)}(y, v^*) - \alpha) \\ &= w(v^*)d^{s_i^+(x,y)}(x, v^*) - w(v^*)d^{s_i^+(x,y)}(y, v^*) \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^+(x,y)}(x, y) \\ &\quad + (w(v^*) + W(T_u^v) - W(T_v^u))\alpha \\ &\leq w(v^*)d^{s_i^+(x,y)}(x, v^*) - w(v'')d^{s_i^+(x,y)}(y, v'') \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^+(x,y)}(x, y) \\ &= f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)). \end{aligned}$$

The first inequality follows from definition of critical vertex and the second inequality concludes from  $\Delta_e^* \geq 0$ .  $\square$



Assume now that  $v'' = v^*$  and  $\Delta_e^* < 0$ . According to Remark 3.4, for all  $v_j$  with index greater than  $v^*$ , we have  $\Delta_e^j < 0$ .

Suppose  $s_i^-(x, y)$  is a scenario contains all edges in  $A_i$  with lower bound of corresponding intervals. Let  $v^-$  be the critical vertex of  $y$  under this scenario.

**Lemma 3.6** *Let  $v^* = v''$  and  $\Delta_e^j < 0$ . Then the following statements are true.*

- a. *If  $v^* = v^-$ , then the length of edge  $e \in A_i$  under scenario  $s_i(x, y)$  is equal to length of this edge under scenario  $s_i^-(x, y)$ .*
- b. *If  $v^* \neq v^-$ , then the length of edge  $e \in A_i$  under scenario  $s_i(x, y)$  is equal to length of this edge under scenario  $s_i^+(x, y)$ .*

**Proof** a. Let  $s_i^-$  be the obtained scenario from  $s_i^+(x, y)$  by decreasing the length of at least one edge on  $A_i$ . Also, assume  $v^+$  is the critical vertex of  $y$  under scenario  $s_i^-$ . Therefore,

$$\begin{aligned} f(x, s_i^-) - f(y, s_i^-) &= w(v^*)d^{s_i^-}(x, v^*) - w(v^+)d^{s_i^-}(y, v^+) \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^-}(x, y) \\ &\leq w(v^*)d^{s_i^-}(x, v^*) - w(v^*)d^{s_i^-}(y, v^*) \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^-}(x, y) \\ &= (-w(v^*) - W(T_u^v) + W(T_v^u))d^{s_i^-}(x, y) \\ &= \Delta_e^* d^{s_i^-}(x, y) \\ &\leq \Delta_e^* d^{s_i^-(x,y)}(x, y) \\ &= f(x, s_i^-(x, y)) - f(y, s_i^-(x, y)). \end{aligned}$$

Also, we have

$$\begin{aligned} f(x, s_i^-(x, y)) - f(y, s_i^-(x, y)) &= w(v^*)d^{s_i^-(x,y)}(x, v^*) - w(v^-)d^{s_i^-(x,y)}(y, v^-) \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^-(x,y)}(x, y) \\ &= (-w(v^*) - W(T_u^v) \\ &\quad + (W(T_v^u))d^{s_i^+(x,y)}(x, y) \\ &\quad + (w(v^*) + W(T_u^v) - W(T_v^u)) \sum_{e \in A_i} \alpha_e \\ &> f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)) \end{aligned}$$

where  $\alpha = l_e^+ - l_e^-$ .

b. In this case, we have

$$\begin{aligned} &-w(v^*)d^{s_i^+(x,y)}(y, v^*) + (-W(T_u^v) + W(T_v^u))d^{s_i^+(x,y)}(x, y) \\ &= -w(v^*)d^{s_i^-(x,y)}(y, v^*) + (-W(T_u^v) + W(T_v^u))d^{s_i^-(x,y)}(x, y) \\ &\quad + (w(v^*) + W(T_u^v) - W(T_v^u)) \sum_{e \in A_i} \alpha_e \\ &\geq -w(v^-)d^{s_i^-(x,y)}(y, v^-) + (-W(T_u^v) + W(T_v^u))d^{s_i^-(x,y)}(x, y) - \Delta_e^* \sum_{e \in A_i} \alpha_e \\ &\geq -w(v^-)d^{s_i^-(x,y)}(y, v^-) + (-W(T_u^v) + W(T_v^u))d^{s_i^-(x,y)}(x, y). \end{aligned}$$

If we add  $w(v^*)d^{s_i^-(x,y)}(x, v^*)$  to the left and right hand sides of above inequality, then we have

$$f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)) \geq f(x, s_i^-(x, y)) - f(y, s_i^-(x, y)).$$

Because we have  $w(v^*)d^{s_i^-(x,y)}(x, v^*) = w(v^*)d^{s_i^+(x,y)}(x, v^*)$ .

By the similar way, we conclude that

$$f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)) \geq f(x, s_i^-) - f(y, s_i^-).$$

□

Now suppose  $v^* \neq v''$ . In this case, the objective function of centdian problem in the vertices  $x$  and  $y$  under scenario  $s_i^+(x, y)$  is

$$\begin{aligned} f(x, s_i^+(x, y)) - f(y, s_i^+(x, y)) &= w(v^*)d^{s_i^+(x,y)}(x, v^*) - w(v'')d^{s_i^+(x,y)}(y, v'') \\ &\quad + (-W(T_u^v) + W(T_v^u))d^{s_i^+(x,y)}(x, y). \end{aligned}$$

**Corollary 3.7** *Let  $v^* \neq v''$ . Therefore, the length of edge  $e \in A_i$  under scenario  $s_i(x, y)$  obtains as follows:*

- If  $\Delta_e^* \leq 0$ , then the length of this edge on scenario  $s_i(x, y)$  is equal to its length under scenario  $s_i^+(x, y)$ .
- If for every vertex  $v_j$  with index greater than  $v^*$ ,  $\Delta_e^j \geq 0$ , then the length of this edge on scenario  $s_i(x, y)$  is equal to its length under scenario  $s_i^+(x, y)$ .

**Subcase2:** Assume now that

$$w(v^*)d^{s_i(x,y)}(x, v^*) = w(v^*)d^{s_i^+(x,y)}(x, v^*) < w(v')d^{s_i(x,y)}(x, v').$$

Let  $v'' \in T_y^x$ . In this case, the critical vertex of  $y$  on scenario  $s_i(x, y)$  is  $v''$ . Because the length of all edges on the path between  $y$  and  $v''$ , which are not in  $A_i$ , on scenarios  $s_i(x, y)$  and  $s_i^+(x, y)$  are the same.

**Case2:** Suppose the conditions of case1 are not met, i.e.  $y \in \bar{T}_x$  and  $v_i$  is not a vertex on subtree  $\bar{T}_x$ . In this case, one of the following conditions occurs.  $y \in \bar{T}_x$  and  $v_i$  is a vertex on subtree  $T \setminus \bar{T}_x$ , both vertices  $y, v_i$  are in the subtree  $T_j$  or  $y, v_i$  are not in the subtree  $T_j$ . If there is no edge on intersection of paths  $P[x, v_i]$  and  $P[x, y]$ , then the length of edges are calculated exactly like the case1. Assume now that  $p[x, v_i] \cap p[x, y] \neq \emptyset$ . In this case, the length of all edges on  $T$ , under scenario  $s_i(x, y)$  is characterized in the following form:

- The edge lengths on path  $P[x, v_i] \setminus p[x, y]$  are equal to upper bounds of corresponding intervals.
- The edge lengths on subgraph  $T \setminus (P[x, y] \cup P[x, v_i])$  are equal to lower bounds of corresponding intervals.
- Let  $e = (u, v) \in P[x, y] \cap P[x, v_i]$ . The length of this edge in the center problem is equal to upper bound of its corresponding interval. But this value in the median problem depends on the amount of weights  $W(T_u^v)$  and  $W(T_v^u)$ . If  $W(T_u^v) \leq W(T_v^u)$ , then the length of this edge in the centdian problem equals to upper bound of the corresponding interval. Let  $W(T_u^v) > W(T_v^u)$ . Define

$$B_i = \{e = (u, v) \in P[x, y] \cap P[x, v_i] : W(T_u^v) > W(T_v^u)\}.$$

Set the edge lengths of this set equal to the upper bounds of their intervals. Let  $w(\bar{v})d^s(y, \bar{v}) = \max_{1 \leq i \leq n} w_i d^s(y, v_i)$ . If  $p[x, y] \cap p[x, v_i] \cap p[y, \bar{v}] = \emptyset$ , then under the scenario  $s_i(x, y)$  the edge lengths in the set  $B_i$  do not decrease. Suppose this condition does not occur. Denote  $\bar{V}$  as the set of all vertices of the tree  $T$  where the intersection of the path  $P[x, y] \cap P[x, v_i]$  and the path between vertex  $v \in \bar{V}$  and  $y$  is empty. Let

$$w(v_y)d^{s_i(x,y)}(y, v_1) = \max_{1 \leq i \leq n} w_i d^{s_i(x,y)}(y, v_i).$$

Note that the edge lengths on the path between vertex  $v \in \bar{V}$  and  $y$  are specified on the scenario  $s_i(x, y)$ . If the condition of subcase1 is met, then we set the length of these edges equal to minimum value in the desired interval. If the condition of subcase2 is met, then we set the length of these edges similar to the subcase2.

**Lemma 3.8** *The robust vertex centdian problem can be solved in  $O(n^4)$  time.*

**Proof** For each pair of vertices in  $T_L \cup Leav(R)$  we compute at most  $O(n)$  worst case scenarios. The required time complexity to calculate these scenarios is linear. Because if vertices  $v_i$  are selected using BFS algorithm, then it is enough to calculate only the length of one edge in each iteration.  $\square$

As mentioned, at most  $O(n)$  scenarios was calculated for each pair of vertices on tree  $T$ . But some of these scenarios may be repetitive. So we are looking for ways to find these repetitive scenarios and ignore them. For this purpose, we use the concept of weight-centroid. The vertex  $v^*$  with adjacent vertices  $v_1, v_2$  and  $v_3$  on the tree  $T$  is called the weight-centroid if the following condition is met for subtrees  $T_{v_j}, j = 1, 2, 3$ .

$$W(T_{v_j}) \leq W(T)/2, \quad j = \{1, 2, 3\}.$$

These subtrees are obtained by removing the vertex  $v^*$  on the tree  $T$ . Using this concept, we obtain the following results, which show that it is not necessary that the worst case scenarios are calculated for all pair of vertices on tree  $T$ . Let  $n_j$  be the number of vertices on subtree  $T_j, j = 1, 2, 3$ .

**Lemma 3.9** *Suppose  $v_i$  and  $x$  are two vertices on subtree  $T_j$  and  $y_1, y_2 \in V(T) \setminus V(T_j)$ . Therefore,  $s_i(x, y_1) = s_i(x, y_2)$ . In this case, for each  $x \in V(T_j)$ , there are at most one vertex  $y$  with different scenario  $s_i(x, y)$ .*

**Proof** Given that  $x$  and  $v_i$  are in a different subtree from  $y_1$  and  $y_2$  (see Fig. 1), the scenarios  $s_i(x, y_1)$  and  $s_i(x, y_2)$  differ only in the length of edges on paths  $p[y_1, v^*]$  and  $p[y_2, v^*]$ . Therefore, we must examine the length of edges of these two paths. According to the Corollary 3.3, on every edge  $(u, v)$  on paths  $p[y_1, v^*]$  and  $p[y_2, v^*]$ ,  $W(T_u^u) \geq W(T_v^v)$ . Then the length of the edge  $(u, v)$  is equal to lower bound of corresponding interval. The last sentence is concluded from the part 3 of **case 1**. Here it suffices to consider  $y \in V(T) \setminus V(T_j)$  as  $v$ .  $\square$

**Lemma 3.10** *Suppose  $v_i$  and  $x$  are two vertices on subtree  $T_j$  and  $y_1, y_2 \in V(T_j)$  (see Fig. 3). If  $\bar{v}$  is the parent of  $y_1$  and  $y_2$  on path between  $x$  and  $r$ ,  $p[x, y_1] \cap p[v_i, x] = \emptyset$  and  $p[x, y_2] \cap p[v_i, x] = \emptyset$ , then  $s_i(x, y_1) = s_i(x, y_2)$ . In this case, for each  $x \in V(T_j)$ , there are at most  $\log n$  vertex  $y$  with different scenario  $s_i(x, y)$ .*

**Proof** This Lemma can be proved similar to Lemma 3.10. For given  $x$  and  $v_i$ , the number of parents for any pair of vertices  $y_1$  and  $y_2$  equals to the high of binary tree. The high of binary tree is at most  $O(\log n)$ .  $\square$

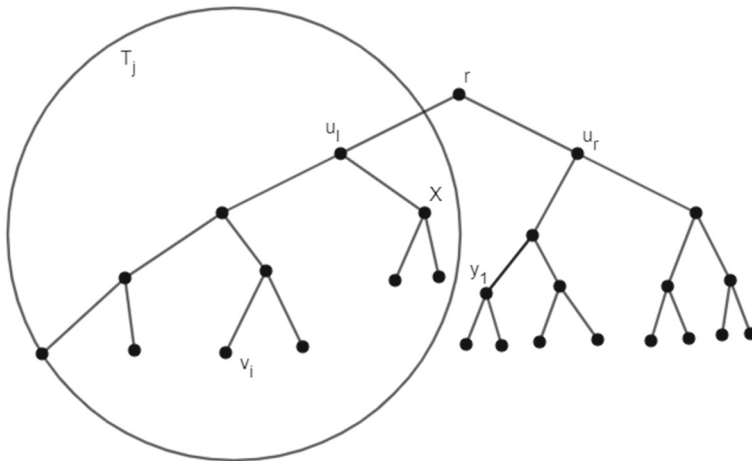


Fig. 2 Related graph of Lemma 3.9

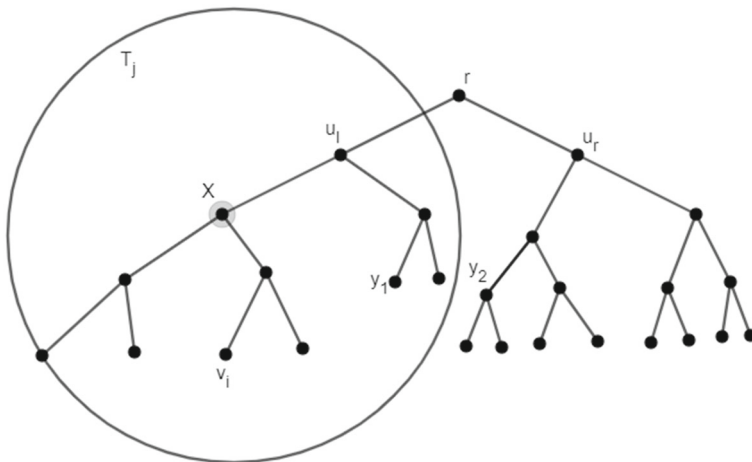


Fig. 3 Related graph of Lemma 3.10

The following three Lemmas can be concluded from case 2.

**Lemma 3.11** Suppose  $v_i$  is a vertex on subtree  $T_j$  such that  $p[x, v_i] \cap p[y_1, v_i] \neq \emptyset$ ,  $p[x, v_i] \cap p[y_2, v_i] \neq \emptyset$  and  $y_1, y_2$  have a common vertex on path between  $x$  and  $r$  (see Fig. 4). Then  $s_i(x, y_1) = s_i(x, y_2)$ . In this case, for each  $x \in V(T_j)$ , there is at most two  $y$  with different scenario  $s_i(x, y)$ .

**Lemma 3.12** Assume that  $v_i$  is not a vertex in subtree  $T_j$  and  $y_1, y_2 \in V(T_j)$ . If  $x$  is on path between  $y_1, y_2$  (see Fig. 5), then  $s_i(x, y_1) = s_i(x, y_2)$ . In this case, for each  $x \in V(T_j)$ , there is at most one  $y$  with different scenario  $s_i(x, y)$ .

**Lemma 3.13** If  $y_1, y_2$  are on path between  $x$  and  $v_i$  (see Fig. 6), then  $s_i(x, y_1) = s_i(x, y_2)$ .

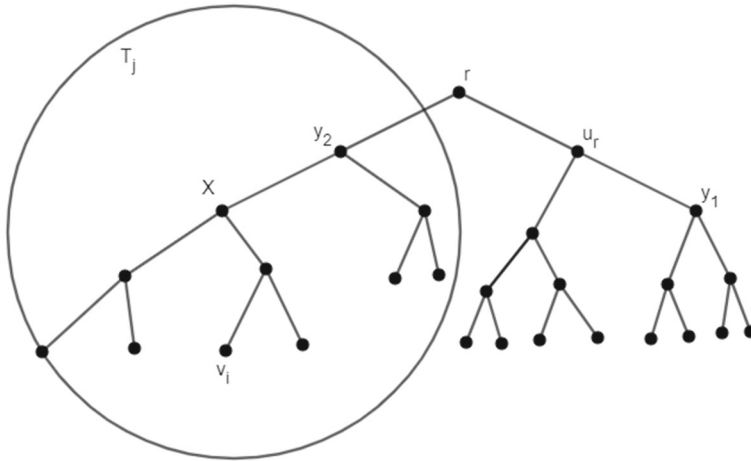


Fig. 4 Related graph of Lemma 3.11

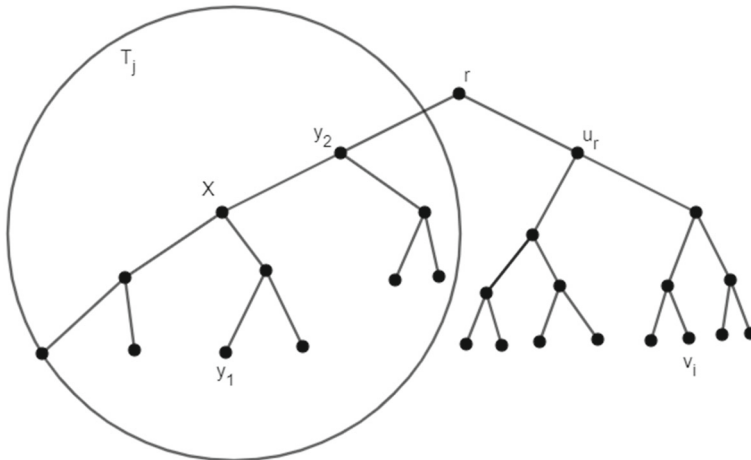


Fig. 5 Related graph of Lemma 3.12

Therefore, for each  $x$  on subtree  $t_j$ , it is sufficient to compute  $s_i(x, y)$  for at most  $O(\log n)$  number of vertex  $y$ . On the other hand, the number of  $s_i(x, y)$  is  $n$ . Then for each vertex  $x$  on tree  $T$ , we can compute at most  $O(n \log n)$  worst case scenarios. Each of these scenarios is calculated in linear time. Therefore, we conclude that

**Lemma 3.14** *The required time complexity to solve the robust centdian problem on tree networks with interval edge length is reduced to  $O(n^3 \log n)$ .*

**Proof** For each pair of vertices, the number of worst case scenarios are atmost  $O(\log n)$  and one of these scenarios is obtained in linear time. Therefore, the total time is reduced to  $O(n^3 \log n)$ . □

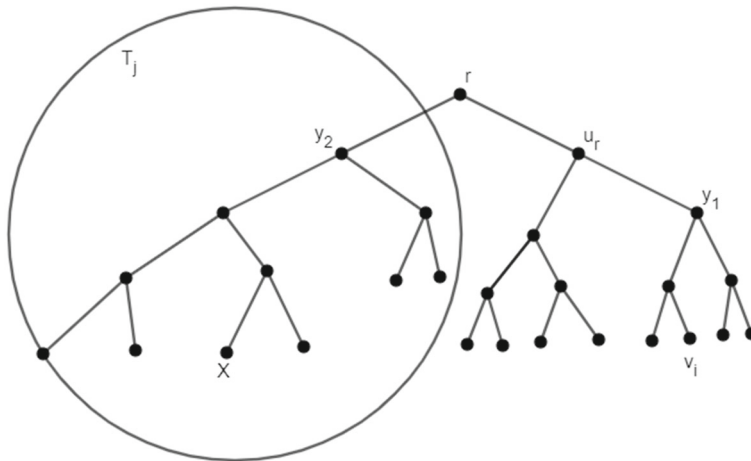


Fig. 6 Related graph of Lemma 3.13

## 4 Conclusions

In this paper, we considered the robust centroid location problem on tree network with interval edge lengths. Here, we used the minmax regret criterion. For simplicity, we decomposed the minmax regret centroid problem to two cases. First we solved the problem in  $O(n^4)$  time complexity. Then we decreased this time to  $O(n^3 \log n)$  with using the concept of weight-centroid vertex on tree networks.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- Aloulou, M. A., Kalai, R., & Vanderpooten, D. (2005). Minmax regret 1-center problem on a network with a discrete set of scenarios. *Cahiers de recherche en ligne du LAMSADE-Document*, (132).
- Averbakh, I. (2003). Complexity of robust single facility location problems on networks with uncertain edge lengths. *Discrete Applied Mathematics*, *127*(3), 505–522.
- Averbakh, I., & Berman, O. (2000). Algorithms for the robust 1-center problem on a tree. *European Journal of Operational Research*, *123*(2), 92–302.
- Averbakh, I., & Berman, O. (2000). Minmax regret median location on a network under uncertainty. *INFORMS Journal on Computing*, *12*(2), 104–110.
- Averbakh, I., & Berman, O. (2003). An improved algorithm for the minmax regret median problem on tree. *Networks*, *41*, 97–103.
- Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of Linear Programming problems contaminated with uncertain data. *Mathematical Programming*, *88*, 411–424.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*, 35–53.
- Burkard, R. E., & Dollani, H. (2002). A note on the robust 1-center problem on trees. *Annals of Operations research*, *110*(1–4), 69–82.
- Chen, B., & Lin, C. S. (1998). Minmax regret robust 1-median location on a tree. *Networks*, *31*, 93–103.

- Conde, E. (2008). A note on the minmax regret Centdian location on tree. *Operations Research Letters*, 36(2), 271–275.
- Korte, B., & Vygen, J. (2012). *Combinatorial optimization: Theory and algorithms* (Vol. 5). Springer.
- Ehrgott, M. (2005). *Multicriteria optimization* (p. 491). UK, vol: Springer.
- Hakimi, S. L. (1965). Optimal distribution of switching centers in a communications and some related graph-theoretic problems. *Operations Research*, 13(3), 462–475.
- Halpern, J. (1978). Finding minimal center-median convex combination of a graph. *Management Science*, 24(5), 535–544.
- Halpern, J. (1980). Duality in the centdian of a graph. *Operations Research*, 28(28), 722–735.
- Halpern, J. (1979). The location of a center-median convex combination on an undirected tree. *Journal of Regional Science*, 16(2), 237–245.
- Kariv, O., & Hakimi, S. L. (1979). An algorithmic approach to network location problems. I: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3), 513–538.
- Kouvelis, P., Vairaktarakis, G. L., & Yu, G. (1993). *Robust 1-median location on a tree in the presence of demand and transportation cost uncertainty*. Department of Industrial and systems Engineering: University of Florida.
- Li, H., Luo, T., Xu, Y., & Xu, J. (2018). Minmax regret vertex centdian location problem on in general dynamic networks. *Omega*, 75, 87–96.
- Mulvey, J. M., Vanderbei, R., & Zenios, S. A. (1995). Robust optimization of large-scale systems. *Operations Research*, 43, 264–281.
- Revelle, C. S., & Swain, R. W. (1970). Central facilities location. *Geographical Analysis*, 2(1), 30–42.
- Soyster, A. L. (1973). Technical note—Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operation Research*, 21(5), 1154–1157.
- Tamir, A., Perez-Brito, D., & Moreno-Perez, J. A. (1998). A polynomial algorithm for the p-centdian problem on a tree. *Network*, 32(4), 255–262.
- Toregas, C. R., Swain, R., & Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, 19(6), 1363–1373.
- Tamir, A. (1996). An  $O(pn^2)$  algorithm for the p-median and the related problems in tree graphs. *Operations Research Letters*, 19, 59–64.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.