Forschungsschwerpunkt

Algorithmen und mathematische Modellierung

Reverse 2-Median Problem on Trees

Rainer Ernst Burkard, Elisabeth Gassner, and Johannes Hatzl

Project Area(s):

Effizient lösbare kombinatorische Optimierungsprobleme

Institut für Optimierung und Diskrete Mathematik (Math B)

Report 2006-15, August 2006

Reverse 2-Median Problem on Trees

Rainer E. Burkard   Elisabeth Gassner   Johannes Hatzl [*]

Abstract

This paper concerns the reverse 2-median problem on trees and the reverse 1-median problems on cacti that contain one cycle. It is shown that both models under investigation can be transformed to an equivalent reverse 2-median problem on a path. For this new problem an $\mathcal{O}(n \log n)$ algorithm is proposed, where n is the number of vertices of the path. It is also shown that there exists an integral solution if the input data are integral.

Keywords: Reverse optimization, facility location, median problems, combinatorial optimization.

1 Introduction

Location problems are due to their relevance in practice of special theoretical interest. Classical location problems deal with finding optimal locations for facilities. However, in practice the facilities may already exist and instead of finding optimal location the task is to improve the given locations by changing some parameters (e. g., traffic connections) within a given budget constraint. This kind of improvement problem is called reverse problem.

This paper deals with reverse p-median problems on special graphs, namely the reverse 2-median problem on trees and the reverse 1-median problem on cacti with one cycle. Both problems are edge improvement problems, i. e., the task is to find an optimal reduction strategy of the edge lengths.

Reverse median and reverse center problems have already been subject of several investigations: The reverse 1-median problem as well as the reverse 1-center problem are known to be $\mathcal{NP}$-hard ([3], [7]). Therefore, special networks have been studied. Berman, Ingco and Odoni investigated the reverse 1-median problem on a tree [2] and Burkard, Gassner and Hatzl [7] developed a linear time algorithm for the reverse 1-median problem on a

[*] {burkard,gassner,hatzl}@opt.math.tu-graz.ac.at. Institute of Optimization, Dynamical Systems and Discrete Mathematics, Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria.

cycle. Furthermore, Berman, Ingco and Odoni [3] and Zhang, Liu and Ma [14] dealt with the reverse 1-center problem and developed polynomial time algorithms.

The term reverse location problem is often used for a similar class of improvement problems, where the task is to modify parameters at minimum cost such that the quality of the prespecified locations is within given bounds. Zhang, Yang and Cai [15] considered the problem of how to reduce the lengths of the edges in a network so that the distances from a given vertex to all other vertices satisfy a given upper bound. The authors prove inapproximability results and suggest an $\mathcal{O}(n \log n)$ time algorithm for the special case of a tree. Inapproximability results for the corresponding problems where the total cost are measured in ℓ_1- or ℓ_2-norm are discussed in Zhang, Yang and Cai [16]. Furthermore, an $\mathcal{O}(n \log n)$ time algorithm for the case of ℓ_∞-norm is developed in [16].

Reverse problems are strongly related to inverse problems. The goal of an inverse location problem is to modify parameters at minimum cost such that prespecified locations become optimal. Burkard, Pleschiutschnig and Zhang [6] proved that inverse p-median problems are solvable in polynomial time and presented fast algorithms for the inverse 1-median on a tree and in the plane with ℓ_1- and ℓ_∞-norm. Cai, Yang and Zhang [9] proved by a reduction from the satisfiability problem that the inverse center problem is $\mathcal{NP}$-hard.

Network improvement problems have also been applied to several other classical combinatorial optimization problems, e. g., shortest paths (Burton, Pulleyblank and Toint [8], Fulkerson and Harding [12], Zhang, Liu and Ma [14]) minimum spanning trees (Frederickson and Solis-Oba [11], Dragmeister et al. [10], Krumke et al.[13]) or bottleneck 0/1-combinatorial optimization problems (Burkard, Klinz and Zhang [4] and Burkard, Lin and Zhang [5]).

This paper is organized as follows: In Section 2 the problem under consideration is introduced. It is proved that the reverse 2-median problem on trees as well as the reverse 1-median problem on cacti with one cycle can be transformed to the reverse 2-median problem on a path. In Section 3 several properties of an optimal solution of the reverse 2-median problem on a path are proved. Furthermore, a solution method which leads to an $\mathcal{O}(n^2)$ time algorithm for linear cost functions is suggested. Based on the structural investigations of Section 3 an $\mathcal{O}(n \log n)$ time algorithm for the reverse 2-median problem on a path with uniform cost functions is developed in Section 4.

2 Problem Formulation

An instance of the reverse 2-median problem is given by a graph $G = (V, E)$ with edge lengths $l_e \in \mathbb{R}_+$ for $e \in E$ and vertex weights $w_v \in \mathbb{R}_+$ for $v \in V$. Furthermore, a budget $B > 0$ and two prespecified vertices, say v' and v'', representing the locations of two facilities are known. The task is to use the budget in order to change the length of some edges such that the overall sum of the weighted distance of the vertices to the respective closest facility becomes as small as possible. Thereby, the distance $d^l(v_i, v_j)$ in G of vertex v_i to vertex v_j is the length of a shortest (v_i, v_j) path in G corresponding to the edge lengths l_e. In order to improve the locations of v' and v'', we are allowed to reduce the edge lengths l_e. The cost for reducing the edge e by x_e units are given by some function $f_e(x_e) > 0$. Thus, the decision variables are x_e, describing the reduction of the edge lengths $\bar{l}_e = l_e - x_e$. Additionally, upper bounds $u_e \leq l_e$ for all $e \in E$ on the maximum allowable edge reduction are taken into account. In the following, we will always denote the distance from a vertex v to facility v' by $d_1^l(v)$, whereas the distance to facility v'' is denoted by $d_2^l(v)$. Using the notation introduced above, the problem can formally be stated as follows:

$$\begin{aligned}
\min \quad & \sum_{v \in V} \min_{i=1,2} w_v \, d_i^{\bar{l}}(v) && && (1)\\
\text{s.t.} \quad & \bar{l}_e = l_e - x_e && \forall e \in E &&\\
& \sum_{e \in E} f_e(x_e) \leq B && && (2)\\
& 0 \leq x_e \leq u_e && \forall e \in E. &&
\end{aligned}$$

In [7] it has already been shown that the reverse 1-median problem, i.e., the reverse 2-median problem where $v' = v''$, is strongly $\mathcal{NP}$-hard even for bipartite graphs and the unit cost model, i.e.,

$$f_e(x_e) = Cx_e \tag{3}$$

for all $e \in E$ and some positive constant C. Furthermore, there does not exist any polynomial time algorithm with constant approximation ratio (unless $\mathcal{NP} = \mathcal{P}$). Obviously, the problem remains $\mathcal{NP}$-hard for the case where $v' \neq v''$, because if we consider an instance of the reverse 1-median problem with a facility given at vertex v' we only have to add a new vertex v, the edge $e = (v, v')$ and set $v = v''$. It is easy to see that it does not make any sense to spend any budget in the artificial edge e and thus, the modified problem can only be solved in polynomial time if and only if the original reverse 1-median problem can be solved in polynomial time.

Thus, we restrict our investigations to the reverse 2-median problem on trees and the reverse 1-median problem on cacti with one cycle. In this paper, we will mainly deal with the unit cost model. Without loss of generality, we may assume that $C = 1$ because otherwise equation (2) can be rewritten as

$$\sum_{e \in E} x_e \leq \frac{B}{C} =: \tilde{B}$$

and we can solve the problem with $\tilde{B}$ instead of B.
From now on, we will always assume that in the reverse 2-median problems $v' \neq v''$ and that the prespecified vertex in the reverse 1-median problems is denoted by v'.

2.1 Reverse 2-Median Problem on Trees

In this subsection a transformation from the reverse 2-median problem on a tree to the reverse 2-median problem on a path is presented.

Let $T = (V, E)$ be a tree with $V = \{v_1, \dots, v_n\}$ and let two prespecified vertices v' and v'' be given. Then there exists a unique path $P[v', v'']$ in T that links v' and v''. If we denote the vertex set of this path by $V(P[v', v'']) = \{v' = v_1, v_2, \dots, v_{p-1}, v_p = v''\}$ the vertex set of the tree can be partitioned into the sets $V(P[v', v''])$ and $V^c(P[v', v'']) := V \setminus V(P[v', v''])$. Similarly, the edge set E of the tree can be written as $E = E(P[v', v'']) \uplus E^c(P[v', v''])$, where $E(P[v', v'']) = \{e_1, \dots, e_{p-1}\}$ is the edge set of the path $P[v', v'']$, $e_i = (v_i, v_{i+1})$ and $E^c(P[v', v'']) := E \setminus E(P[v', v''])$. For the sake of simplicity, we write $e \in P[v', v'']$ instead of $e \in E(P[v', v''])$ and $v \in P[v', v'']$ instead of $v \in V(P[v', v''])$ if no confusion is possible. If all edges $e \in P[v', v'']$ are deleted from the tree T we obtain p disjoint trees each of them containing exactly one vertex of $V(P[v', v''])$. Consider these trees $T_{v_k} = (V_k, E_k)$ for $k = 1, \dots, p$ rooted at the vertices v_k of the path $P[v', v'']$. Furthermore, we will write $r(v) = v_k$ if vertex $v \in V$ is in the tree T_{v_k}.
Using these trees, a new weight for each vertex $v_k \in P[v', v'']$ is introduced by

$$\tilde{w}_{v_k} := \sum_{v \in V_k} w_v. \tag{4}$$

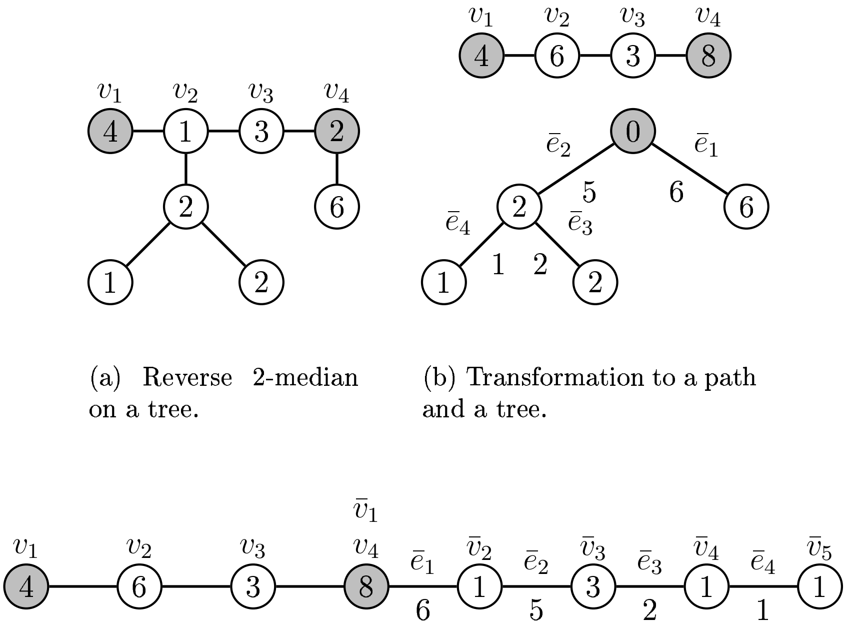
Notice that the path from $v \in V$ to v' is the disjoint union of $P[v, r(v)]$ and $P[r(v), v']$. An analogue result holds for v''. Hence, we get

$$d_i^{\bar{l}}(v) = d^{\bar{l}}(v, r(v)) + d_i^{\bar{l}}(r(v))$$

for $i = 1, 2$ and $v \in V$. Using these notations, (1) can be rewritten as

$$\begin{aligned}
&\sum_{v \in V} \Big(w_v d^{\bar{l}}(v, r(v)) \Big) + \sum_{v \in V} \Big(\min_{i=1,2} w_v \, d_i^{\bar{l}}(r(v)) \Big) = \\
&\sum_{v \in V} \Big(w_v d^{\bar{l}}(v, r(v)) \Big) + \sum_{v_k \in P[v', v'']} \Big(\min_{i=1,2} d_i^{\bar{l}}(v_k) \sum_{v \in V_k} w_v \Big) = \\
&\sum_{v \in V} \Big(w_v d^{\bar{l}}(v, r(v)) \Big) + \sum_{v_k \in P[v', v'']} \Big(\min_{i=1,2} \tilde{w}_{v_k} \, d_i^{\bar{l}}(v_k) \Big).
\end{aligned} \tag{5}$$

Now the two terms of the objective function given in (5) are investigated separately. The first term corresponds to the objective function of a reverse 1-median problem on the tree which can be obtained by identifying $v_1, v_2, \dots, v_p$ with each other. The identified vertices correspond to the prespecified location of one facility. The second term is equivalent to the objective function of a reverse 2-median problem on the path $P[v', v'']$ with facilities v' and v'' and the new vertex weights $\tilde{w}_v$ (see Figure 1(b)).



v_1 v_2 v_3 v_4
4 1 3 2
2 6
1 2
v_1 v_2 v_3 v_4
4 6 3 8
$\bar{e}_2$ 0 $\bar{e}_1$
5 6
$\bar{e}_4$ 2 $\bar{e}_3$ 6
1 1 2 2
$\bar{v}_1$
v_1 v_2 v_3 v_4 $\bar{e}_1$ $\bar{v}_2$ $\bar{e}_2$ $\bar{v}_3$ $\bar{e}_3$ $\bar{v}_4$ $\bar{e}_4$ $\bar{v}_5$
4 6 3 8 1 3 1 1
6 5 2 1

(a) Reverse 2-median on a tree.

(b) Transformation to a path and a tree.

(c) Transformation to one path.

Figure 1: The transformation from the reverse 2-median problem on a tree to a reverse 2-median problem on a path. The numbers within the vertices are the vertex weights, the numbers on the edges are the edge weights W_e and the prespecified locations are marked grey.)

The next step is to transform the reverse 1-median problem on a tree into a reverse 1-median problem on a path. Note that

$$d^{\bar{l}}(v, r(v)) = \sum_{e \in P[v,r(v)]} (l_e - x_e) \tag{6}$$

holds. For every $e \in E^c(P[v', v'']) = \biguplus_{k=1}^{p} E_k$ we define its edge weight as the sum of weights travelling along e:

$$W_e := \sum_{\{v \in V : e \in P[v,r(v)]\}} w_v. \tag{7}$$

Using (6) and (7), we get

$$\sum_{v \in V} \left(w_v d^{\bar{l}}(v, r(v)) \right) = \sum_{v \in V} w_v \sum_{e \in E(P[v,r(v)])} (l_e - x_e) = \sum_{e \in E^c(P[v',v''])} (l_e - x_e) W_e. \tag{8}$$

However, this modified part of the objective function is now equivalent to the objective function arising from a reverse 1-median problem on a path. We only have to sort the edges of $E^c(P[v', v''])$ in decreasing order corresponding to their coefficients W_e, i.e., $E^c(P[v', v'']) = \{\bar{e}_1, \ldots, \bar{e}_q\}$, where $W_{\bar{e}_i} \geq W_{\bar{e}_{i+1}}$. Using this sorting, the path $\bar{P} = (V(\bar{P}), E(\bar{P}))$ with $V(\bar{P}) = \{\bar{v}_1, \ldots, \bar{v}_{q+1}\}$, $E(\bar{P}) = \{\bar{e}_1, \ldots, \bar{e}_q\}$ and $\bar{e}_i = (\bar{v}_i, \bar{v}_{i+1})$ can be constructed. In order to state the reverse 1-median problem on $\bar{P}$, we set $v' = \bar{v}_1$ and define the weights by

$$w_{\bar{v}_i} = \begin{cases} W_{\bar{e}_{i-1}} - W_{\bar{e}_i} & \text{for } i = 1, \ldots, q; \\ W_{\bar{e}_q} & \text{for } i = q+1. \end{cases} \tag{9}$$

Observe that the objective function of the reverse 1-median problem on path $\bar{P}$ with vertex weights $w_{\bar{v}_i}$ for $i = 1, \ldots, q+1$ is of the form

$$\begin{aligned} \sum_{i=1}^{q+1} w_{\bar{v}_i} d^{\bar{l}}(\bar{v}_i, \bar{v}_1) &= \sum_{i=1}^{q+1} \left(w_{\bar{v}_i} \sum_{j=1}^{i-1} (l_{\bar{e}_j} - x_{\bar{e}_j}) \right) \\ &= \sum_{j=1}^{q} (l_{\bar{e}_j} - x_{\bar{e}_j}) \sum_{i=j+1}^{q+1} w_{\bar{v}_i} \\ &= \sum_{j=1}^{q} (l_{\bar{e}_j} - x_{\bar{e}_j}) W_{\bar{e}_j} \end{aligned} \tag{10}$$

Hence, the objective function (8) of the reverse 1-median problem on a tree is equivalent to the objective function (10) for the reverse 1-median problem on path $\bar{P}$. Thus, we have constructed a reverse 3-median problem on a disconnected graph with two components $P[v', v'']$ and $\bar{P}$, where each component is a path. However, these two components can be combined by identifying v_p with $\bar{v}_1$. Hence, a reverse 2-median problem on a path with vertex weights given in (4) and (9) is obtained, which is equivalent to the original reverse 2-median problem on a tree (see Figure 1(c)). This transformation can be done in $\mathcal{O}(n \log n)$ time since we have to sort the edges in $E^c(P[v', v''])$ according to W_e.

The reverse 2-median problem on a path with the unit cost model, R2MP for short, can be stated in the following form:

Let $P = (V, E)$ be a path, i.e., $|V| = |E| + 1 = n$ and $V = \{1, 2, \ldots, p-1, p, p+1, \ldots, n\}$ and $E = \{e_1, \ldots, e_{n-1}\}$ where $e_i = (i, i+1)$ for $i = 1, \ldots, n-1$. Furthermore, for each edge e_i there is given the length l_i and a bound $u_i \leq l_i$ on the edge reduction. In addition, vertex weights $w_i \geq 0$ for all $i \in V$, two prespecified vertices $v' = 1$ and $v'' = p$ and a budget $B > 0$ are given. R2MP is then defined in the following way:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{p} w_j \min_{i=1,2} d_i^{\bar{l}}(j) + \sum_{j=p+1}^{n} \left(w_j \sum_{i=p}^{j-1} \bar{l}_i \right) && && (11)\\
\text{s.t.} \quad & \bar{l}_i = l_i - x_i && i = 1, \dots, n-1 && \\
& 0 \le x_i \le u_i && i = 1, \dots, n-1 && (12)\\
& \sum_{i=1}^{n-1} x_i \le B. && && (13)
\end{aligned}$$

Note that in (11)

$$\min_{i=1,2} d_i^{\bar{l}}(j) = \min \left(\sum_{i=1}^{j-1} \bar{l}_i, \sum_{i=j}^{p-1} \bar{l}_i \right)$$

holds for $j = 1, \dots, p$ and

$$\sum_{i=p}^{j-1} \bar{l}_i = d_2^{\bar{l}}(j) \le d_1^{\bar{l}}(j) \qquad j = p+1, \dots, n$$

hold. In order to exclude trivial cases, we assume throughout this paper that

$$\sum_{i=1}^{n-1} u_i > B \tag{14}$$

holds. Otherwise $x_i^* = u_i$ for $i = 1, \dots, n-1$ would be an optimal solution. Using assumption (14), it is clear that there always exists an optimal solution such that the whole budget is used. Thus, constraint (13) can be replaced by

$$\sum_{i=1}^{n-1} x_i = B. \tag{15}$$

Furthermore, we assume without loss of generality that $w_j > 0$ holds for all $j \in V$. If $w_n = 0$ vertex n and edge e_{n-1} can be deleted. If $w_j = 0$ for $j = 1, \dots, n-1$ then we identify vertex j with $j+1$ and set $u_{j-1} = u_{j-1} + u_j$ and $l_{j-1} = l_{j-1} + l_j$. This procedure reduces the number of zero-weight vertices by one and is therefore repeated until no zero-weight vertex exists.

2.2 Reverse 1-Median Problem on Cacti with one cycle

In this subsection another reverse location problem is discussed, namely the reverse 1-median problem on cacti with one cycle. At the beginning we need some definitions in order to introduce the discussed graph class. Let $G = (V, E)$ be a connected graph. A vertex $v \in V$ is called a cut vertex if the graph after removing v and all edges incident to v is disconnected. A connected graph without a cut vertex is called a nonseparable graph, e.g., a cycle is nonseparable. A block of a graph is a maximal nonseparable subgraph. A block that is a cycle is called a cyclic block. G is called cactus if every block with three or more vertices is a cyclic block. Note that a cactus without a cyclic block is a tree. In this paper, the reverse 1-median problem on cacti that have exactly one block $C = (V(C), E(C))$ that is cyclic is discussed. This means that the vertex set of the given graph $G = (V, E)$ can be partitioned into the set of vertices of the cycle denoted by $V(C) = \{v_1, \dots, v_p\}$ and the set $V^c(C) := V \setminus V(C)$. Similarly, the edge set E of the considered graph can be written as $E = E(C) \uplus E^c(C)$, where $E(C) = \{e_1, \dots, e_p\}$ and $e_i = (v_i, v_{i+1})$ for $i = 1, \dots, p-1$ and $e_p = (p, 1)$.

Note that if all the edges $e \in E(C)$ are removed from the original graph p disjoint trees are obtained where each of them contains exactly one vertex of $V(C)$. These trees are denoted by $T_{v_k} = (V_k, E_k)$ for $k = 1, \dots, p$ and are rooted at the vertices v_k of the cycle C. Without loss of generality it is assumed that $v' \in T_{v_1}$. As in the previous subsection, we write $r(v) = v_k$ if vertex $v \in V$ is in the tree T_{v_k}. Hence, $v_1 = r(v')$.
Now a new weight function for the vertices $v_k \in V(C)$ can be defined by

$$\tilde{w}_{v_k} := \begin{cases} \sum_{i=2}^{p} \sum_{v \in V_i} w_v + w_{r(v')} & \text{for } k = 1; \\ \sum_{v \in V_k} w_v & \text{for } k = 2, \dots, p. \end{cases}$$

Let $v \in V_k$ for $k \neq 1$. On the path from v to v' vertex v travels along edges $e \in E_k \subseteq E^c(C)$, followed by some edges $e \in E(C)$ and finally ends with some edges $e \in E_1 \subseteq E^c(C)$. Thus,

$$d^{\bar{l}}(v, v') = d^{\bar{l}}(v, r(v)) + d^{\bar{l}}(r(v), r(v')) + d^{\bar{l}}(r(v'), v')$$

for all $v \in \bigcup_{j=2}^{p} V_j$, where the following equation holds for $r(v) = v_k$

$$d^{\bar{l}}(v_k, r(v')) = d^{\bar{l}}(v_k, v_1) = \min \left(\sum_{i=1}^{k-1} \bar{l}_{e_i}, \sum_{i=k}^{p} \bar{l}_{e_i} \right).$$

Hence, the corresponding objective function can be written as

$$\begin{aligned} &\sum_{v \in V} w_v d^{\bar{l}}(v, v') \\ &= \sum_{v \in V \setminus V_1} w_v \left(d^{\bar{l}}(r(v), r(v')) + d^{\bar{l}}(v, r(v)) + d^{\bar{l}}(r(v'), v') \right) + \sum_{v \in V_1} w_v d^{\bar{l}}(v, v') \\ &= \sum_{v_k \in V(C)} \tilde{w}_{v_k} d^{\bar{l}}(v_k, r(v')) + \sum_{v \in V \setminus V_1} w_v d^{\bar{l}}(v, r(v)) + \sum_{v \neq v_1 \in V_1} w_v d^{\bar{l}}(v, v') + \tilde{w}_{v_1} d^{\bar{l}}(v_1, v'). \qquad (16) \end{aligned}$$

Observe, that the first term in (16) corresponds to the traffic on the cycle C, the second term corresponds to the traffic on the trees $T_{v_2}, \dots, T_{v_p}$, the third term describes the cost of all vertices $v \in V_1 \backslash \{v_1\}$ travelling to v' and finally the fourth term describes the cost for transporting the whole weight that travels through v_1 to v'. The terms of the objective function (16) are now analyzed separately. The first term is the objective function of the reverse 1-median problem on the cycle C with vertex weights $\tilde{w}_{v_k}$ and prespecified vertex $r(v') = v_1$. The remaining terms are equivalent to a reverse 1-median problem on a tree which is obtained by identifying the vertices $v', v_2, \dots, v_p$ with each other and using vertex weights w_v for all $v \neq v_1$ and $\tilde{w}_{v_1}$. The identified vertices are the prespecified vertex. Thus, the reverse 1-median problem on a cactus with one cycle is now equivalent to a reverse 2-median problem on a disconnected graph with two components, where one component is a tree and the other one is a cycle each of them having one prespecified vertex (see Figure 2(b)).

Now it is shown that this modified instance can again be transformed to a reverse 2-median problem on a path. In the previous section we have already presented the transformation from the reverse 1-median problem on a tree to a reverse 1-median problem on a path. In addition, the reverse 1-median problem on the cycle $C = (V(C), E(C))$ with vertex weights $\tilde{w}_k$ for all $v_k \in V(C)$ and prespecified vertex v_1 is equivalent to the following reverse 2-median problem on a path P. Let $P = (V(P), E(P))$, where $V(P) := V(C) \cup v_{p+1}$ and $E(P) = \{e_1, \dots, e_p\}$ with $e_i = (v_i, v_{i+1})$ for $i = 1, \dots, p$, vertex weights $\tilde{w}$ and $\tilde{w}_{v_{p+1}} = 0$. Furthermore, let $v' = v_1$ and $v'' = v_{p+1}$ be the new prespecified vertices. Note that for this problem

$$d_1^{\bar{l}}(v_j) = \sum_{i=1}^{j-1} \bar{l}_{e_j}$$

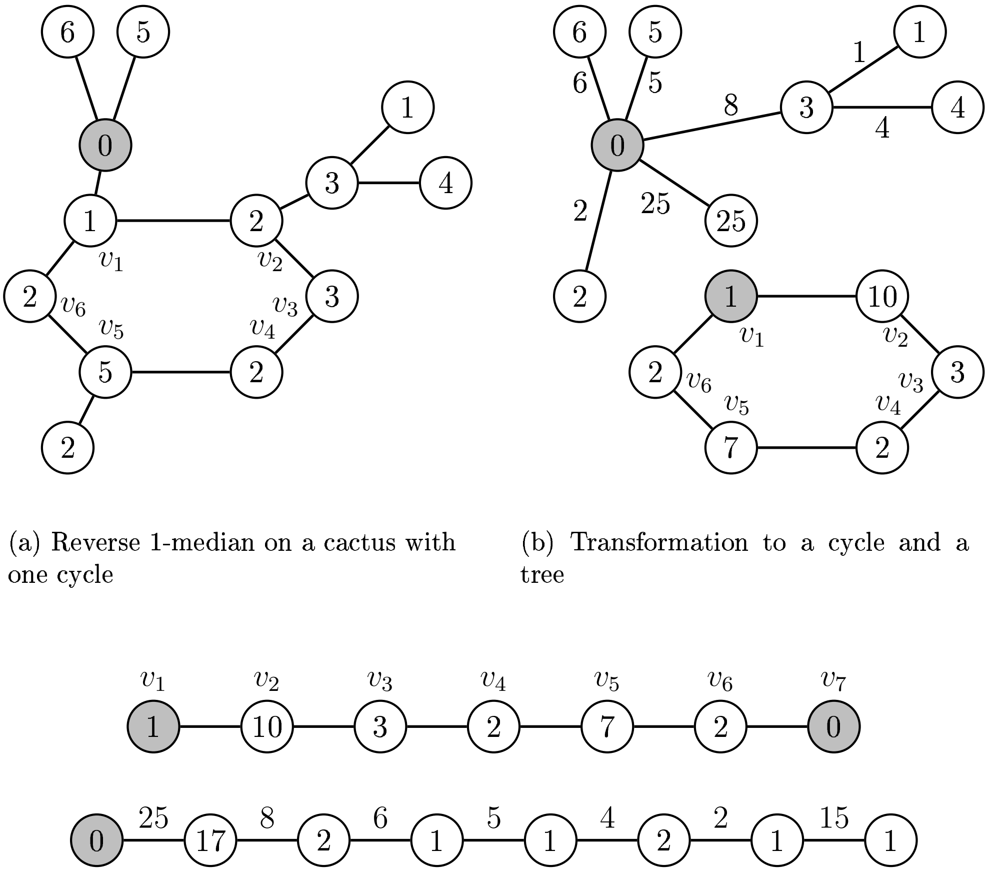
and

$$d_2^{\bar{l}}(v_j) = \sum_{i=j}^{p} \bar{l}_{e_j}$$

hold. Thus, the objective function for this reverse 2-median problem on the path is given by

$$\begin{aligned}
&\sum_{v_k \in V(C)} \min_{i=1,2} \tilde{w}_{v_k}\, d_i^{\bar{l}}(v_k) = \\
&\sum_{v_k \in V(C)} \tilde{w}_{v_k} \min\left(\sum_{i=1}^{k-1} \bar{l}_{e_j}, \sum_{i=k}^{p} \bar{l}_{e_j} \right) = \\
&\sum_{v_k \in V(C)} \tilde{w}_{v_k} d^{\bar{l}}(v_k, r(v')).
\end{aligned}$$

This is exactly the objective function of the reverse 1-median problem of the cycle. Thus the original problem is again equivalent to a reverse 3-median problem on a disconnected graph with two paths as components (see Figure 2(c)). As it was shown in the previous section this problem can be transformed to the reverse 2-median problem on a path given in (11) – (12) and (15).



(a) Reverse 1-median on a cactus with one cycle

(b) Transformation to a cycle and a tree

(c) Transformation to two paths which are then connected to one path

Figure 2: The transformation from the reverse 1-median problem on a cactus with one cycle to a reverse 2-median problem on a path.

3 Structural investigations

Before an efficient algorithm of the R2MP is discussed, some properties of an optimal solution are proved. The following definition is crucial for the ideas behind the proofs and will finally lead to an $\mathcal{O}(n \log n)$ time algorithm.

Definition 3.1. An edge $e = (k, k+1)$ for $k = 1, \dots, p-1$ is called critical with respect to modified edge lengths $\bar{l} = (\bar{l}_1, \dots, \bar{l}_{n-1})$, if

$$\min_{i=1,2} d_i^{\bar{l}}(k) = d_1^{\bar{l}}(k)$$

and

$$\min_{i=1,2} d_i^{\bar{l}}(k+1) = d_2^{\bar{l}}(k+1).$$

If $e = (k, k+1)$ is critical with respect to $\bar{l}$, the index k is called a critical index with respect to $\bar{l}$.

This definition immediately leads to the following observation.

Observation 3.2. *Let $x = (x_1, \dots, x_{n-1})$ be a feasible solution and k a critical index with respect to $\bar{l} = (l_1 - x_1, \dots, l_{n-1} - x_{n-1})$. Then the inequalities*

$$\sum_{i=1}^{j-1} \bar{l}_i = d_1^{\bar{l}}(j) \leq d_2^{\bar{l}}(j) \qquad \text{for } j = 1, \dots, k,$$

and

$$\sum_{i=j}^{p-1} \bar{l}_i = d_2^{\bar{l}}(j) \leq d_1^{\bar{l}}(j) \qquad \text{for } j = k+1, \dots, p,$$

hold.

Using Observation 3.2, the objective value of a feasible solution $x = (x_1, \dots, x_{n-1})$ can be obtained by

$$\begin{aligned}
\sum_{j=1}^{p} w_j \min_{i=1,2} d_i^{\bar{l}}(j) + \sum_{j=p+1}^{n} \left(w_j \sum_{i=p}^{j-1} \bar{l}_i \right) &= \\
\sum_{j=1}^{k} \left(w_j \sum_{i=1}^{j-1} \bar{l}_i \right) + \sum_{j=k+1}^{p} \left(w_j \sum_{i=j}^{p-1} \bar{l}_i \right) + \sum_{j=p+1}^{n} \left(w_j \sum_{i=p}^{j-1} \bar{l}_i \right) &= \\
\sum_{i=1}^{p-1} \bar{l}_i W_i^k + \sum_{i=p}^{n-1} \bar{l}_i W_i^k &= \\
\sum_{i=1}^{p-1} (l_i - x_i) W_i^k + \sum_{i=p}^{n-1} (l_i - x_i) W_i^k &
\end{aligned}$$

where

$$W_i^k = \begin{cases} \sum_{j=i+1}^{k} w_j & i = 1, \dots, k-1, \\ 0 & i = k, \\ \sum_{j=k+1}^{i} w_j & i = k+1, \dots, p-1, \\ \sum_{j=p+1}^{n} w_j & i = p, \dots, n-1. \end{cases}$$

and $e = (k, k+1)$ is a critical edge with respect to $\bar{l} = (l_1 - x_1, \ldots, l_{n-1} - x_{n-1})$. Note that the following relations are true:

$$\begin{aligned}
W_i^{k+1} &= W_i^k + w_{k+1} && \text{for } i = 1, \ldots, k, \\
W_i^{k+1} &= W_i^k - w_{k+1} && \text{for } i = k+1, \ldots, p-1, \\
W_i^{k+1} &= W_i^k && \text{for } i = p, \ldots, n-1.
\end{aligned}$$

Let $x^* = (x_1^*, \ldots, x_{n-1}^*)$ be an optimal solution and k a critical index with respect to $\bar{l} = (l_1 - x_1^*, \ldots, l_{n-1} - x_{n-1}^*)$. Then x^* is an optimal solution of the following linear program $LP(k)$:

$$\begin{array}{lll}
\min & \displaystyle\sum_{i=1}^{n-1} (l_i - x_i) W_i^k & \\
\text{s.t.} & 0 \leq x_i \leq u_i & i = 1, \ldots, n-1 \\
& \displaystyle\sum_{i=1}^{n-1} x_i = B. &
\end{array}$$

The assumption that $w_j > 0$ holds for all $j = 1, \ldots, n$ implies

$$W_1^k > W_2^k > \ldots > W_{k-1}^k > W_k^k = 0 < W_{k+1}^k < \ldots < W_{p-1}^k \tag{17}$$

and

$$W_p^k > \ldots > W_{n-2}^k > W_{n-1}^k. \tag{18}$$

It is easy to see that $LP(k)$ can be solved for any fixed $k = 1, \ldots, p-1$ in linear time because of the monotonicity properties (17) and (18). This observation immediately leads to an $\mathcal{O}(n^2)$ algorithm by solving $LP(k)$ for any $k \in \{1, \ldots, p-1\}$. However, using information from an optimal solution of $LP(k-1)$ for problem $LP(k)$ yields a faster algorithm which is developed in the next section.

The time complexity of $\mathcal{O}(n^2)$ can even be achieved for a more general model, namely using cost functions $f_i(x_i) = C_i x_i$ where $C_i \in \mathbb{R}_+$ for all $e_i \in E = \{e_1, \ldots, e_{n-1}\}$. In this model the budget constraint (15) has to be replaced by

$$\sum_{i=1}^{n-1} C_i x_i = B.$$

The modified $LP(k)$ is a continuous knapsack problem and can therefore be solved in linear time (see Balas and Zemel [1]).

At the end of this section some observations are stated which can be proved in a straightforward way using (17), (18) and the special form of $LP(k)$.

Lemma 3.3. *Let $x = (x_1, \dots, x_{n-1})$ be a feasible solution of $LP(k)$. Then the following properties are equivalent:*

1. *x is an optimal solution*
2. *If $x_j > 0$ then $x_i = u_i$ for all i for which $W_i^k > W_j^k$*

Let $x = (x_1, \dots, x_{n-1})$ be a feasible solution of $LP(k)$. Then an edge e_i is called fully reduced if $x_i = u_i$ and is called partially reduced if $0 < x_i < u_i$.

Theorem 3.4. *There exists an optimal solution $x^* = (x_1^*, \dots, x_{n-1}^*)$ of the reverse 2-median problem on a path such that $0 < x_i^* < u_i$ holds for at most one index i, i. e., there exists at most one partially reduced edge.*

Theorem 3.5. *If B and u_i for $i = 1, \dots, n-1$ are integer values then there exists an integral optimal solution of the reverse 2-median problem on a path.*

4 An $\mathcal{O}(n \log n)$ Algorithm

In this section an $\mathcal{O}(n \log n)$ time algorithm for the reverse 2-median problem on a path defined by (11) – (12) and (15) is developed. The main idea of the algorithm is to solve $LP(k)$ for $k = 1, \dots, p-1$. Instead of solving each $LP(k)$ from the scratch we use some information from the already solved problem $LP(k-1)$. In the following we call the procedure to solve $LP(k)$ an iteration. Hence, there are $p - 1 = \mathcal{O}(n)$ iterations in total.

In order to describe the computations needed in iteration k the following notation is introduced:

$$\begin{aligned} L^k &:= \{e_1, \dots, e_{k-1}\}; \\ R^k &:= \{e_k, \dots, e_{p-1}\}; \\ \bar{P} &:= \{e_p, \dots, n\}. \end{aligned}$$

In order to simplify the notation, we write W_i instead of W_i^k for $e_i \in \bar{P}$ since these edge weights are independent of the iteration k.

The next task is to describe how an optimal solution of $LP(k)$ given an optimal solution of $LP(k-1)$ can be obtained. Recall that a feasible solution x of $LP(k)$ is optimal if and only if the following property holds: If $x_j > 0$ then $x_i = u_i$ for all edges i with $W_i^k > W_j^k$. The idea of the following procedure is to start with an optimal solution x^{k-1} of $LP(k-1)$ and to shift the investment successively from an edge i to an edge j with $W_i^k < W_j^k$.

Now the shift procedure described in Algorithm 1 is explained in more detail.

Algorithm 1 Shift Procedure: Solves $LP(k)$ given an optimal solution of $LP(k-1)$

```
optimal = false
while optimal = false do
    Determine $e^+ \in E$ with $W^k_{e^+} = \max\{W^k_e : x_e < u_e\}$ {$e^+$ is called valuable shift edge}
    Determine $e^- \in E$ with $W^k_{e^-} = \min\{W^k_e : x_e > 0\}$ {$e^-$ is called trashy shift edge}
    if $W^k_{e^+} > W^k_{e^-}$ then {$x$ is not optimal}
        $\delta = \min\{u_{e^+} - x_{e^+}, x_{e^-}\}$
        $x_{e^+} = x_{e^+} + \delta$, $x_{e^-} = x_{e^-} - \delta$
        $z(x) = z(x) - \delta(W^k_{e^+} - W^k_{e^-})$
    else
        optimal = true
    end if
end while
```

Definition 4.1. Let x be a feasible solution of $LP(k)$.

- An edge denoted by $e^+(L^k) \in L^k$ is called to be *valuable in* L^k if

  $$W^k_{e^+(L^k)} = \max\{W^k_e : e \in L^k, x_e < u_e\}.$$

  In an analogue way we define an edge $e^+(R^k) \in R^k$ to be *valuable in* R^k and an edge $e^+(\bar{P}) \in \bar{P}$ to be *valuable in* $\bar{P}$. Among all valuable edges that one with maximum weight is called valuable shift edge.

- An edge denoted by $e^-(L^k) \in L^k$ is called to be *trashy in* L^k if

  $$W^k_{e^-(L^k)} = \min\{W^k_e : e \in L^k, x_e > 0\}.$$

  A trashy edge $e^-(R^k)$ in R^k and a trashy edge $e^-(\bar{P})$ in $\bar{P}$ is defined analogously. Among all trashy edges that one with minimum weight is called trashy shift edge.

In order to determine the valuable and trashy edges, respectively, at the beginning of a new iteration, we use the fact that the edges are already partially ordered. Recall that $W^k_i = W^k_{i+1} + w_{i+1} > W^k_{i+1}$ holds for all edges $e_i \in L^k$ and $W^k_{i+1} = W^k_i + w_{i+1} > W^k_i$ holds for all edges $e_i \in R^k$. Since the edges in $\bar{P}$ are already ordered, we have $W_i > W_{i+1}$ for all edges $e_i \in \bar{P}$.

Let x^{k-1} be an optimal solution of L^{k-1}. Notice that $L^k = L^{k-1} \cup e_{k-1}$ and $W^k_i = W^{k-1}_i + w_k > w_k = W^k_{k-1}$ holds for all $e_i \in L^k$. We distinguish two cases:

- There exists a valuable edge for x^{k-1} in L^{k-1} with respect to W^{k-1}: Due to the optimality of x^{k-1} we conclude that the trashy and valuable edges in L^{k-1} with respect to W^{k-1} are also trashy and valuable, respectively, with respect to W^k.

- If there does not exist a valuable edge for x^{k-1} in L^{k-1} with respect to W^{k-1} then all edges of L^{k-1} are fully reduced. Hence, e_{k-1} becomes valuable in L^k with respect to W^k. Furthermore, if $x^{k-1}_{k-1} > 0$ then e_{k-1} becomes also a trashy edge in L^k otherwise the edge that was trashy in L^{k-1} for x^{k-1} with respect to W^{k-1} remains trashy with respect to W^k.

A similar analysis yields for R^k:

- If e_{k-1} was valuable in R^{k-1} for x^{k-1} with respect to W^{k-1} then there does not exist any valuable edge for x^{k-1} in R^k with respect to W^k and e_k becomes a trashy edge in R^k.

- If e_{k-1} was not a valuable edge in R^{k-1} for x^{k-1} with respect to W^{k-1} then the trashy and valuable edges in R^{k-1} with respect to W^{k-1} are also trashy and valuable, respectively, in R^k with respect to W^k.

Since the weights of the edges in $\bar{P}$ remain unchanged the corresponding trashy and valuable edges in $\bar{P}$ do not change if a new iteration starts. Hence, given the trashy and valuable edges of the previous iteration we can determine the initial trashy and valuable edges of the new iteration in constant time.

Now we show how the valuable and trashy edges change if a shift operation is performed during iteration k. We distinguish two cases:

- $\delta = u_{e^+} - x_{e^+}$:
  If $e^+ = e_i \in L^k$ then the new valuable edge in L^k is equal to e_{i+1} (with $W^k_{i+1} = W^k_i - w_{i+1}$) and the new trashy edge in L^k is equal to e_i. If $e^+ = e_i \in R^k$ then the new valuable edge in R^k is equal to e_{i-1} (with $W^k_{i-1} = W^k_i - w_i$) and the new trashy edge in R^k is equal to e_i. And if $e^+ = e_i \in \bar{P}$ then the new valuable edge in $\bar{P}$ to e_{i+1} and the new trashy edge in $\bar{P}$ is equal to e_i.

- $\delta = x_{e^-}$:
  If $e^- = e_i \in L^k$ then the new trashy edge in L^k is equal to e_{i-1} (with $W^k_{i-1} = W^k_i + w_i$) and the new valuable edge in L^k is equal to e_i. If $e^- = e_i \in R^k$ then the new trashy edge in R^k is equal to e_{i+1} (with $W^k_{i+1} = W^k_i + w_{i+1}$) and the new valuable edge in R^k is equal to e_i. And if $e^- = e_i \in \bar{P}$ then the new trashy edge in $\bar{P}$ is equal to e_{i-1} (with $W^k_{i-1} = W^k_i + w_i$) and the new valuable edge in $\bar{P}$ is equal to e_i.

Summarizing, the trashy (valuable) shift edge can be determined by computing the minimum (maximum) weight among the trashy (valuable) edges. Since the trashy and valuable edges can be determined in constant time, we are now interested in the number of shift operations. The following observation is a direct consequence from Algorithm 1.

Observation 4.2. *The weight of the trashy shift edge is greater or equal to the weight of the previous trashy shift edge. Analogously, the weight of the new valuable shift edge is less or equal to the weight of the previous valuable shift edge.*

Furthermore, the following lemma holds.

Lemma 4.3. *An edge that has been a trashy (valuable) shift edge cannot become a valuable (trashy) shift edge within the same iteration.*

Proof. Assume that there exists an edge $e_i \in E$ that has been a trashy shift edge in iteration k and is now a valuable shift. Let e_j be the current trashy shift edge. From Observation 4.2 we conclude that $W_j^k \geq W_i^k$ since e_i has been a trashy shift edge before e_j becomes a trashy shift edge. This leads to a contradiction, because $W_j^k < W_i^k$ must hold if a shift operation is performed on e_i and e_j. □

In the following lemma it is proved that the investment into L^k only increases and the investment into R^k only decreases.

Lemma 4.4. *Let e^+ be a valuable shift edge and let e^- be a trashy shift edge, then*

$$e^+ \in L^k \cup \bar{P} \text{ and } e^- \in R^k \cup \bar{P}.$$

Proof. Assume that Algorithm 1 chooses $e_i \in L^k$ as trashy shift edge. Let $e_j \in R^k \cup \bar{P}$ be the corresponding valuable shift edge. Then, we know that $W_i^k < W_j^k$. Let x^{k-1} be an optimal solution of $LP(k-1)$. We distinguish two cases:

1. $x_i^{k-1} > 0$: Since $W_i^{k-1} \leq W_i^k < W_j^k \leq W_j^{k-1}$ holds and x_i^{k-1} is an optimal solution of $LP(k-1)$, we conclude that $x_j^{k-1} = u_j$. In order to achieve $x_j < u_j$ edge e_j must have been a trashy edge during iteration k. This leads to a contradiction to Lemma 4.3.

2. $x_i^{k-1} = 0$: In order to achieve $x_i > 0$ edge e_i must have been a valuable shift edge. This leads to a contradiction to Lemma 4.3.

The observation for valuable edges in R^k is proved in the same way. □

Unfortunately, it may happen, that the investment spent into $\bar{P}$ is increased in iteration k while it is decreased in iteration $k+1$ or vice versa. This is the reason why shift operations involving edges of $\bar{P}$ are of special interest and a more careful analysis is needed in order to achieve an $\mathcal{O}(n \log n)$ algorithm.

In the following we denote by y the amount of budget used for the edges in $\bar{P}$. In order to minimize the second term in (11), i.e., the amount the vertices in $\bar{P}$ contribute to the objective value of the reverse 2-median problem on a path, we totally reduce the lengths of the edges $e_p, e_{p+1}, \ldots, e_m$ for some $m \leq n-1$ and partially decrease edge e_{m+1}. In the remaining edges of $\bar{P}$ no budget is invested at all. Note that this special form is always maintained throughout the shift procedure. We assume that the following values are already determined in a preprocessing

$$U_i = \sum_{j=p}^{i} u_j \quad \text{for } i = p, \ldots, n-1.$$

Then the trashy edge $e^-(\bar{P})$ and the valuable edge $e^+(\bar{P})$ with respect to y can be determined in the following way:

$$\begin{aligned} e^+(\bar{P}) &= e_i \text{ such that } U_{i-1} \leq y < U_i, \\ e^-(\bar{P}) &= e_j \text{ such that } U_{j-1} < y \leq U_j. \end{aligned}$$

Furthermore, it is useful to introduce a function $z_{\bar{P}}(y)$ that measures the contribution of the vertices $\bar{P}$ to the original objective function depending on y. In order to give a formal definition, let y be the amount of investment spent into $\bar{P} = \{e_p, \ldots, e_{n-1}\}$ in such a way that if in edge e_i some money is invested the edges $e_p, \ldots, e_{i-1}$ are totally reduced. If we denote the modified edge lengths with respect to these changes by $\bar{l}_i(y)$ the function $z_{\bar{P}}(y)$ is defined as

$$z_{\bar{P}}(y) := \sum_{j=p+1}^{n} \left(w_j \sum_{i=p}^{j-1} \bar{l}_i(y) \right).$$

Obviously, the following equation holds

$$z_{\bar{P}}(y) = z_{\bar{P}}(U_{i-1}) - W_i(y - U_{i-1}),$$

where

$$z_{\bar{P}}(U_{i-1}) = \sum_{j=p}^{n-1} W_j l_j - \sum_{j=p}^{i} W_j u_j \quad \text{for } i = p, \ldots, n-1.$$

Using the notation introduced above the shift operation where some edges of $\bar{P}$ are involved can be explained in more detail. We distinguish two cases: Either $e^+(\bar{P})$ is a valuable shift edge or $e^-(\bar{P})$ is a trashy shift edge.

1. $e^+(\bar{P})$ is a valuable shift edge:
   Due to Lemma 4.4 the corresponding trashy shift edge e^- is element of R^k. Define

   $$\begin{aligned} W^+ &:= \min\{W_j : j = p, \ldots, n-1, W_j > W^k_{e^-(R^k)}, W_j \geq W^k_{e^+(L^k)}\}, \\ E^+(\bar{P}) &:= \{e_j \in \bar{P} : W^+ \leq W_j \leq W_{e^+(\bar{P})}\}. \end{aligned}$$

   Since $W_e \geq W^k_{e^+(L^k)}$ holds for all $e \in E^+(\bar{P})$, we conclude that $e^+(L^k)$ does not become a valuable shift edge unless all edges in $e \in E^+(\bar{P})$ are fully reduced. On the other hand $W_e > W^k_{e^-(R^k)}$ holds. Therefore, the solution is not optimal unless either the total investment is fully removed from $e^-(R^k)$ or all edges in $e \in E^+(\bar{P})$ are fully reduced.

Hence, the edges in $E^+(\bar{P})$ can be accumulated to one valuable shift edge with upper bound $u_{E^+(\bar{P})} = \sum_{e \in E^+(\bar{P})} u_e$ and investment $x_{E^+(\bar{P})} = \sum_{e_j \in E^+(\bar{P})} x_j = x_{e^+(\bar{P})}$. Observe, that W^+ can be found by using binary search. Let $W_i = W_{e^+(\bar{P})}$ and $W_j = W^+$ then $u_{E^+(\bar{P})} = U_j - U_{i-1}$. The shift operation is then performed in the following way:

- Let y be the current investment spent into edges of $\bar{P}$. Set $z' = z_{\bar{P}}(y)$.
- Determine $\delta = \min\{u_{E^+(\bar{P})} - x_{E^+(\bar{P})}, x_{e^-}\}$
- Set $y = y + \delta$ and $x_{e^-} = x_{e^-} - \delta$
- Determine the new valuable and trashy edge in $\bar{P}$.
- Compute $z_{\bar{P}}(y)$ and the new objective value $z(x) = z(x) + \delta W_{e^-} + z_{\bar{P}}(y) - z'$.
- Update the trashy edge in R^k.

2. $e^-(\bar{P})$ is a trashy shift edge:
Due to Lemma 4.4 the corresponding valuable shift edge e^+ is in L^k. Define

$$\begin{aligned} W^- &:= \max\{W_j : j = p, \dots, n-1, W_j \leq W^k_{e^-(R^k)}, W_j < W^k_{e^+(L^k)}\} \\ E^-(\bar{P}) &:= \{e_j \in \bar{P} : W_{e^-(\bar{P})} \leq W_j \leq W^-\} \end{aligned}$$

The edges in $E^-(\bar{P})$ are accumulated to one trashy shift edge. W^- can again be found by using binary search. Let $W_i = W^-$ and $W_j = W_{e^-(\bar{P})}$ then the upper bound of the accumulated edge is equal to $u_{E^+(\bar{P})} = U_j - U_{i-1}$. The shift operation involving $E^-(\bar{P})$ is performed in an analogue way.

Notice that the effort for one shift operation involving the accumulated shift edge is equal to $\mathcal{O}(\log n)$ because we have to do one binary search for the determination of W^+ and W^-, respectively, and a further binary search for the determination of the new valuable and trashy edge in $\bar{P}$. All further operations can be done in constant time.

See Algorithm 2 for a formal description of the developed algorithm for R2MP.

At the end of this section, we bound the running time of Algorithm 2. Consider a shift operation involving $e^+ \in L^k$ and $e^- \in R^k$: Either e^+ is fully reduced or the total investment into e^- is removed. If the shift operation results in a fully reduction of $e^+ \in L^k$ we charge the effort for this operation to this edge e^+. Otherwise, if the shift operation results in a fully removement of the investment of e^- we charge the effort of this operation to e^-. Notice that the investment into edges in L^k only increases and the investment into edges in R^k only decreases during the whole algorithm. Hence, an edge can be charged only once as valuable shift edge in L^k and only once as trashy shift edge in R^k.

Algorithm 2 Determines an optimal solution of R2MP

```
Preprocessing:
Determine $U_i$ and $z_{\bar{P}}(U_i)$ for $i = q, \dots, n-1$.
Solve $LP(1)$. Let $x^1$ be an optimal solution of $LP(1)$.
Main loop:
for $k$ from 2 to $p-1$ do
    Solve $LP(k)$ using Algorithm 1
    $x^k \Leftarrow$ optimal solution of $LP(k)$
    $z(x^k) \Leftarrow$ objective value of $x^k$
end for
Output:
$x^* = \operatorname{argmin}\{z(x^k) : k = 1, \dots, p-1\}$, $x^*$ is an optimal solution.
```

Notice that if there is no accumulated shift edge, i. e., the underlying graph is a cycle which was transformed to a path, then Algorithm 2 runs in $\mathcal{O}(n)$ time. Observe that this running time was already achieved in [7] for the reverse 1-median problem on a cycle.

In the general case shift operations including an accumulated edge have to be considered as well. Due to the definition of an accumulated edge we conclude that a shift operation including an accumulated edge is either followed by a shift operation on edges in L^k and R^k or results in an optimal solution of $LP(k)$. Thus, the number of shift operations including an accumulated edge (and hence the number of binary searches) is bounded by the number of shift operations between L^k and R^k plus the number of iterations. Since there are $\mathcal{O}(n)$ shift operations between L^k and R^k and $\mathcal{O}(n)$ iterations, we conclude that the number of shift operations including an accumulated edge is of the order $\mathcal{O}(n)$. These observations prove the following theorem:

Theorem 4.5. *Algorithm 2 determines an optimal solution of the reverse 2-median problem on a path in $\mathcal{O}(n \log n)$ time, where n is the number of vertices of P.*

5 Conclusion

In the previous sections, we outlined an $\mathcal{O}(n \log n)$ time algorithm for the reverse 2-median problem on a tree and the reverse 1-median problem on a cactus with just one cycle based on the unit cost model. We have shown that both problems are equivalent in the sense that they can be transformed to a reverse 2-median problem on a path. Moreover, it was assumed that the vertex weights are positive.

Since the problem is $\mathcal{NP}$-hard on general graphs, different other special graph classes should be investigated. Moreover, it is a topic for further research to extend the results in this paper to different cost functions and/or negative vertex weights. Another interesting problem arises if the allowable changes are not restricted to the edge lengths, but it is also possible to change some vertex weights.

References

[1] E. Balas, E. Zemel, An algorithm for large zero-one knapsack problems, Oper. Res. 28 (1980) 1130–1154.

[2] O. Berman, D.I. Ingco, and A. Odoni, Improving the location of minisum facilities through network modification, Ann. Oper. Res. 40 (1992), 1–16.

[3] O. Berman, D.I. Ingco, and A. Odoni, Improving the location of minimax facilities through network modification, Networks 24 (1994), 31–41.

[4] R.E. Burkard, B. Klinz, and J. Zhang, Bottleneck capacity expansion problems with general budget constraints, RAIRO Recherche Operationelle 35 (2001), 1–20.

[5] R.E. Burkard, Y. Lin, and J. Zhang, Weight reduction problems with certain bottleneck objectives, European J. Oper. Res. 153 (2004), 191–199.

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

[7] R.E. Burkard, E. Gassner, J. Hatzl, A Linear Time Algorithm for the Reverse 1-Median Problem on a Cycle, to appear in Networks.

[8] D. Burton, W.R. Pulleyblank, and Ph.L. Toint, The inverse shortest paths problem with upper bounds on shortest paths costs, Network optimization Gainesville, FL (1996), 156-171. Springer: Berlin, 1997.

[9] M.C. Cai, X.G. Yang, and J. Zhang, The complexity analysis of the inverse center location problem, J. Global Opt. 15 (1999) 213–218.

[10] K.U. Dragmeister, S.O. Krumke, M.V. Marathe, H. Noltemeier, and S.S. Ravi, Modifying edges of a network to obtain short subgraphs, Theoretical Comp. Sci. 203 (1998), 91–121.

[11] G.N. Frederickson and R. Solis-Oba, Increasing the weight of minimum spanning trees, J. of Algorithms 33 (1999), 244–266.

[12] D.R. Fulkerson and G.C. Harding, Maximizing the minimum source-sink path subject to a budget constraint, Math. Programming 13 (1977), 116–118.

[13] S.O. Krumke, M.V. Marathe, H. Noltemeier, R. Ravi, and S.S. Ravi, Approximation algorithms for certain network improvement problems, J. Comb. Opt. 2 (1998), 257–288.

[14] J. Zhang, Z. Liu, and Z. Ma, Some reverse location problems, Eur. J. Oper. Res. 124 (2000), 77–88.

[15] J. Zhang, X.G. Yang and M.C. Cai, Inapproximability and a polynomially solvable special case of a network improvement problem, Europ. J. of Operational Research 155 (2004), 251–257.

[16] J. Zhang, X.G. Yang and M.C. Cai, A network improvement problem under different norms, Computational Optimization and Applications 27 (2004), 305–319.