



# The Uniform Cost Reverse 1-Centdian Location Problem on Tree Networks with Edge Length Reduction

Kien Trung Nguyen<sup>1</sup> · Wen Chean Teh<sup>2</sup>

Received: 17 April 2020 / Accepted: 6 May 2021 / Published online: 13 October 2021  
© Vietnam Academy of Science and Technology (VAST) and Springer Nature Singapore Pte Ltd. 2021

## Abstract

In location theory, the 1-centdian function is a convex combination of the 1-median and the 1-center functions. We consider the uniform cost reverse 1-centdian problem on networks, where edge lengths are reduced within a given budget such that the 1-centdian function at a prespecified point on the network is minimized. We first prove that the problem on general networks is NP-hard by reducing the set cover problem to it. Then, we focus on the special case of the problem on tree networks. Based on the strategy that we reduce either one edge or several edges simultaneously in each step to obtain an optimal solution, we develop a combinatorial algorithm that solves the corresponding problem on trees in quadratic time.

**Keywords** Combinatorial optimization · Location problem · Reverse optimization · Centdian function

**Mathematics Subject Classification (2010)** 90B10 · 90B80 · 90C27

## 1 Introduction

Classical location theory aims to find the optimal location of one or several new facilities. For the location problem concerning public services such as warehouses, supermarkets, etc., we use the median function. On the other hand, the center function is applied to model the location problem regarding emergency services such as hospitals and fire stations. Location problems on the plane and on networks have been intensively studied, for example, by Das et al. [9], Drezner and Hamacher [10], Eiselt and Marianov [11], and Kariv and Hakimi [19, 20]. Halpern [16, 17] further considered a convex combination of the median and the center

---

✉ Kien Trung Nguyen  
trungkien@ctu.edu.vn

Wen Chean Teh  
dasmenteh@usm.my

<sup>1</sup> Department of Mathematics, Teacher College, Can Tho University, Can Tho, Vietnam

<sup>2</sup> School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Malaysia

objective functions in his proposed locational model for the case where both public services and emergency services are taken into account, for instance, the location of a school. He also proposed a linear time algorithm for the problem on trees and a polynomial time algorithm for the problem on general networks. Since then, the centdian location problem has become an interesting topic in the field of locational analysis with numerous publications. A few of them are Brito et al. [26] on the 2-facility centdian problem, Tamir et al. [29] on the  $p$ -centdian problem on trees, Kang et al. [18] on the connected  $p$ -centdian problem on blockgraphs, and many references therein.

As a counterpart to the classical location problem, the previous researchers proposed two types of modern problems concerning adjusting the input data in order to change the structure of the original problem with respect to a task (given by a decision maker). One of them is the inverse location problem, which seeks to change the parameters at a minimum total cost so as to make prespecified facilities optimal in the modified network. The inverse location problem has been investigated due to their potential in real life applications and in theory (see [2, 3, 23, 25, 27]). On the other hand, the reverse location problem aims to modify parameters of a network subject to a given budget constraint such that the network is improved (decided by a decision maker) as much as possible. In the scope of this paper, we focus on the reverse location problem. Hence, we review the existing literature concerning the reverse location problem to ease the readers in the following discussion.

The reverse location problem first appeared in the seminal works of Berman et al. [5, 6] in terms of improving the minisum and minimax functions on a network by reducing the edge lengths. The authors considered the reverse 1-median problem on trees with a linear time algorithm and proved that the corresponding problem on networks is NP-hard. They also proposed a heuristic approach for the problem on general networks. Additionally, the authors proposed an iterative approach to the reverse 1-center problem on trees without detailed analysis on the complexity and the correctness of the algorithm. They also proved NP-hardness of the problem on general networks and proposed a heuristic approach as well. Since then, the reverse location problem has garnered interest among researchers in the location problem community. Reverse 1-center location problem on unweighted trees was investigated by Zhang et al. [30, 31]. They proposed minimum cut approaches for tackling the problem in  $O(n^2 \log n)$  time, where  $n$  is the number of vertices in the underlying tree. Later, Nguyen [24] considered the uniform-cost reverse 1-center problem on weighted trees and developed a quadratic algorithm for the problem. Meanwhile, concerning the reverse 1-median problem, Burkard et al. [7] proved further that the problem on networks is NP-hard and the optimal solution cannot be approximated by a constant number. They also devised a linear time algorithm to solve the corresponding problem on cycles. Also, Burkard et al. [8] showed that both the reverse 2-median problem on trees and the reverse 1-median problem on uni-cyclic graphs can each be reduced to an equivalent 2-median problem on paths. This result helped to solve the two mentioned problems in  $O(n \log n)$  time. For the reverse version of the obnoxious location problem, that is, one or several servers are not desirable in the network system, Alizadeh and Etemad [4] developed a linear time algorithm to solve the problem under the rectilinear norm or the Hamming distance. Furthermore, the reverse selective obnoxious center location problem on trees can be solved in quadratic time generally and in linear time for the uniform cost situation (see Etemad and Alizadeh [12]). Another type of network improving problem is the upgrading or downgrading problem. In this case, there is no prespecified facility and one or several facilities are optimally located after modifying the network. Interested reader can refer to Afrashteh et al. [1], Gassner [13, 14], and Sepasian [28] for algorithmic approaches related to the upgrading and downgrading problems.

From previous literature, we observe that the reverse 1-median and the reverse 1-center problems have already been investigated with some efficient solution algorithms found. However, the reverse 1-centdian problem has not been studied so far although the classical 1-centdian problem has been extensively studied. Due to the importance of the 1-centdian function in modeling real life problems, the corresponding reverse 1-centdian problem is promising in theory and in application as well. For example, let a network and a predetermined vertex, designating the location of a supplier, be given. In the network, each vertex plays the role of a demander and each edge represents the connection between two demanders. The size of the population of a demander is the associated vertex weight and the travel time between two demanders of the network is represented by the edge length. The efficiency of the given supplier is measured by the 1-centdian function according to its special purpose, serving as a public or emergency service like a school or a hospital. In order to improve the efficiency of the supplier, a budget is provided. However, the supplier cannot be moved to other locations due to security and cost reasons. Instead, we can shorten the travel time between adjacent vertices of the network by improving the edge capacities or modernizing the related technologies and this incurs relevant cost. Hence, the question is how to use the budget to improve the efficiency of the supplier in an optimal way. This real-life situation is indeed modeled as a reverse 1-centdian problem on networks.

We focus in this paper the algorithmic approach to the reverse 1-centdian problem on networks and organize the paper as follows. Section 2 introduces fundamental concepts of the reverse 1-centdian problem on networks. We also derive NP-hardness result for the problem in this section. In Section 3, we study the problem on tree networks and developed a novel  $O(n^2)$  algorithm for the problem on a tree with  $n$  vertices.

## 2 Problem Definition and Complexity

We first introduce some basic concepts concerning the 1-centdian location problem on a network (or graph). Let an undirected connected graph  $G = (V, E, w, \ell)$  be given with  $V$  ( $|V| = n$ ) and  $E$  being respectively the vertex set and the edge set. Furthermore, the function  $w : V \rightarrow \mathbb{R}_+$  maps each vertex  $v$  to a positive vertex weight, say  $w(v)$ . Also, the function  $\ell : E \rightarrow \mathbb{R}_+$  maps each edge  $e$  to a positive edge length, say  $\ell(e)$ . Each edge of the graph can be considered as a continuous interval containing points of the graph. The distance  $d_\ell(a, b)$  between two points  $a$  and  $b$  is the length of the shortest path  $P(a, b)$  connecting them with respect to  $\ell$ . The 1-centdian function at a point  $\rho$ , according to Halpern [16, 17], can be presented as a convex combination of the 1-median and the 1-center function at the same point, that is,

$$\lambda \sum_{v \in V} w(v)d_\ell(v, \rho) + (1 - \lambda) \max_{v \in V} \{w(v)d_\ell(v, \rho)\},$$

where  $\lambda \in (0, 1)$ . In this work, readers are assumed to be familiar with the 1-median function  $\sum_{v \in V} w(v)d_\ell(v, \rho)$  and the 1-center function  $\max_{v \in V} w(v)d_\ell(v, \rho)$  on a network  $G$ . Equivalently, the 1-centdian problem is to find a point  $\rho$  that minimizes

$$F(\rho) := \sum_{v \in V} w(v)d_\ell(v, \rho) + \alpha \max_{v \in V} w(v)d_\ell(v, \rho)$$

for  $\alpha := (1 - \lambda)/\lambda$ . Hence, from here onwards, we regard the function  $F(\rho)$  as the 1-centdian function.

Assuming that we have found a desired position on a network  $G$  to locate a facility and it can not be relocated due to security or logistics reasons. Then, the decision maker would like to improve the efficiency of the network by reducing the objective function, say the 1-centdian function, at the prespecified facility location as much as possible by reducing the distances between servers and customers within a given budget. The corresponding model is the so-called reverse 1-centdian problem on networks.

In order to formulate the model, let a prespecified point on  $G$ , where the server is located, be given. One always assumes that the point, denoted by  $v^*$ , is a vertex as otherwise, we split the edge that contains  $v^*$  into two edges to induce a new graph. We can reduce the distances from customers to  $v^*$  by reducing the length of each edge  $e$  by  $x(e)$ . The modified length of  $e$  is written as  $\tilde{\ell}(e) := \ell(e) - x(e)$ . Additionally, for each  $e \in E$ , the modified length  $\tilde{\ell}(e)$  is restricted to a given bound, say  $\tilde{\ell}(e) \in [\underline{\ell}(e), \ell(e)]$ . We set  $\bar{x}(e) := \ell(e) - \underline{\ell}(e)$  to be the largest possible modification of  $e$  in  $E$ . In real life situation, the distance between vertices are always positive, this implies that the length of each edge is also positive, that is,  $\underline{\ell}(e) > 0$  or  $\bar{x}(e) < \ell(e)$  for all  $e \in E$ . We denote by  $c(e)$  the (positive) cost to modify an edge  $e$  and it is proportional to the magnitude of modification. The total cost of modifying the edge lengths is limited within a given budget  $B$ . Let  $d_{\tilde{\ell}}$  be the distance function with respect to the new edge lengths  $\tilde{\ell}$ . Then, the reverse 1-centdian problem on  $G$  can be formulated as follows.

$$\begin{aligned} \min \tilde{F}(v^*) &:= \sum_{v \in V} w(v)d_{\tilde{\ell}}(v, v^*) + \alpha \max_{v \in V} w(v)d_{\tilde{\ell}}(v, v^*), \\ \text{s.t. } \tilde{\ell}(e) &= \ell(e) - x(e), \quad \forall e \in E, \\ \sum_{e \in E} c(e)x(e) &\leq B, \\ 0 \leq x(e) &\leq \bar{x}(e), \quad \forall e \in E. \end{aligned}$$

In this paper, we consider the uniform cost situation, that is,  $c(e) := C$  for all  $e \in E$ . Moreover, one can always assume that  $c(e) := 1$  for all  $e \in E$  as otherwise, we can reduce to this case with the new budget  $B' := B/C$ . With our assumption, the budget constraint is translated into  $\sum_{e \in E} x(e) \leq B$ . Let us further make an assumption that  $\sum_{e \in E} \bar{x}(e) > B$  since otherwise we can get a trivial optimal solution  $x(e) := \bar{x}(e)$  for all  $e \in E$ .

We first consider the complexity of the problem on general networks.

**Theorem 1** *The reverse 1-centdian problem is NP-hard even on unweighted networks.*

*Proof* We start with an instance of the Set Cover Problem (SC) given below:

Given: A finite set  $E = \{1, 2, \dots, n\}$ , a collection  $E_1, \dots, E_m$  of subsets of  $E$  whose union is  $E$ , and a number  $k \leq m$ .

Question: Does there exist  $k$  subsets in  $\{E_1, \dots, E_m\}$  such that their union is exactly the ground set  $E$ ?

The (SC) problem is NP-complete (see Garey and Johnson [15]).

The decision version of the reverse 1-centdian problem on networks (RCPN) is stated as follows:

Given: An instance of the unweighted network  $G = (V, E, w, \ell)$ , that is,  $w(v) = 1$  for all  $v \in V$ , with a prespecified vertex  $v^*$ , modification bounds  $\bar{x}(e)$  for all  $e$  in  $E$ , a budget  $B$ , and a value  $M$ .

Question: Does there exist an edge length modification such that the total cost is limited within  $B$  and the 1-centdian function at  $v^*$  that is at most  $M$ ?

We prove that (RCPN) is NP-hard by reducing an instance of (SC) to someone in polynomial time and prove that the answer to (SC) is ‘yes’ if and only if the answer to (RCPN) is ‘yes’. Given an instance of (SC), we construct an instance of (RCPN) as follows.

- The vertex set is  $V := \{v^*\} \cup \{u_i : i = 1, \dots, m\} \cup \{v_j : j = 1, \dots, n\}$  and the edge set is  $E := E_1 \cup E_2$ , where  $E_1 := \{(v^*, u_i) : i = 1, \dots, m\}$  and  $E_2 := \{(u_i, v_j) : i = 1, \dots, m, j = 1, \dots, n \text{ and } j \in E_i\}$ .
- The weight of each vertex in  $V$  is 1. The lengths are defined as  $\ell(e) := 2$  if  $e \in E_1$  and  $\ell(e) := 1$  if  $e \in E_2$ . Modification bounds are  $\bar{x}(e) = 1$  if  $e \in E_1$  and  $\bar{x}(e) = 0$  if  $e \in E_2$ .
- We set  $B := k$  and  $M := 2m + 2n - k + 2$ .
- Furthermore, we choose  $\alpha = 1$ , that is, the objective function can be written as

$$F(v^*) = \sum_{v \in V} d_\ell(v, v^*) + \max_{v \in V} d_\ell(v, v^*).$$

If the answer to (SC) is ‘yes’, we obtain a collection of subsets  $E_{i_1}, E_{i_2}, \dots, E_{i_k}$  such that  $\bigcup_{j=1}^k E_{i_j} = E$ . We set  $x(v^*, u_{i_j}) := \bar{x}(v^*, u_{i_j}) = 1$  for  $j = 1, \dots, k$  and other edge lengths are not modified. By elementary computations, the total cost is  $k$  and the 1-centdian function at  $v^*$  with respect to the modified graph is  $2m + 2n - k + 2$ . Hence, the answer to (RCPN) is ‘yes’.

Conversely, we assume that there exists a modification of the edge lengths within a budget  $k$  such that the objective function at  $v^*$  is at most  $2m + 2n - k + 2$ . As  $\sum_{e \in E_1} \bar{x}(e) = m > k$ , we can also assume that the modification cost is exactly  $k$ , that is,  $\sum_{e \in E_1} x(e) = k$ , because otherwise, we can reduce the edge lengths further without increasing the objective function at  $v^*$ . Based on the structure of the graph  $G$ , we obtain that

$$\sum_{v \in V_1} d_{\tilde{\ell}}(v, v^*) = \sum_{v \in V_1} d_\ell(v, v^*) - \sum_{e \in E_1} x(e) = 2m - k$$

and  $d_{\tilde{\ell}}(v, v^*) \geq 2$  for all  $v \in V_2$ . Therefore, we obtain

$$\sum_{v \in V} d_{\tilde{\ell}}(v, v^*) = \sum_{v \in V_1} d_{\tilde{\ell}}(v, v^*) + \sum_{v \in V_2} d_{\tilde{\ell}}(v, v^*) \geq 2m - k + 2n$$

and

$$\max_{v \in V} d_{\tilde{\ell}}(v, v^*) \geq 2.$$

This implies that the 1-centdian objective value is always at least  $2m + 2n - k + 2$ . By the hypothesis, the objective function is exactly  $2m + 2n - k + 2$  and thus  $d_{\tilde{\ell}}(v, v^*) = 2$  for all  $v \in V_2$ . Let us denote by  $\mathcal{I} := \{i \in \{1, \dots, m\} : x(v^*, u_i) = 1\}$ . Then, we get  $|\mathcal{I}| \leq k$ ,  $d_{\tilde{\ell}}(u_j, v^*) = 1$  if  $j \in \mathcal{I}$ , and  $d_{\tilde{\ell}}(u_j, v^*) > 1$  otherwise. As  $d_{\tilde{\ell}}(v, v^*) = 2$  for all  $v \in V_2$ , the shortest path from a vertex  $v \in V_2$  to  $v^*$  must contain a vertex  $u_i$  for some  $i \in \mathcal{I}$ . Hence, it implies that  $\bigcup_{i \in \mathcal{I}} E_i = E$ . In other words, the answer to (SC) is ‘yes’. □

By Theorem 1, we can hardly find an algorithm that solves the (RCPN) problem in polynomial time unless  $P = NP$ . It is interesting to further consider special cases of the problem with polynomial time solution algorithms. In the next section, we study the reverse 1-centdian location problem on tree networks.

### 3 Reverse 1-Centdian Problem on Trees

In this section, we focus on the reverse 1-centdian problem on a tree (RCPT), say  $T = (V, E, w, \ell)$ . In the given tree, the path  $P(v, v^*)$  connecting a vertex  $v$  and the prespecified vertex  $v^*$  is unique for each  $v$  in  $V$ . We can write  $d_{\ell}^-(v, v^*) := d_{\ell}(v, v^*) - \sum_{e \in P(v, v^*)} x(e)$  and hence the 1-centdian function is a decreasing function on the modification of edge lengths. Based on this fact, we deliver a trivial but useful result that helps to understand the structure of an optimal solution of the (RCPT) problem.

**Proposition 1** *The budget constraint holds with equality for every optimal solution of the (RCPT) problem.*

By Proposition 1, we can restate the budget constraint as  $\sum_{e \in E} x(e) = B$  and denote by

$$\mathcal{F} := \left\{ (x(e))_{e \in E} : \sum_{e \in E} x(e) = B \text{ and } 0 \leq x(e) \leq \bar{x}(e) \text{ for all } e \in E \right\},$$

the set of feasible solutions to the (RCPT) problem. The 1-centdian objective function is said to be improved by  $\mathcal{R}$  for  $(x(e))_{e \in E} \in \mathcal{F}$  if  $\mathcal{R} := F(v^*) - \tilde{F}(v^*)$ , where  $\tilde{F}(v^*)$  is the 1-centdian function at  $v^*$  with respect to the modification  $(x(e))_{e \in E}$ . As the aim of (RCPT) is to minimize the 1-centdian function  $\tilde{F}(v^*)$ , it is equivalent to finding a feasible solution  $(x(e))_{e \in E} \in \mathcal{F}$  which maximizes the improvement of the objective function.

#### 3.1 Problem with a Sufficiently Small Budget $B := \varepsilon$

From here onwards, we root the tree  $T$  at the prespecified vertex  $v^*$ . An edge  $e := (u, v)$ , written in this ordering, means that  $u$  is a parent of  $v$  or  $v$  is a child of  $u$ . Observe that the (length) reduction of an edge  $e = (u, v)$  can be shifted to an edge along the path  $P(u, v^*)$  if there exists an edge  $e' \in P(u, v^*)$  with  $\bar{x}(e') > 0$ . This shift helps to reduce the 1-centdian value at  $v^*$  even further without increasing the cost. An edge  $e = (u, v)$  with  $\bar{x}(e) > 0$  is, by definition, a *candidate edge* if no edge  $e' \in P(u, v^*)$  exists such that  $\bar{x}(e') > 0$ . In particular, if  $e = (v^*, v)$  and  $\bar{x}(e) > 0$ , then  $e$  is a candidate edge. Also, we denote by  $\mathcal{C}$  the set of candidate edges and focus on the edges in  $\mathcal{C}$  when considering edges to be reduced.

Let us denote by  $\mathcal{T}$  the maximal subtree of  $T$  with  $v^*$  in  $\mathcal{T}$  and without any candidate edge. Let  $\mu_1 := \max_{v \in V(\mathcal{T})} w(v)d_{\ell}(v, v^*)$  be the maximum weighted distance from vertices in  $\mathcal{T}$  to the vertex  $v^*$ . If  $\mu_1 = \max_{v \in V} w(v)d_{\ell}(v, v^*)$ , the 1-center part  $\max_{v \in V} w(v)d_{\ell}^-(v, v^*)$  cannot be reduced any further. Then, the (RCPT) problem reduces to an equivalent reverse 1-median problem on the same tree and can be solved in linear time by Berman et al. [5]. Therefore, we can assume that  $\mu_1 < \max_{v \in V} w(v)d_{\ell}(v, v^*)$  with respect to the current subtree  $\mathcal{T}$  and the tree  $T$ . This condition can be checked efficiently as finding  $\mu_1$  and  $\max_{v \in V} w(v)d_{\ell}(v, v^*)$  can be done in linear time by a breadth first search (BFS) approach.

Now we aim to find which edges in  $\mathcal{C}$  should be reduced to attain the maximum improvement for a sufficiently small fixed cost  $\varepsilon$ . For each candidate edge  $e := (u, v) \in \mathcal{C}$ , we denote by  $T_e$  the subtree rooted at  $v$  and  $W(T_e)$  the total weight of vertices in  $T_e$ . As the 1-centdian value is dependent on both the 1-median function  $\sum_{u \in V} w(u)d_{\ell}^-(u, v^*)$  and the 1-center function  $\max_{u \in V} w(u)d_{\ell}^-(u, v^*)$ , we now analyze the reduction of these functions with respect to edge length modification.

If we reduce a single edge  $e$  in  $\mathcal{C}$  by  $\varepsilon$ , it is easy to verify that the 1-median function is reduced by  $W(T_e)\varepsilon$ . We set  $e_{\max} := \arg \max_{e:=(u,v) \in \mathcal{C}} W(T_e)$ , that is, a specific candidate edge for which the corresponding tree has the largest total vertex weights. Then, reducing  $e_{\max}$  by  $\varepsilon$  yields the maximum reduction of the 1-median function, that is, by  $W(T_{e_{\max}})\varepsilon$ .

For the 1-center function, we set

$$v_e := \arg \min \left\{ w(v) : v \in T_e \text{ and } w(v)d_\ell(v^*, v) = \max_{u \in V(T_e)} w(u)d_\ell(v^*, u) \right\},$$

and call it an *associated vertex* in  $T_e$ . Note that, for a sufficiently small reduction of  $e$ , an associated vertex  $v_e$  satisfies  $w(v_e)d_\ell(v_e, v^*) = \max_{u \in V(T_e)} w(u)d_\ell(u, v^*)$ .

In Fig. 1, for a candidate edge  $e = (v^*, v_1)$ , both  $v_3$  and  $v_4$  have maximum weighted distance to  $v^*$ , namely 24. As  $w(v_3) < w(v_4)$ , we set  $v_e := v_3$ .

Let us abbreviate

$$\mathcal{M} := \bigcup_{e \in \mathcal{C}} \left\{ v_e : w(v_e)d_\ell(v^*, v_e) = \max_{u \in V} w(u)d_\ell(v^*, u) \right\}$$

the set of all associated vertices  $v_e$  such that the 1-center value of  $T$  is attained at these vertices. Furthermore, let  $\mathcal{C}(\mathcal{M}) := \{e \in \mathcal{C} : v_e \in \mathcal{M}\}$  be the set of candidate edges with corresponding vertices in  $\mathcal{M}$ . In order to reduce the 1-center function, it is required to simultaneously reduce all the edges in  $\mathcal{C}(\mathcal{M})$ . Therefore, we now analyze how much the 1-centdian function improves when all the edges in  $\mathcal{C}(\mathcal{M})$  (and no other edges) are reduced according to some modification satisfying a natural constraint. We will see that this analysis leads essentially to our main result.

Suppose  $e_1, e_2, \dots, e_k$  are all the edges in  $\mathcal{C}(\mathcal{M})$ . By definition,  $v_{e_i} = \arg \max_{u \in V} w(u)d_\ell(u, v^*)$  for all  $i = 1, \dots, k$ . Suppose  $e_i$  is reduced by  $x(e_i)$  for each  $i = 1, \dots, k$  given a sufficiently small budget  $\varepsilon$ . To achieve maximum improvement, we may assume that  $\sum_{j=1}^k x(e_j) = \varepsilon$ . More importantly, we suppose our edge length modification satisfies the following natural special condition:

$$x(e_1)w(v_{e_1}) = x(e_2)w(v_{e_2}) = \dots = x(e_k)w(v_{e_k}). \tag{1}$$

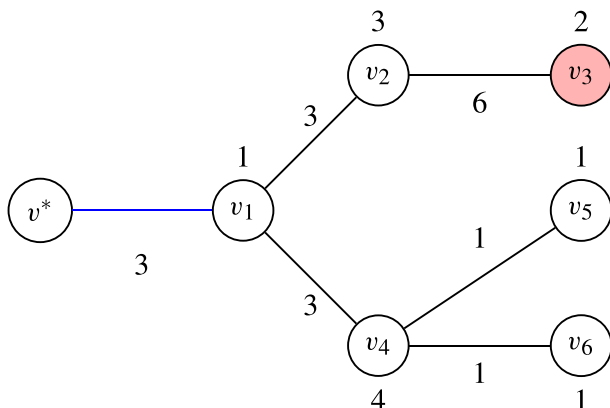


Fig. 1 The associated vertex in  $T_{(v^*, v_1)}$  (in red)

We denote by  $P := \prod_{j=1}^k w(v_{e_j})$ ,  $w'(v_{e_i}) := \prod_{j=1, j \neq i}^k w(v_{e_j})$ , and  $S := \sum_{i=1}^k w'(v_{e_i})$ . By elementary computation, for each  $i = 1, \dots, k$ , we get  $x(e_i) = \frac{w'(v_{e_i})}{S} \varepsilon$ , or in other words

$$x(e_i) = \frac{\varepsilon}{w(v_{e_i}) \sum_{j=1}^k \frac{1}{w(v_{e_j})}}. \tag{2}$$

Now, our constraint immediately implies that

$$\max_{u \in V} w(u) d_{\tilde{c}}(u, v^*) = \max_{u \in V} w(u) d_{\ell}(u, v^*) - x(e_1)w(v_{e_1}).$$

Hence, the corresponding improvement of the 1-centdian function can be written as

$$\begin{aligned} \underbrace{\sum_{i=1}^k x(e_i)W(T_{e_i})}_{\text{1-median}} + \underbrace{\alpha x(e_1)w(v_{e_1})}_{\text{1-center}} &= \frac{1}{S} \left( \sum_{i=1}^k w'(v_{e_i})W(T_{e_i}) + \alpha P \right) \varepsilon \\ &= \frac{P}{S} \sum_{i=1}^k \left( \frac{W(T_{e_i})}{w(v_{e_i})} + \alpha \right) \varepsilon \\ &= \frac{1}{\sum_{i=1}^k \frac{1}{w(v_{e_i})}} \left( \sum_{i=1}^k \frac{W(T_{e_i})}{w(v_{e_i})} + \alpha \right) \varepsilon. \end{aligned}$$

Let

$$\Delta_{\mathcal{M}} := \left( \sum_{e \in \mathcal{C}(\mathcal{M})} \frac{W(T_e)}{w(v_e)} + \alpha \right) / \left( \sum_{e \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_e)} \right) \tag{3}$$

and we can interpret  $\Delta_{\mathcal{M}}$  as the efficiency of reducing all edges in  $\mathcal{C}(\mathcal{M})$  according to (1). Moreover, we can trace back the reduction of the 1-centdian function based on the modifications of the edges in  $\mathcal{C}(\mathcal{M})$  and the efficiency  $\Delta_{\mathcal{M}}$ . Indeed, by (2) we can write  $\varepsilon := \sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_{e'})} w(v_e) x(e)$  for any edge  $e$  in the set  $\mathcal{C}(\mathcal{M})$ . The corresponding reduction of the 1-centdian function is thus presented by

$$\sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_{e'})} \Delta_{\mathcal{M}} w(v_e) x(e) \tag{4}$$

for any edge  $e \in \mathcal{C}(\mathcal{M})$ . Equations (3) and (4) can be calculated in  $O(|\mathcal{C}(\mathcal{M})|)$  time. As  $|\mathcal{C}(\mathcal{M})| \leq n$ , the corresponding computation costs linear time in the worst case scenario.

We are now ready for our main result of this section.

**Proposition 2** *Let a sufficiently small budget  $\varepsilon$  be given. If  $\Delta_{\mathcal{M}} > W(T_{e_{\max}})$ , we reduce each edge in  $\mathcal{C}(\mathcal{M})$  according to (2) to obtain the maximum reduction of the 1-centdian function, namely  $\Delta_{\mathcal{M}}\varepsilon$ . Otherwise, if  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ , we reduce the edge  $e_{\max}$  to obtain the maximum reduction of the 1-centdian function, namely  $W(T_{e_{\max}})\varepsilon$ .*

*Proof* Suppose that  $\Delta_{\mathcal{M}} > W(T_{e_{\max}})$  and the edges  $e_1, e_2, \dots, e_k$  are reduced by  $x(e_1), x(e_2), \dots, x(e_k)$ , where  $\sum_{i=1}^k x(e_i) = \varepsilon$ . First, we prove that to yield the maximum reduction of the 1-centdian function, all edges in  $\mathcal{C}(\mathcal{M})$  must be reduced or else it is better off just by reducing the single edge  $e_{\max}$ . Indeed, assume not all edges in  $\mathcal{C}(\mathcal{M})$



are reduced. Then the 1-center part  $\max_{v \in V} w_v d_{\tilde{\mathcal{C}}}^-(v^*, v)$  in the 1-centdian function is not reduced. Hence, the improvement is at most

$$\sum_{i=1}^k x(e_i)W(T_{e_i}) \leq W(T_{e_{\max}}) \sum_{i=1}^k x(e_i) \leq W(T_{e_{\max}}) \varepsilon.$$

Next, we prove that assuming we reduce exactly all edges in  $\mathcal{C}(\mathcal{M})$ , then the modification satisfying condition (1) yields the maximum reduction to the 1-centdian function, namely  $\Delta_{\mathcal{M}}\varepsilon$  according to our analysis before. Suppose  $e_1, e_2, \dots, e_k$  are all the edges in  $\mathcal{C}(\mathcal{M})$  but condition (1) does not hold. Without loss of generality,  $w(v_{e_1})x(e_1) \leq w(v_{e_2})x(e_2) \leq \dots \leq w(v_{e_k})x(e_k)$ . Let  $i$  be the smallest index such that  $w(v_{e_i})x(e_i) < w(v_{e_{i+1}})x(e_{i+1})$ . Then, we consider  $x'(e_{i+1}), \dots, x'(e_k)$  with  $x'(e_s) < x(e_s)$  for all  $s = i + 1, \dots, k$  such that

$$\begin{aligned} w(v_{e_1})x(e_1) &= w(v_{e_2})x(e_2) = \dots = w(v_{e_i})x(e_i) = w(v_{e_{i+1}})x'(e_{i+1}) \\ &= w(v_{e_{i+2}})x'(e_{i+2}) = \dots = w(v_{e_k})x'(e_k). \end{aligned}$$

We set  $\varepsilon_1 := \sum_{r=1}^i x(e_r) + \sum_{s=i+1}^k x'(e_s)$  and  $\varepsilon_2 := \sum_{s=i+1}^k [x(e_s) - x'(e_s)] = \varepsilon - \varepsilon_1$ . By the analysis before, the improvement of the 1-centdian function is

$$\begin{aligned} \Delta_{\mathcal{M}}\varepsilon_1 + \sum_{s=i+1}^k W(T_{e_s})[x(e_s) - x'(e_s)] &\leq \Delta_{\mathcal{M}}\varepsilon_1 + W(T_{e_{\max}})\varepsilon_2 \\ &< \Delta_{\mathcal{M}}(\varepsilon_1 + \varepsilon_2) = \Delta_{\mathcal{M}}\varepsilon. \end{aligned}$$

Finally, it remains to observe that if we reduce all edges in  $\mathcal{C}(\mathcal{M})$  according to condition (1) with budget  $\varepsilon_1$  and also some other edges  $e'_1, \dots, e'_j$  that are not in  $\mathcal{C}(\mathcal{M})$  with budget  $\varepsilon_2$ , then the improvement of the 1-centdian function is

$$\begin{aligned} \Delta_{\mathcal{M}}\varepsilon_1 + \sum_{s=1}^j W(T_{e'_s})x(e'_s) &\leq \Delta_{\mathcal{M}}\varepsilon_1 + W(T_{e_{\max}}) \sum_{s=1}^j x(e'_s) \\ &< \Delta_{\mathcal{M}}(\varepsilon_1 + \varepsilon_2) = \Delta_{\mathcal{M}}\varepsilon. \end{aligned}$$

Therefore, our proof of the first part is complete.

Now, suppose  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ . The preceding case analysis can be easily modified to show that the maximum reduction of the 1-centdian function is  $W(T_{e_{\max}})\varepsilon$ . This can be achieved by reducing the edge  $e_{\max}$  by  $\varepsilon$ , noting that  $\mathcal{C}(\mathcal{M}) \neq \{e_{\max}\}$  (see the next remark). □

*Remark 1* If  $\mathcal{C}(\mathcal{M}) = \{e_{\max}\}$ , without relying on Proposition 2, it is obvious that reducing  $e_{\max}$  by  $\varepsilon$  leads to the maximum reduction of the 1-centdian function, namely,  $(W(T_{e_{\max}}) + \alpha w(v_{e_{\max}}))\varepsilon$ . In this case,  $\Delta_{\mathcal{M}} = W(T_{e_{\max}}) + \alpha w(v_{e_{\max}}) > W(T_{e_{\max}})$  and thus the choice of reduction is consistent with Proposition 2.

*Remark 2* As the 1-center function is continuous on the edge lengths, we can choose  $\varepsilon$  to be sufficiently small such that  $\mathcal{C}, \mathcal{M}$ , and  $\mathcal{C}(\mathcal{M})$  do not change in the perturbed tree.

### 3.2 A Greedy Iterative Approach

We previously discussed which edges should be reduced for the maximum improvement of the 1-centdian function under a sufficiently small cost  $\varepsilon$  once  $\mathcal{C}(\mathcal{M})$  is known. Obviously, the set  $\mathcal{C}(\mathcal{M})$  depends directly on the sets  $\mathcal{C}$  and  $\mathcal{M}$ . Throughout the reduction of edge

lengths, the sets  $\mathcal{C}$  and  $\mathcal{M}$  may change. Hence, now we investigate the situations where one of the sets  $\mathcal{C}$  and  $\mathcal{M}$  changes.

**Case 1. Where the set  $\mathcal{C}$  of Candidate Edges Changes** Assuming that an edge  $e$  (which is either  $e_{\max}$  or is in  $\mathcal{C}(\mathcal{M})$ ) is reduced by  $\bar{x}(e)$ , its updated value of  $\bar{x}(e)$  is 0 and thus  $e$  is not a candidate edge anymore. In this case, we apply a (BFS) approach to find a new candidate edge(s) within the subtree  $T_e$  to update  $\mathcal{C}$  in  $O(|T_e|)$  time. Other related data must be updated accordingly as described below.

**Part 1.** If  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ , we reduce the edge  $e := e_{\max}$  and the 1-center function  $\max_{v \in V} w(v)d_{\tilde{\mathcal{C}}}(v, v^*)$  is not reduced. If there is a vertex  $v$  in  $T_{e_{\max}} \cap \mathcal{M}$ , we set  $\mathcal{M} := \mathcal{M} \setminus \{v\}$  and  $\mathcal{C}(\mathcal{M}) := \mathcal{C}(\mathcal{M}) \setminus \{e_{\max}\}$ . Otherwise, these sets  $\mathcal{M}$  and  $\mathcal{C}(\mathcal{M})$  do not change. Moreover, we update  $e_{\max}$ , the subtree  $\mathcal{T}$ , and the value  $\mu_1$  according to the new candidate set  $\mathcal{C}$ . To summarize, we have to update  $\mathcal{C}$ ,  $\mathcal{M}$ ,  $\mathcal{C}(\mathcal{M})$ ,  $\mathcal{T}$ ,  $\mu_1$ ,  $e_{\max}$ , and  $\Delta_{\mathcal{M}}$  accordingly. The complexity of this part is  $O(|T_{e_{\max}}| + |\mathcal{C}(\mathcal{M})|)$  due to the (BFS) approach and the complexity of computing  $\Delta_{\mathcal{M}}$ . As both  $|T_{e_{\max}}|$  and  $|\mathcal{C}(\mathcal{M})|$  are at most  $n$ , the complexity of this part is linear.

We study an instance in Fig. 2 with  $\alpha = 1$  and  $\mathcal{C} = \{(v_1, v_2), (v_1, v_4)\}$ . Furthermore,  $\mathcal{T} = \{v^*, v_1\}$  with  $\mu_1 = 2$ . We can compute  $e_{\max} = \{(v_1, v_4)\}$  with  $W(T_{e_{\max}}) = 13$ . As  $\mathcal{M} = \{v_3, v_5\}$  and  $\mathcal{C}(\mathcal{M}) = \mathcal{C}$ , we obtain  $\Delta_{\mathcal{M}} = 23/3 < W(T_{e_{\max}})$ . Thus, we reduce  $e_{\max} = \{(v_1, v_4)\}$ . Assuming that  $\bar{x}(v_1, v_2) = \bar{x}(v_1, v_4) = \bar{x}(v_4, v_5) = \bar{x}(v_4, v_6) = 1$ , after reducing  $(v_1, v_4)$  by 1, we update  $\mathcal{C} = \{(v_1, v_2), (v_4, v_5), (v_4, v_6)\}$ ,  $\mathcal{M} = \{v_3\}$ ,  $\mathcal{C}(\mathcal{M}) = \{(v_1, v_2)\}$ ,  $\mathcal{T} = \{v^*, v^1, v^4\}$ ,  $\mu_1 = 15$ ,  $e_{\max} = (v_4, v_5)$ , and  $\Delta_{\mathcal{M}} = 5$ .

**Part 2.** If  $\Delta_{\mathcal{M}} > W(T_{e_{\max}})$ , we reduce all edges in  $\mathcal{C}(\mathcal{M})$  according to (2). Suppose there is an edge  $e \in \mathcal{C}(\mathcal{M})$  with  $x(e) = \bar{x}(e)$ . In this situation, both the 1-median and the 1-center parts are reduced. Thus, we have to update the set  $\mathcal{C}$ ,  $\mathcal{M}$  and  $\mathcal{C}(\mathcal{M})$  by applying the (BFS) approach on the subtree  $T_e$ . Also, we update the subtree  $\mathcal{T}$  and the corresponding value  $\mu_1$  as some edges can not be further reduced. To summarize, we have to update  $\mathcal{C}$ ,  $\mathcal{M}$ ,  $\mathcal{C}(\mathcal{M})$ ,  $\mathcal{T}$ ,  $\mu_1$ ,  $e_{\max}$ , and  $\Delta_{\mathcal{M}}$ . The complexity of this part is  $O(|T_e| + |\mathcal{C}(\mathcal{M})|)$  or linear time because both  $|T_e|$  and  $|\mathcal{C}(\mathcal{M})|$  are at most  $n$ .

For example, we consider Fig. 3 with  $\alpha = 1$  and  $\mathcal{C} = \{(v_1, v_2), (v_1, v_4)\}$ . We can compute  $e_{\max} = \{(v_1, v_4)\}$  with  $W(T_{e_{\max}}) = 6$ . As  $\mathcal{M} = \{v_3, v_5\}$  and  $\mathcal{C}(\mathcal{M}) = \mathcal{C}$ ,

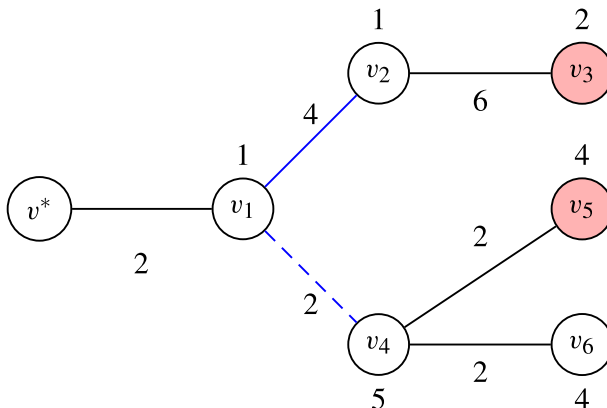


Fig. 2 An instance to illustrate Part 1 of Case 1

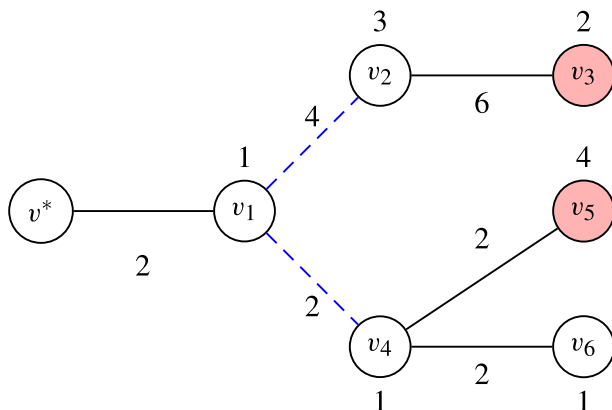


Fig. 3 An instance to illustrate Part 2 of Case 1

we obtain  $\Delta_{\mathcal{M}} = 20/3 > W(T_{e_{\max}})$ . Thus, we reduce edges in  $\mathcal{C}(\mathcal{M})$ . For  $\bar{x}(v_1, v_2) = \bar{x}(v_1, v_4) = \bar{x}(v_4, v_5) = \bar{x}(v_4, v_6) = 1$ , we simultaneously reduce  $(v_1, v_2)$  and  $(v_1, v_4)$  by 1 and  $1/2$ . As  $x(v_1, v_2) = \bar{x}(v_1, v_2)$ , we update  $\mathcal{C} = \{(v_1, v_4), (v_2, v_3)\}$ ,  $\mathcal{M} = \{v_3, v_5\}$ ,  $\mathcal{C}(\mathcal{M}) = \{(v_1, v_4), (v_2, v_3)\}$ ,  $e_{\max} = (v_1, v_4)$ , and  $\Delta_{\mathcal{M}} = 14/3$ .

**Case 2. Where the Set  $\mathcal{M}$  of Associated Vertices Changes** As the set  $\mathcal{M}$  is related to the 1-center function, we assume in this case that all edges in  $\mathcal{C}(\mathcal{M})$  are reduced and we consider an edge  $e$  in the set.

**Part 1** We investigate when the vertex  $v_e$  can change from one to another inside a subtree  $T_e$ . From the analysis in Nguyen [24], we know that  $v_e$  shifts to  $v$  if  $v$  is not in  $P(v_e, v^*)$  and  $w(v) < w(v_e)$  while the modification of  $e$  is  $x^v(e)$  satisfying  $w(v_e)(d_\ell(v^*, v_e) - x^v(e)) = w(v)(d_\ell(v^*, v) - x^v(e))$ , that is,

$$x^v(e) := \frac{w(v_e)d_\ell(v^*, v_e) - w(v)d_\ell(v^*, v)}{w(v_e) - w(v)}.$$

For each vertex  $v \in V(T_e)$  such that  $v$  is either in  $P(v_e, v^*)$  or  $w(v) \geq w(v_e)$ , we set  $x^v(e) := +\infty$ . Then, we denote by  $x^C(e) := \min_{v \in V(T_e)} \{x^v(e)\}$  the minimum modification of  $e$  such that  $v_e$  changes. Furthermore, if we modify  $e$  by  $x^C(e)$ , then we update  $\mathcal{M} := (\mathcal{M} \setminus \{v_e\}) \cup \{v\}$  for a vertex  $v$  satisfying

$$v := \arg \min \left\{ w(v') : v' \in T_e \text{ and } w(v')d_{\tilde{\ell}}(v', v^*) = \max_{v'' \in V} w(v'')d_{\tilde{\ell}}(v'', v^*) \right\}$$

and  $x^v(e) = x^C(e)$ . To compute  $x^C(e)$  for  $e \in \mathcal{C}(\mathcal{M})$ , we have to compute  $x^v(e)$  for all  $v \in T_e$  in constant time. Hence, the complexity for computing  $x^C(e)$  is  $O(|T_e|) = O(n)$  time.

For example, say we reduce the edge  $(v_1, v_2)$  in Fig. 4, then  $x^{v_3}((v_1, v_2)) = 2$  and  $x^{v_4}((v_1, v_2)) = 6$ . Thus, we derive  $x^C((v_1, v_2)) = 2$ . If we reduce the length of  $(v_1, v_2)$  by 2, the corresponding set  $\mathcal{M} = \{v_3\}$ .

**Part 2** We investigate the cases where new vertices in  $T \setminus \bigcup_{e \in \mathcal{C}(\mathcal{M})} T_e$  are added to  $\mathcal{M}$ . Let  $\mu_2 := \max\{w_{v_e}d_\ell(v_e, v^*) : e \in \mathcal{C}(\mathcal{M})\}$  and  $\mu := \max\{\mu_1, \mu_2\}$ , where we know  $\mu_1 := \max\{w_v d_\ell(v, v^*) : v \in \mathcal{T}\}$  in the previous section. We observe that  $\mu$  is not reduced

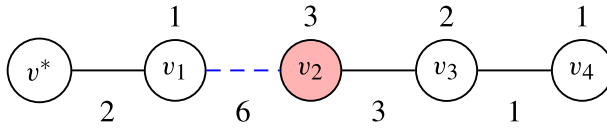


Fig. 4 An instance to illustrate Part 1 of Case 2

for the reduction of edges in  $\mathcal{C}(\mathcal{M})$ . In the case where we reduce all edges of  $\mathcal{C}(\mathcal{M})$ , if an edge  $e \in \mathcal{C}(\mathcal{M})$  is reduced by  $x^M(e)$  satisfying  $w(v_e)(d(v^*, v_e) - x^M(e)) = \mu$ , that is,

$$x^M(e) := \frac{w(v_e)d(v^*, v_e) - \mu}{w(v_e)},$$

then we get into one of the following two situations. If  $\mu_1 < \mu_2$ , there exists one or several new vertices  $v_e$  for  $e \in \mathcal{C} \setminus \mathcal{C}(\mathcal{M})$  that are added into  $\mathcal{M}$  and the corresponding edges are added to  $\mathcal{C}(\mathcal{M})$ . If  $\mu_1 \geq \mu_2$ , there exist a vertex  $u$  in  $\mathcal{T}$  with  $u := \arg \max_{v \in V} w(v)d_{\mathcal{T}}(v, v^*)$  in the modified tree  $T$ . As the 1-center part cannot be reduced any further from here on, the (RCPT) problem reduces to the reverse 1-median problem on the modified tree.

We consider Fig. 5 to illustrate this part for the case  $\mu_1 < \mu_2$ . For  $\mathcal{C} = \{(v_1, v_2), (v_1, v_4)\}$ , we know that  $(v_1, v_2)$  in  $\mathcal{M}$ . We obtain  $\mu = \mu_2 = 18$  and  $x^M(e) := 1$ . If we reduce  $(v_1, v_2)$  by 1, then  $\mathcal{M} := \{v_3, v_4\}$ .

In Fig. 6, we get  $\mu_1 > \mu_2$ . If we reduce  $(v_1, v_2)$  by 4, then

$$w(v_1)d(v_1, v^*) = \max_{v \in V} w(v)d_{\mathcal{T}}(v, v^*).$$

The 1-center part in the 1-centdian function can not be improved any further.

We have already analyzed the two cases regarding the change of  $\mathcal{C}$  or  $\mathcal{M}$ . Therefore, an edge  $e$  should be reduced by the minimum of the mentioned amounts in Case 1 and Case 2 as well as adapting to the budget constraint. Precisely, assuming we reduce the edges in  $\mathcal{C}(\mathcal{M})$  according to (1) subject to the budget constraint, that is,  $\sum_{e \in \mathcal{C}(\mathcal{M})} x(e) \leq B$ , then by (2),  $x(e) \leq \frac{1}{w(e) \sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(e')}} B$  must hold for each  $e \in \mathcal{C}(\mathcal{M})$ . The following result states the minimum reduction of an edge  $e$  according to Proposition 2 such that the set  $\mathcal{C}$  or  $\mathcal{M}$  changes and the budget constraint holds.

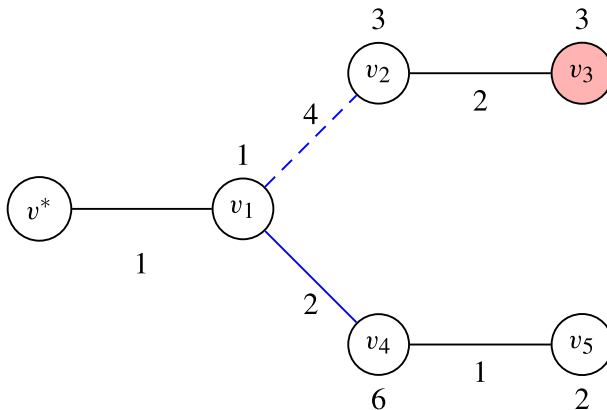
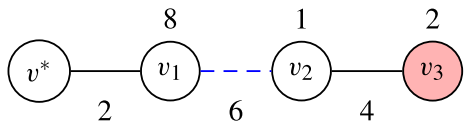


Fig. 5 An instance to illustrate Part 2 of Case 2 with  $\mu_1 < \mu_2$

**Fig. 6** An instance to illustrate Part 2 of Case 2 with  $\mu_1 \geq \mu_2$



**Proposition 3** If  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ , the minimum reduction of the edge  $e_{\max}$  such that the set  $\mathcal{C}$  changes or the budget is fully used is

$$\delta(e_{\max}) := \min \{ \bar{x}(e_{\max}), B \}. \tag{5}$$

Otherwise, if  $\Delta_{\mathcal{M}} > W(T_{e_{\max}})$ , the minimum reduction of an edge  $e$  in  $\mathcal{C}(\mathcal{M})$  such that the set  $\mathcal{C}$  or  $\mathcal{M}$  changes or the budget is fully used, is

$$\delta(e) := \min \left\{ \bar{x}(e), x^{\mathcal{C}}(e), x^{\mathcal{M}}(e), \frac{1}{w(v_e) \sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_{e'})}} B \right\}. \tag{6}$$

*Proof* Based on Proposition 2, we focus on reducing  $e_{\max}$  if  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ . As maximum possible reduction of the edge  $e_{\max}$  is  $\bar{x}(e_{\max})$  and is bounded by the budget, we get the desired result. Otherwise, if  $\Delta_{\mathcal{M}} > W(T_{e_{\max}})$ , we reduce all edges in  $\mathcal{C}(\mathcal{M})$ . Then, we get the result due to the analysis in Part 2 of Case 1 and the whole Case 2.  $\square$

By Proposition 3, we can identify the minimum reduction of an edge such that  $\mathcal{C}$  or  $\mathcal{M}$  changes. If  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$ , reducing the edge  $e_{\max}$  by  $x(e_{\max}) = \delta(e_{\max})$  yields the reduction of the 1-centdian function by  $W(T_{e_{\max}})\delta(e_{\max})$ . Otherwise, we can compute the minimum reduction of the 1-centdian function such that  $\mathcal{C}$  or  $\mathcal{M}$  changes based on (4) and Proposition 3.

**Proposition 4** If we reduce all edges in  $\mathcal{C}(\mathcal{M})$ , then the minimum reduction of the 1-centdian function which leads to the change of  $\mathcal{C}$  or  $\mathcal{M}$  is

$$\kappa := \sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_{e'})} \Delta_{\mathcal{M}} \min_{e \in \mathcal{C}(\mathcal{M})} \{ w(v_e) \delta(e) \}. \tag{7}$$

*Proof* By the formula for the reduction of the 1-centdian function given by (4) and the minimum reduction of an edge in  $\mathcal{C}(\mathcal{M})$  that leads to the change of  $\mathcal{C}$  or  $\mathcal{M}$  given by Proposition 3, we can trivially identify the minimum reduction of the 1-centdian function as in (7).  $\square$

Based on (7), we can trace back the reduction of each edge  $e$  in  $\mathcal{C}(\mathcal{M})$  by (2).

In the next page, we present a combinatorial algorithm for the (RCPT) problem. The idea of the algorithm is to reduce the length of the edge  $e_{\max}$  or all edges in  $\mathcal{C}(\mathcal{M})$  according to Proposition 3 and update necessary information in each iteration according to the analysis in Case 1 and Case 2. We repeat the iterations until the given budget is fully used. By

Proposition 2, we know that the 1-centdian value is reduced as much as possible. Hence, we finally get the maximum reduction of the objective function after completing Algorithm 1.

---

**Algorithm 1** (Solves the reverse centdian problem on a tree  $T = (V, E, w, \ell)$ ).

---

- 1: **Input:** An instance of the (RCPT) problem with the prespecified vertex  $v^*$ .
  - 2: Compute  $W(T_e)$  for all  $e \in E$  and set  $\mathcal{R} := 0$ .
  - 3: Identify  $\mathcal{C}, \mathcal{T}, \mathcal{M}, \mathcal{C}(\mathcal{M}), e_{\max}$ ; compute  $\Delta_{\mathcal{M}}$  (see (3)); compute  $\mu_1, x^{\mathcal{C}}(e), x^{\mathcal{M}}(e)$  (see Case 2).
  - 4: Compute the current 1-center value  $\theta := \max_{v \in V} w(v)d_{\ell}(v, v^*)$ .
  - 5: **while**  $B > 0$  **do**
  - 6:     **if**  $\Delta_{\mathcal{M}} \leq W(T_{e_{\max}})$  **then**
  - 7:         Reduce  $e_{\max}$  by  $x(e_{\max}) = \delta(e_{\max}) := \min\{\bar{x}(e_{\max}), B\}$  (see (5)) and  $\mathcal{R} := \mathcal{R} + W(T_{e_{\max}})x(e_{\max})$ .
  - 8:         Update  $\bar{x}(e_{\max}) := \bar{x}(e_{\max}) - \delta(e_{\max})$ .
  - 9:         Update the set  $\mathcal{C}, \mathcal{C}(\mathcal{M}), \mathcal{T}, \mu_1$  and the edge  $e_{\max}$  (see Part 1 of Case 1).
  - 10:     **else**
  - 11:         Compute  $\delta(e)$  for all  $e \in \mathcal{C}(\mathcal{M})$  (see (6)) and  $\kappa$  (see (7)).
  - 12:         Update  $\mathcal{R} := \mathcal{R} + \kappa$ .
  - 13:         Reduce each edge  $e$  in  $\mathcal{C}(\mathcal{M})$  by  $x(e) := \frac{\sum_{e' \in \mathcal{C}(\mathcal{M})} \frac{1}{w(v_{e'})} \kappa}{w(v_e)}$  (see (2)).
  - 14:         Update  $B := B - \sum_{e \in \mathcal{C}(\mathcal{M})} x(e)$ .
  - 15:         Update the current 1-center value  $\theta := \theta - w(v_e)x(v_e)$  for any vertex  $e$  in  $\mathcal{C}(\mathcal{M})$ .
  - 16:     **for**  $e \in \mathcal{C}(\mathcal{M})$  **do**
  - 17:         **if**  $x(e) := \bar{x}(e)$  **then**
  - 18:             Update the set  $\mathcal{C}, \mathcal{C}(\mathcal{M}), \mathcal{T}, \mu_1$  and the edge  $e_{\max}$  (see Part 2 of Case 1).
  - 19:         **end if**
  - 20:         **if**  $x(e) = x^{\mathcal{C}}(e)$  **then**
  - 21:             Update  $\mathcal{M}$  and  $\Delta_{\mathcal{M}}$  (see Part 1 of Case 2).
  - 22:         **end if**
  - 23:         **if**  $x(e) = x^{\mathcal{M}}(e)$  **then**
  - 24:             Update  $\mathcal{M}, \mathcal{C}(\mathcal{M}), \mu_1$ , and  $\Delta_{\mathcal{M}}$  (see Part 2 of Case 2).
  - 25:         **end if**
  - 26:         Update  $\bar{x}(e) := \bar{x}(e) - x(e)$ .
  - 27:     **end for**
  - 28:     **end if**
  - 29:     **if**  $\mu_1 = \theta$  **then**
  - 30:         Solve the equivalent reverse 1-median problem on the current instance (see Berman et. al [5]).
  - 31:         Break the while loop.
  - 32:     **end if**
  - 33: **end while**
  - 34: **Output:** Optimal 1-centdian value and the corresponding modified tree.
- 

Now, we analyze the complexity of the algorithm. We start with considering the number of iterations in the worst case scenario. For Case 1, after an edge is modified by its upper bound, it is not considered in future iterations. Therefore, there are at most  $O(n)$  iterations such that Case 1 occurs. For Part 1 of Case 2, we know that  $v_e$  for  $e \in \mathcal{C}(\mathcal{M})$  can be only shifted to its descendants inside the subtree  $T_e$ . Therefore, there are also at most  $O(n)$

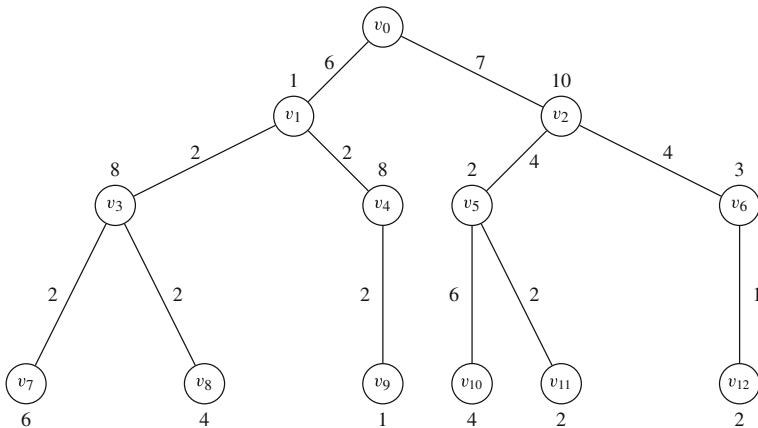


Fig. 7 An instance of the (RCPT) problem with  $\alpha = 2$

iterations such that this part of Case 2 happens. Part 2 of Case 2 happens if there exist one or several edges in  $\mathcal{C} \setminus \mathcal{C}(\mathcal{M})$ . Furthermore, if there is an edge  $e \in \mathcal{C} \setminus \mathcal{C}(\mathcal{M})$  such that  $v_e$  is added into  $\mathcal{M}$  in the current iteration, we later either reduce this edge by Case 1 or Part 1 of Case 2. In other words, we do not revisit the edge  $e$  again under Part 2 of Case 2 from the current iteration onwards. Therefore, the number of iterations such that Part 2 of Case 2 happens at most equals the number of edges in the tree, and thus there are linearly many such iterations. In the initialization step (lines 2–4), we can compute all related quantities in linear time as discussed in Section 3.1. In each iteration, we complete the computation as in Case 1 and Case 2 in linear time as discussed before. If the condition  $\mu_1 = \theta$  (line 29) holds, we apply the algorithm of Berman et al. [5] to solve the current instance in linear time. Therefore, the complexity of the algorithm is  $O(n^2)$ , where  $n$  is the number of vertices in  $T$ .

**Theorem 2** *The reverse centdian problem on trees can be solved in  $O(n^2)$  time, where  $n$  is the number of vertices in the tree.*

As an illustration of Algorithm 1, we provide the following example.

*Example 1* Consider an instance of the reverse 1-centdian problem on a tree  $T$  in Fig. 7 with  $\alpha = 2$ ,  $\bar{x}(v_0, v_1) = \bar{x}(v_0, v_2) = 2$  and  $\bar{x}(e) = 1$  otherwise. Under the given budget  $B = 5$ , Algorithm 1 solves the corresponding problem in 5 iterations (see Table 1).

Table 1 Iterations of the (RCPT) by applying Algorithm 1

Iter.	C	$\mathcal{M}$	$\mathcal{C}(\mathcal{M})$	$e_{\max}$	$W(T_{e_{\max}})$	$\Delta_{\mathcal{M}}$	Reduction	$\mathcal{R}$	B
1	$\{(v_0, v_1), (v_0, v_2)\}$	$\{v_2\}$	$\{(v_0, v_2)\}$	$(v_0, v_1)$	28	43	$x((v_0, v_2)) = \frac{1}{3}$	$\frac{43}{3}$	$\frac{14}{3}$
2	$\{(v_0, v_1), (v_0, v_2)\}$	$\{v_{10}\}$	$\{(v_0, v_2)\}$	$(v_0, v_1)$	28	31	$x((v_0, v_2)) = \frac{2}{3}$	35	4
3	$\{(v_0, v_1), (v_0, v_2)\}$	$\{v_3, v_{10}\}$	$\{(v_0, v_1), (v_0, v_2)\}$	$(v_0, v_1)$	28	30	$x((v_0, v_1)) = \frac{1}{2}$ $x((v_0, v_2)) = 1$	80	$\frac{5}{2}$
4	$\{(v_0, v_1), (v_2, v_5), (v_2, v_6)\}$	$\{v_3, v_{10}\}$	$\{(v_0, v_1), (v_2, v_5)\}$	$(v_0, v_1)$	28	20	$x((v_0, v_1)) = \frac{3}{2}$	122	1
5	$\{(v_1, v_3), (v_1, v_4), (v_2, v_5), (v_5, v_6)\}$	$\{v_{10}\}$	$\{(v_2, v_5)\}$	$(v_1, v_3)$	18	16	$x((v_1, v_3)) = 1$	140	0

Therefore, the maximum improvement of the 1-centdian function within the budget  $B$  is  $\mathcal{R} = 140$ .

## 4 Conclusions

This paper addressed the uniform cost reverse 1-centdian problem on networks, where the 1-centdian function is a convex combination of the 1-median and the 1-center functions. We first proved that the problem is NP-hard by reducing it to the set cover problem. Specifically, we investigated the problem on trees and developed a greedy-like algorithm that solves the problem in quadratic time. To the best of our knowledge, this is the first work concerning the reverse centdian problem. Hence, the reverse centdian problem on other types of networks such as cycles, cactus, etc., is a promising research direction for the future. Another potential direction is to apply prediction models (see [21, 22]) to location theory in order to construct a forecasting approach for the centdian problem on networks in both the classical and the inverse/reverse versions.

**Acknowledgements** This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 101.01-2019.325. The authors would like to acknowledge anonymous referees for their valuable suggestions which helped to improve the paper.

## References

1. Afrashteh, E., Alizadeh, B., Baroughi, F.: Optimal approaches for upgrading selective obnoxious  $p$ -median location problems on tree networks. *Ann. Oper. Res.* **289**, 153–172 (2020)
2. Alizadeh, B., Afrashteh, E.: Budget-constrained inverse median facility location problem on tree networks. *Appl. Math. Comput.* **375**, 125078 (2020)
3. Alizadeh, B., Burkard, R.E.: Combinatorial algorithms for inverse absolute and vertex 1-center location problems on trees. *Networks* **58**, 190–200 (2011)
4. Alizadeh, B., Etemad, R.: The linear time optimal approaches for reverse obnoxious center location problems on networks. *Optimization* **65**, 2025–2036 (2016)
5. Berman, O., Ingco, D.I., Odoni, A.: Improving the location of minisum facilities through network modification. *Ann. Oper. Res.* **40**, 1–16 (1992)
6. Berman, O., Ingco, D.I., Odoni, A.: Improving the location of minimax facilities through network modification. *Networks* **24**, 31–41 (1994)
7. Burkard, R.E., Gassner, E., Hatzl, J.: A linear time algorithm for the reverse 1-median problem on a cycle. *Networks* **48**, 16–23 (2006)
8. Burkard, R.E., Gassner, E., Hatzl, J.: Reverse 2-median problem on trees. *Discrete Appl. Math.* **156**, 1963–1976 (2008)
9. Das, S.K., Roy, S.K., Weber, G.W.: Heuristic approaches for solid transportation- $p$ -facility location problem. *Central. Eur. J. Oper. Res.* **28**, 939–961 (2020)
10. Drezner, Z., Hamacher, H.W. (eds.): *Facility Location: Applications and Theory*. Springer, Berlin (2004)
11. Eiselt, H.A., Marianov, V.: *Foundations of Location Analysis*. International Series in Operations Research and Management Science. Springer, New York (2011)
12. Etemad, R., Alizadeh, B.: Reverse selective obnoxious center location problems on tree graphs. *Math. Methods Oper. Res.* **87**, 431–450 (2018)
13. Gassner, E.: A game-theoretic approach for downgrading the 1-median in the plane with Manhattan metric. *Ann. Oper. Res.* **172**, 393–404 (2009)
14. Gassner, E.: Up- and downgrading the 1-center in a network. *Eur. J. Oper. Res.* **198**, 370–377 (2009)
15. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco (1979)
16. Halpern, J.: Finding minimal center-median combination (cent-dian) of a graph. *Manag. Sci.* **24**, 535–544 (1978)



17. Halpern, J.: The location of a center-median convex combination on an undirected tree. *J. Reg. Sci.* **16**, 237–245 (1976)
18. Kang, L., Zhou, J., Shan, E.: Algorithms for connected  $p$ -centdian problem on block graphs. *J. Comb. Optim.* **36**, 252–263 (2018)
19. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems I: The  $p$ -centers. *SIAM J. Appl. Math.* **37**, 513–538 (1979)
20. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems II: The  $p$ -medians. *SIAM J. Appl. Math.* **37**, 539–560 (1979)
21. Kropat, E., Özmen, A., Weber, G.-W., Meyer-Nieberg, S., Defterli, O.: Fuzzy prediction strategies for gene-environment networks - Fuzzy regression analysis for two-modal regulatory systems. *RAIRO Oper. Res.* **50**, 413–435 (2016)
22. Nalcaci, G., Özmen, A., Weber, G.W.: Long-term load forecasting: models based on MARS, ANN and LR methods. *Central Eur. J. Oper. Res.* **27**, 1033–1049 (2019)
23. Nguyen, K.T.: Inverse 1-median problem on block graphs with variable vertex weights. *J. Optim. Theory Appl.* **168**, 944–957 (2016)
24. Nguyen, K.T.: Reverse 1-center problem on weighted trees. *Optimization* **65**, 253–264 (2016)
25. Nguyen, K.T., Huong, N.T., Hung, N.T.: Combinatorial algorithms for the uniform-cost inverse 1-center problem on weighted trees. *Acta Math. Vietnam.* **44**, 813–831 (2019)
26. Pérez-Brito, D., Moreno-Pérez, J.A., Rodríguez-Martín, I.: The 2-facility centdian network problem. *Locat. Sci.* **6**, 369–381 (1998)
27. Pham, V.H., Nguyen, K.T., Le, T.T.: Inverse stable point problem on trees under an extension of Chebyshev norm and Bottleneck Hamming distance. *Optim. Method Softw.* <https://doi.org/10.1080/10556788.2020.1713778> (2020)
28. Sepasian, A.R.: Upgrading the 1-center problem with edge length variables on a tree. *Discrete Optim.* **29**, 1–17 (2018)
29. Tamir, A., Pérez-Brito, D., Moreno-Pérez, J.A.: A polynomial algorithm for the  $p$ -centdian problem on a tree. *Networks* **32**, 255–262 (1998)
30. Zhang, J., Yang, X., Cai, M.-C.: Reverse center location problem. In: Aggarwal, A., Rangan, C.P. (eds.) *Algorithms and Computation. Lecture Notes in Computer Science*, vol. 1741, pp. 279–294. Springer, Berlin (1999)
31. Zhang, J., Liu, Z., Ma, Z.: Some reverse location problems. *Eur. J. Oper. Res.* **124**, 77–88 (2000)

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