Jrl Syst Sci & Complexity (2008) 21: 651–664

INVERSE CENTER LOCATION PROBLEM ON A TREE*

Xiaoguang YANG · Jianzhong ZHANG

Received: 25 March 2008 / Revised: 29 August 2008

Abstract This paper discusses the inverse center location problem restricted on a tree with different costs and bound constraints. The authors first show that the problem can be formulated as a series of combinatorial linear programs, then an $O(|V|^2 \log |V|)$ time algorithm to solve the problem is presented. For the equal cost case, the authors further give an $O(|V|)$ time algorithm.

Key words Center location, combinatorial linear program, tree, two-terminal series parallel graphs.

1 Introduction

The center location problem, which is to find the "best" position for a facility in a network to minimize the distance from the facility to the farthest communities of the network, is a very practical operations research (OR) problem which has attracted much attention. The criterion of optimization exhibits equity of some facilities, such as police station, fire station, hospital, etc, which provide "emergency" service to all communities. The problem is called a minimax location problem and is well-solved, see, for example, [1–2].

But in an established network, the location of a facility has already been fixed. The changing environment might make the existing facility deviate from the center place of the network. For example, the travelling times on some links have changed due to the changes of the traffic flows. To restore the equity of the facility, we may need to modify the weights of edges in the network to "re-locate" the facility to the center of the network under the new weight function. For instance, we may alter the travelling time by controlling the traffic passing through an edge. This arises what we call the inverse center location problem. It is to modify the weights (lengths or travelling times) of a network as less possible as to make a given vertex become a center under the new weights. Heuburger[3] gave a comprehensive survey on the development of inverse combinatorial optimization problems.

We describe the inverse center location problem formally as follows.

Let $G = (V, E, w)$ be a connected graph, where V is a vertex set, E is an edge set, and $w : E \to R_+$ is a weight function. Let s be a specific vertex in V. The inverse center location problem is to change $w > 0$ to $w^* > 0$ such that

a) s becomes a center of G under w^*;

Xiaoguang YANG
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, *China.*
Email: xgyang@iss.ac.cn.
Jianzhong ZHANG
United International College, Hong Kong Baptist University, Zhuhai 519085, *China.*
*The research is supported by the National Natural Science Foundation of China under Grant Nos. 70425004, 70221001, and the National Key Research and Development Program of China under Grant No. 2002CB312004.

b) $-b^-(e) \le w^*(e) - w(e) \le b^+(e)$ for $e \in E$; and

c) the total cost incurred by the adjustment,

$$C(w^*) := \sum_{e\in E}[c^+(e)\max\{w^*(e)-w(e),0\}+c^-(e)\max\{w(e)-w^*(e),0\}],$$

is minimum.

In the above description, $b^+ > 0$ and $b^- > 0$ are bound constraints on the modification, and $c^+ > 0$ and $c^- > 0$ are corresponding unit modification costs, respectively. To ensure $w^* > 0$, we may assume that $b^-(e) < w(e)$.

In [4], the general inverse center location problem is shown to be strongly NP-hard even if there is no bound constraints and all unit modification costs $c^+(e)$ and $c^-(e)$ are equal. It is interesting to consider special cases which can be solved in polynomial times. In this paper, we find such a special case and show that the inverse center location problem on a tree is polynomially solvable, and provide a combinatorial strong polynomial algorithm to solve it.

The paper is organized as follows. In Section 2, we investigate the characteristic structure of the optimal solution for the inverse problem, and formulate the inverse problem by a series of combinatorial linear programs. A combinatorial algorithm running in $O(|V|^2\log(|V|))$ time is presented in Section 3. In Section 4, we show that the inverse problem can be solved in linear time if the unit modification costs $c(e)$ are identical. Some concluding remarks are given in Section 5.

2 Combinatorial Linear Program Formulation

Let $L(T)$ be the set of leaves of T, i.e., the vertices of degree 1. For each $v \in V$, let $R_w(v)$ denote the radius of v with respect to weight w, i.e., $R_w(v) = \max\{d_w(v,u) \mid u \in V\}$, where $d_w(v,u)$ is the distance between v and u under weight w, which is the length of shortest path between u and v under w. Since T is a tree, there is a unique path from v to u (denoted by $P(v,u)$), and

$$d_w(v,u) = \sum_{e\in P(v,u)} w(e).$$

Throughout this paper, we use (u,v) and $[u,v]$ for a pair of vertices and an edge with two endpoints u and v, respectively.

If $s \in L(T)$, then s is a center if and only if E is a singleton. In fact, if E is a singleton, there is only one edge in T, and the two vertices are both centers of T. So, there is no need to change weights. If E is not a singleton, for any inner vertex $u \in V \setminus L(T)$, we have $R_{w^*}(s) > R_{w^*}(u)$ since $w^* > 0$, that is, s cannot be a center under w^*.

Therefore, $s \in L(T)$ is a trivial case, and in the sequel, we always assume $s \notin L(T)$.

For each $l \in L(T)$, let $P(s,l)$ be the unique path between s and l, and $v(l)$ be the first vertex from s on $P(s,l)$.

Assertion 1 *Assume that $d_w(s,l_s) = R_w(s)$, i.e., $P(s,l_s)$ is the longest path from s to the leaf set. If $|P(s,l_s)| = 1$, i.e., $P(s,l_s) = \{[s,l_s]\}$, then s must be a center of T under w.*

Proof For any vertex $v \in V$, the path from v to the leaf l_s must take (s,l_s) as the last edge, as $|P(s,l_s)| = 1$ implies that $P(s,l_s)$ contains edge $[s,l_s]$ only. So, $R_w(v) \ge d_w(v,l_s) \ge d_w(s,l_s)$ and hence $R_w(s) \le R_w(v)$ for all $v \in V$, i.e., s is a center of T under w. ∎

Therefore, below we consider only the case that $s \in V \setminus L(T)$ and $|P(s,l_s)| > 1$.

If we delete edge $[s,v(l_s)]$ from E, then T is divided into two disjoint subtrees. We denote by T_1 the subtree rooted at $v(l_s)$, and by T_2 the other subtree rooted at s. Let $L(T_1)$ and $L(T_2)$

be the leaf sets of T_1 and T_2, respectively, with an exception that the roots $v(l_s)$ and s are not put into these two sets even if their degrees in T_1 or T_2 are 1.

Assertion 2 *s is a center under w if and only if*

$$d_w(s, l_s) \leq \max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\}. \tag{1}$$

Proof Suppose (1) is true. In order to show that s is a center, let us prove that

$$R_w(s) \leq R_w(v), \quad \text{for all} \ \ v \in V. \tag{2}$$

By the definition of l_s, we have $R_w(s) = d_w(s, l_s)$.

We consider the following three cases.

Case 1. For any $v \in V(T_2)$ and $v \neq s$, we have $d_w(v, l_s) > d_w(s, l_s)$ since all paths between $v \in V(T_2)$ and l_s must pass s and $w > 0$. Hence, $R_w(v) > d_w(v, l_s) = R_w(s)$.

Case 2. Consider the vertex $v(l_s)$. By (1), we have

$$R_w(v(l_s)) \geq \max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\} \geq R_w(s).$$

Case 3. Consider any $u \in V(T_1)$ and $v \neq v(l_s)$. Since all paths between u and $v \in L(T_2)$ must pass $v(l_s)$, we have

$$R_w(u) \geq \max\{d_w(u, v) \mid \forall v \in L(T_2)\} \geq \max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\}.$$

Hence, again by (1), $R_w(u) \geq R_w(s)$.

Combining the above three cases, we know that (2) is true and s is a center of T under w.

Conversely, suppose s is a center under w, we prove (1) by contradiction. If (1) is not true, then

$$\max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\} < d_w(s, l_s) = R_w(s). \tag{3}$$

Since any path from s to every vertex in $L(T_1)$ must pass through $v(l_s)$ and $w([s, v(l_s)]) > 0$, we have

$$\begin{aligned} &\max\{d_w(v(l_s), v) \mid v \in L(T_1)\} \\ &= \max\{d_w(s, v) - w([s, v(l_s)]) \mid v \in L(T_1)\} \\ &= d_w(s, l_s) - w([s, v(l_s)]) \\ &< d_w(s, l_s) = R_w(s). \end{aligned} \tag{4}$$

Combining (3) and (4), we obtain that

$$\begin{aligned} R_w(v(l_s)) &= \max\{\max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\}, \max\{d_w(v(l_s), v) \mid v \in L(T_1)\}\} \\ &< R_w(s). \end{aligned}$$

This means that s is not a center of T, a contradiction. ∎

By Assertions 1 and 2, we know that if $s \in V \setminus L(T)$ is not a center of the tree under w, then we have $|P(s, l_s)| > 1$ and $d_w(s, l_s) > \max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\}$.

Assertion 3 *Suppose that there are two leaves $l_1, l_2 \in L(T)$ such that $d_w(s, l_1) = d_w(s, l_2) = R_w(s)$, and $P(s, l_1) \cap P(s, l_2) = \emptyset$, i.e., there are no common edges between path $P(s, l_1)$ and path $P(s, l_2)$, then s is a center of T under w.*

Proof Delete s from T, T can be divided into at least two subtrees. Let T_1' be the subtree containing l_1, and T_2' be the subtree containing l_2, and let $V' = V \setminus (T_1' \cup T_2' \cup \{s\})$. For any $v \in T_1'$, we have $R_w(v) \geq d_w(v, l_2) \geq d_w(s, l_2) = R_w(s)$ since the path between v and l_2 must pass s. Similarly, for any $v \in T_2'$, we have $R_w(v) \geq d_w(v, l_1) \geq d_w(s, l_1) = R_w(s)$. For any $v \in V'$, we have $R_w(v) \geq d_w(v, l_1) \geq d_w(s, l_1) = R_w(s)$, too. Hence, s is a center of T. ▮

Assertion 4 *Assume that $s \in V \setminus L(T)$ is not a center of the tree under w and the inverse center location problem is feasible, then there exists an optimal solution w^* such that*

i) $R_{w^*}(s) = d_{w^*}(s, l_s) \leq R_w(s)$;

ii) $w^*(e) \leq w(e)$ *for* $e \in T_1$;

iii) *there is only one* $l^+ \in L(T_2)$ *such that* $w^*(e) \geq w(e)$ *for* $e \in P(s, l^+)$, *and* $d_{w^*}(v(l_s), l_s) = d_{w^*}(s, l^+)$;

iv) *and* $w^*(e) = w(e)$ *for all other* $e \in E \setminus (T_1 \cup P(v(l_s), l^+))$.

Proof As the inverse center location problem is feasible, the optimal solution w^* exists. First we show that $R_{w^*}(s) \leq d_w(s, l_s)$.

If $R_{w^*}(s) > d_w(s, l_s)$, let $l' \in L(T)$ be a leaf such that

$$d_{w^*}(s, l') = \max\{d_{w^*}(s, l) \mid l \in L(T)\}.$$

We consider the following two cases.

Case 1. $l' \in L(T_2)$. By the assumptions made on l' and s, we have

$$d_{w^*}(s, l') > d_w(s, l_s) > \max\{d_w(v(l_s), v) \mid \forall v \in L(T_2)\}.$$

Consider $[s, v']$ the first edge from s on the path $P(s, l')$ (We call it a focus edge). Delete $[s, v']$ from T, we obtain two subtrees T_1' and T_2' such that $T_1 \subset T_1'$ and $T_2' \subset T_2$.

Define a new weight vector such that

$$\overline{w}(e) = \begin{cases} w(e), & e \in T_1', \\ w^*(e), & e \in T_2' \end{cases}$$

and

$$\overline{w}([s, v']) = \begin{cases} w^*([s, v']), & w^*([s, v']) \leq w([s, v']), \\ \max\{w([s, v']), w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s)\}, & w^*([s, v']) > w([s, v']). \end{cases}$$

From the definition of $\overline{w}$, we can see that either $\overline{w}(e) = w(e)$ or $\overline{w}(e) = w^*(e)$, except that $\overline{w}(s, v') = w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s) \leq w^*([s, v'])$ when $w^*([s, v']) > w([s, v'])$ and $w([s, v']) \leq w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s)$. Hence, we know that $\overline{w}$ is feasible and $C(\overline{w}) \leq C(w^*)$.

Let us consider $d_{\overline{w}}(s, l_s)$ and $d_{\overline{w}}(s, l')$. It is directly seen that

$$d_{\overline{w}}(s, l_s) = \max\{d_{\overline{w}}(s, l) \mid l \in L(T_1')\} = \max\{d_w(s, l) \mid l \in L(T_1')\} = d_w(s, l_s). \tag{5}$$

Moreover, if $w^*([s, v']) \leq w([s, v'])$, we have

$$d_{\overline{w}}(s, l') = \max\{d_{\overline{w}}(s, l) \mid l \in L(T_2')\} = \max\{d_{w^*}(s, l) \mid l \in L(T_2')\} = d_{w^*}(s, l') > d_w(s, l_s);$$

if $w^*([s, v']) > w([s, v'])$, then either

$$d_{\overline{w}}(s, l') = w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s) + d_{w^*}(v', l') = d_w(s, l_s),$$

if $w([s, v']) \leq w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s)$; or

$$\begin{aligned} d_{\overline{w}}(s, l') &= w([s, v']) + d_{w^*}([v', l']) \\ &> w^*([s, v']) - d_{w^*}(s, l') + d_w(s, l_s) + d_{w^*}([v', l']) \\ &= d_w(s, l_s), \end{aligned}$$

otherwise.

Then we conclude

$$d_{\overline{w}}(s,l') \geq d_w(s,l_s). \tag{6}$$

Furthermore, for $l \in L(T_1')$, we have $d_{\overline{w}}(s,l) = d_w(s,l)$ since $\overline{w}(e) = w(e)$ for $e \in T_1'$ and hence $d_{\overline{w}}(s,l) \leq d_w(s,l_s)$ since $d_w(s,l) \leq d_w(s,l_s)$; for $l \in L(T_2')$, we have $d_{\overline{w}}(s,l) = \overline{w}([s,v']) + d_{w^*}(v',l) \leq d_{\overline{w}}(s,l')$ since $\overline{w}(e) = w^*(e)$ for $e \in T_2'$ and $d_{w^*}(v',l) \leq d_{w^*}(v',l')$ for $l \in L(T_2')$ by the definition of l'. Therefore, we have $d_{\overline{w}}(s,l') = R_{\overline{w}}(s)$.

If $d_{\overline{w}}(s,l') = d_w(s,l_s)$, by Assertion 2, we know s is a center under $\overline{w}$.

If $d_{\overline{w}}(s,l') > d_w(s,l_s)$, we consider the next edge $[v',v'']$ after $[s,v']$ from s on the path $P(s,l')$ as a focus edge.

Let us delete $[v',v'']$ from T to obtain two subtrees T_1'' and T_2''. We have $T_1' \subset T_1''$ and $T_2'' \subset T_2'$.

Define $\overline{w}'$ such that $\overline{w}'(e) = w(e)$ for $e \in T_1'' \setminus \{[s,v']\}$, $\overline{w}'(e) = w^*(e)$ for $e \in T_2''$, $\overline{w}'([s,v']) = \overline{w}([s,v'])$, and

$$\overline{w}'([v',v'']) = \begin{cases} w^*([v',v'']), & w^*([v',v'']) \leq w([v',v'']), \\ \max\{w([v',v'']), w^*([v',v'']) - d_{\overline{w}}(s,l') + d_w(s,l_s)\}, & w^*([v',v'']) > w([v',v'']). \end{cases}$$

It is clear that $\overline{w}'(e) \leq w(e)$ for $e \in T_1''$, $\overline{w}'(e) = w^*(e)$ for $e \in T_2''$, and $C(\overline{w}') \leq C(\overline{w}) \leq C(w^*)$.

Using the similar arguments, it is not difficult to show that $\overline{w}'$ is feasible, $R_{\overline{w}'}(s) = d_{\overline{w}'}(s,l') \geq d_{\overline{w}'}(s,l_s) = d_w(s,l_s)$, $d_{\overline{w}'}(s,l') \leq d_{\overline{w}}(s,l')$.

Similarly, if $d_{\overline{w}'}(s,l') = d_{\overline{w}'}(s,l_s)$, s becomes the center under $\overline{w}'$; if $d_{\overline{w}'}(s,l') > d_{\overline{w}'}(s,l_s)$, we go further to consider the next edge $[v'',v''']$ on the path $P(s,l')$ as a focus edge, and repeat the above procedure.

Note that if the procedure does not stop, the weights of the focus edges are not greater than their original weights during the procedure, e.g., $\overline{w}''([s,v']) \leq w([s,v'])$, $\overline{w}''([v',v''] \leq w([v',v''])$, $\overline{w}''([v'',v''']) \leq w([v'',v'''])$, etc. Since $d_{w^*}(s,l') > d_w(s,l_s) \geq d_w(s,l')$ and the path $P(s,l')$ has finite edges, the procedure must stop with a $\overline{w}^*$ such that $d_{\overline{w}^*}(s,l') = d_{\overline{w}^*}(s,l_s) = R_{\overline{w}^*}(s)$. By Assertion 3, we know that s is a center of T under $\overline{w}^*$. Moreover, as the modification costs never increase in the procedure, we have $C(\overline{w}^*) \leq C(w^*)$.

Case 2. $l' \in L(T_1)$.

From Assertion 2, we know that s is the center under w^* if and only if

$$d_{w^*}(s,l') \leq \max\{d_{w^*}(v(l_s),l) \mid \forall l \in L(T_2)\}.$$

Note that $d_{w^*}(s,l') = w^*([s,v(l_s)]) + d_{w^*}(v(l_s),l')$ and $d_{w^*}(v(l_s),l) = w^*([s,v(l_s)]) + d_{w^*}(s,l)$ for $l \in L(T_2)$. Therefore, without loss of generality, we can assume $w^*([s,v(l_s)]) = w([s,v(l_s)])$. Moreover, we can easily get

$$d_{w^*}(v(l_s),l') \leq \max\{d_{w^*}(s,l) \mid \forall l \in L(T_2)\}.$$

Hence, there exists $l'' \in L(T_2)$ such that

$$d_{w^*}(s,l'') = \max\{d_{w^*}(s,l) \mid \forall l \in L(T_2)\} \geq d_{w^*}(v(l_s),l') > d_w(v(l_s),l_s),$$

the last inequality holds because $d_{w^*}(s,l') > d_w(s,l)$ by the assumption on l', and $w^*([s,v(l_s)]) = w([s,v(l_s)])$.

Define $\overline{w}$ as

$$\overline{w}(e) = \begin{cases} w(e), & e \in T_1, \\ w^*(e), & e \in T_2, \\ w([s, v(l_s)]), & e = [s, v(l_s)]. \end{cases}$$

We have $d_{\overline{w}}(v(l_s), l_s) = d_w(v(l_s), l_s)$, and $d_{\overline{w}}(s, l'') = \max\{d_{w^*}(s, l) \mid l \in L(T_2)\} = d_{w^*}(s, l'') > d_w(v(l_s), l_s)$.

We further claim that $\max\{d_w(s, l) \mid l \in L(T_2)\} < d_w(v(l_s), l_s)$.

In fact, if $\max\{d_w(s, l) \mid l \in L(T_2)\} \geq d_w(v(l_s), l_s)$, then $d_w(s, l_s) = w([s, v(l_s)]) + d_w(v(l_s), l_s) \leq \max\{d_w(v(l_s), l) \mid l \in L(T_2)\}$. Hence, s is already the center under w. This contradicts with the assumption that s is not a center under w.

Now let $[s, v']$, the first edge from s on the path $P(s, l'')$, be a focus edge. Delete $[s, v']$ from T, we obtain two subtrees T_1' and T_2' such that $T_1 \subset T_1'$ and $T_2' \subset T_2$.

Define a new weight vector such that

$$\overline{w}(e) = \begin{cases} w(e), & e \in T_1', \\ w^*(e), & e \in T_2'; \end{cases}$$

and

$$\overline{w}([s, v']) = \begin{cases} w^*([s, v']), & w^*([s, v']) \leq w([s, v']), \\ \max\{w([s, v']), w^*([s, v']) - d_{w^*}(s, l'') + d_w(v(l_s), l_s)\}, & w^*([s, v']) > w([s, v']). \end{cases}$$

Use the same technique as Case 1, we can construct $\overline{w}$ by reducing the over-increasing on T_2 such that

$$\begin{aligned} &d_{\overline{w}}(s, l_s) = d_w(s, l_s) = R_{\overline{w}}(s), \\ &d_{\overline{w}}(s, l'') = \max\{d_{\overline{w}}(s, l) \mid l \in L(T_2')\} = d_{\overline{w}}(v(l_s), l_s) = d_w(v(l_s), l_s). \end{aligned}$$

Therefore, we obtain that

$$\begin{aligned} d_{\overline{w}}(s, l_s) &= w([s, v(l_s)]) + d_w(v(l_s), l_s) \\ &\leq w([s, v(l_s)]) + d_{\overline{w}}(s, l'') \\ &\leq \max\{d_{\overline{w}}(v(l_s), l) \mid l \in L(T_2')\}. \end{aligned}$$

By Assertion 2, s is also a center of T under $\overline{w}$. Once again, $C(\overline{w}) \leq C(w^*)$.

Combining both cases, we have $R_{w^*}(s) \leq R_w(s)$.

Second, we prove $R_{w^*}(s) = d_{w^*}(s, l_s)$.

By the assumptions on w^* and l_s, we have $R_{w^*}(s) \geq d_{w^*}(s, l_s)$. If $R_{w^*}(s) > d_{w^*}(s, l_s)$, we have $d_{w^*}(s, l_s) < d_w(s, l_s)$ from $R_{w^*}(s) \leq R_w(s) = d_w(s, l_s)$.

The same as before, let $l' \in L(T)$ be a leaf such that $d_{w^*}(s, l') = \max\{d_{w^*}(s, l) \mid l \in L(T)\}$. We discuss two cases.

Case A. $l' \in L(T_2)$. We now prove that there exists $\overline{w}$ such that $d_{\overline{w}}(s, l_s) = R_{\overline{w}}(s) = R_{w^*}(s) = d_{\overline{w}}(s, l')$, and $C(\overline{w}) \leq C(w^*)$.

By the the assumptions on l_s, w^*, and l', we have $d_w(s, l_s) \geq d_{w^*}(s, l') > d_{w^*}(s, l_s)$.

Let u' be the first vertex from l_s on path $P(s, l_s)$. First of all, let $\overline{w}(e) = w^*(e)$ for all $e \in E$, and update $\overline{w}([u', l_s]) = \min\{w([u', l_s]), w^*([u', l_s]) + R_{w^*}(s) - d_{w^*}(s, l_s)\}$ if $w^*([u', l_s]) < w([u', l_s])$.

It is easy to see that under $\overline{w}$ we have

$$R_{\overline{w}}(s) = R_{w^*}(s) = d_{\overline{w}}(s, l') = \max\{d_{\overline{w}}(s, l) \mid l \in L(T_2)\}, \tag{7}$$
$$\max\{d_{\overline{w}}(s, l) \mid l \in L(T_1)\} \le R_{w^*}(s), \tag{8}$$
$$d_{w^*}(s, l_s) \le d_{\overline{w}}(s, l_s) \le R_{w^*}(s). \tag{9}$$

If $d_{\overline{w}}(s, l_s) = R_{\overline{w}}(s)$, s is a center under $\overline{w}$ by Assertion 3. Otherwise, we have $w^*([u', l_s]) \ge w([u', l_s])$. Let u'' be the next vertex of u' on the path $P(s, l_s)$. Denote by $L(u'') \subset L(T_1)$ the set of leaves connected with s via u''.

For each $q \in L(u'')$ and $q \ne l_s$, if $d_{\overline{w}}(s, q) > d_{\overline{w}}(s, l_s)$, we have $\overline{w}([u', l_s]) < d_{\overline{w}}(u', q)$, and $w([u', l_s]) \ge d_w(u', q)$ since $d_w(s, l_s) \ge d_w(s, q)$. Hence, $d_{\overline{w}}(u', q) > d_w(u', q)$.

Now let us set $\widehat{w}([u'', u'])$ by $\min\{w([u'', u']), \overline{w}([u'', u']) + R_{\overline{w}}(s) - d_{\overline{w}}(s, l_s)\}$ if $\overline{w}([u'', u']) < w([u'', u'])$. Let Δ be the difference between the new $\widehat{w}([u'', u'])$ and the old $\overline{w}([u'', u'])$. Let us update $d_{\widehat{w}}(u', q)$ by $\max\{d_{\overline{w}}(u', q) - \Delta, d_w(u', q)\}$. It is clear that $d_{\widehat{w}}(s, l_s) \le R_{w^*}(s)$. Moreover, we claim $d_{\widehat{w}}(s, q) \le R_{w^*}(s)$.

In fact, by the assumption on $d_{\widehat{w}}(u', q)$, we have

$$d_{\widehat{w}}(s, q) = d_{\overline{w}}(s, u') + \Delta + d_{\widehat{w}}(u', q) \le d_{\overline{w}}(s, u') + \Delta + \max\{d_{\overline{w}}(u', q) - \Delta, d_w(u', q)\}.$$

If $d_{\overline{w}}(u', q) - \Delta \ge d_w(u', q)$, we have

$$d_{\widehat{w}}(s, q) = d_{\overline{w}}(s, q) \le R_{w^*}(s),$$

otherwise we have

$$\begin{aligned} d_{\widehat{w}}(s, q) &= d_{\widehat{w}}(s, u') + d_w(u', q) \\ &\le d_{\widehat{w}}(s, u') + w([u', l_s]) \\ &= d_{\widehat{w}}(s, l_s) \le R_{w^*}(s). \end{aligned}$$

Therefore, (7)–(9) are still true for $\widehat{w}$.

Therefore, step by step, we can obtain a $\overline{w}$ satisfying (7)–(9) and

$$d_{\overline{w}}(s, l_s) = R_{\overline{w}}(s) = R_{w^*}(s).$$

By Assertion 3, s is a center under $\overline{w}$, and it is clear that $C(\overline{w}) \le C(w^*)$.

Case B. $l' \in L(T_1)$. By Assertion 2, we have

$$d_{w^*}(s, l') \le \max\{d_{w^*}(v(l_s), l) \mid \forall l \in L(T_2)\}.$$

It is clear that $w^*([s, v(l_s)]) = w([s, v(l_s)])$, for otherwise we set $w^*([s, v(l_s)]) = w([s, v(l_s)])$ which will make s be a center of T under w^*, and the cost incurred by changing $w([s, v(l_s)])$ to $w^*([s, v(l_s)])$ is unnecessary.

Hence, we have

$$\max\{d_{w^*}(s, l) \mid \forall l \in L(T_2)\} \ge d_{w^*}(v(l_s), l') = R_{w^*}(s) - w([s, v(l_s)]).$$

Since $d_{w^*}(s, l_s) < R_{w^*}(s)$ and $d_w(s, l_s) < d_w(s, l')$, we have

$$\begin{aligned} &d_{w^*}(v(l_s), l_s) < d_{w^*}(v(l_s), l'), \\ &d_w(v(l_s), l_s) \ge d_{w^*}(v(l_s), l'). \end{aligned}$$

Using the technique in Case A, we can construct a $\overline{w}$ by restoring the over-reduction on T_1 such that

$$d_{\overline{w}}(v(l_s), l_s) = d_{w^*}(v(l_s), l'),$$
$$\max\{d_{\overline{w}}(s, l) \mid \forall l \in L(T_2)\} = \max\{d_{w^*}(s, l) \mid \forall l \in L(T_2)\}.$$

Hence, by Assertion 2, s is a center of T under $\overline{w}$. Moreover, we have $C(\overline{w}) \le C(w^*)$.

Therefore, we can conclude that

$$d_{w^*}(s, l_s) = R_{w^*}(s) \le R_w(s).$$

Third, as $d_w(v(l_s), l_s) > \max\{d_w(s, l) \mid l \in L(T_2)\}$, in order to make $\overline{w}^*$ satisfy that $d_{w^*}(v(l_s), l_s) \le \max\{d_{w^*}(s, l) \mid l \in L(T_2)\}$, we only need to decrease (never increase) weights of edges in T_1, and increase (never decrease) weights of edges in T_2, and we only need to make $d_{w^*}(v(l_s), l_s) = d_{w^*}(s, l^+)$ since any over-increment or over-decrement is useless.

Moreover, let $l^+ = \arg\max\{d_{w^*}(v(l_s), v) \mid \forall v \in L(T_2)\}$. Obviously, we only need to modify the weights of $P(v(l_s), l^+)$, and keep the weights of other edges in T_2 unchanged.

Furthermore, as pointed out in Case B, $w^*([s, v(l_s)]) = w([s, v(l_s)])$. Therefore, we conclude that

$$\begin{aligned} w^*(e) &\le w(e), && e \in T_1, \\ w^*(e) &\ge w(e), && e \in P(s, l^+), \\ w^*(e) &= w(e), && e \in T \setminus (T_1 \cup P(v(l_s), l^+)). \end{aligned}$$

The assertion is proved. ∎

Define

$$\widetilde{w}(e) = \begin{cases} w(e) - b^-(e), & e \in T_1, \\ w(e) + b^+(e), & e \in T_2. \end{cases}$$

Combining Assertions 2 and 4, we have that the inverse center location problem is feasible if and only if

$$\min\{d_{\tilde{w}}(v(l_s), l) \mid l \in L(T_1)\} \le \max\{d_{\tilde{w}}(s, l) \mid l \in L(T_2)\}.$$

By the special structure of tree, the computation for all $\{d_{\tilde{w}}(v(l_s), l) \mid l \in L(T_1)\} \cup \{d_{\tilde{w}}(s, l) \mid l \in L(T_2)\}$ and finding the minimum of $\{d_{\tilde{w}}(v(l_s), l), \mid l \in L(T_1)\}$ and maximum of $\{d_{\tilde{w}}(s, l) \mid l \in L(T_2)\}$ can be done in linear time. Therefore,

Theorem 1 *Checking the feasibility of the inverse center location problem restricted on a tree can be done in linear time.*

By Assertion 4, we can formulate the inverse center location problem by a series of combinatorial linear programming problems.

For any $l^+ \in L(T_2)$, we can define a combinatorial linear program $LP(l^+)$ as follows:

$$\begin{aligned} \min\ & \sum_{e \in T_1} c^-(e) x^-(e) + \sum_{e \in P(s, l^+)} c^+(e) x^+(e) \\ \text{s.t.}\ & \sum_{e \in P(v(l_s), l)} [w(e) - x^-(e)] \le \sum_{e \in P(v(l_s), l_s)} [w(e) - x^-(e)], \ l \in L(T_1) \setminus \{l_s\}, \\ & \sum_{e \in P(v(l_s), l_s)} [w(e) - x^-(e)] = \sum_{e \in P(s, l^+)} [w(e) + x^+(e)], \\ & 0 \le x^-(e) \le b^-(e), \qquad e \in T_1, \\ & 0 \le x^+(e) \le b^+(e), \qquad e \in P(s, l^+). \end{aligned} \tag{10}$$

From Assertion 4, we know that if the optimal modification w^* exists, it must be an optimal solution of some problems $LP(l^+)$. Hence, if each $LP(l^+)$ is infeasible, the inverse center location problem is infeasible. We can check the feasibility of each $LP(l^+)$ in linear time (see the next section). Note that the optimal solution of some problems $LP(l^+)$ may not correspond to a feasible solution of the inverse center location problem, i.e., s may not be a center under the modified weights. Therefore, when we enumerate all possible cases of problems $LP(l^+)$, we need to judge whether the optimal solution of $LP(l^+)$ corresponds to a feasible solution of the inverse center location problem. We can do this in the following way.

If $LP(l^+)$ is infeasible, discard it. If $LP(l^+)$ has an optimal solution w^+, let $w'(e) = w^+(e)$ for $e \in T_1 \cup P(s, l^+)$ and $w'(e) = w(e)$ for $e \in E \setminus (T_1 \cup P(s, l^+))$. Check whether s is a center under w'. If the answer is no, discard $LP(l^+)$. The one with minimum objective value of remaining problems $LP(l^+)$ corresponds to an optimal solution of the inverse center location problem.

Notice that the number of variables in $LP(l^+)$ is $|T_1| + |P(s, l^+)| \le |V| - 1$, the number of constraints is $|L(T_1)| + 2|T_1| + 2|P(s, l^+)| \le 3|V|$, and the coefficients of variables $x^+(e), x^-(e)$ are ± 1 or 0. $LP(l^+)$ is a combinatorial linear program in the definition of Tardos[5], and it can be solved in a strongly polynomial time. Moreover since the number of $LP(l^+)$s is $|L(T_2)| < |V|$, we can conclude that in the following theorem.

Theorem 2 *The inverse center location problem restricted on a tree can be solved in a strongly polynomial time.*

3 Combinatorial Polynomial Algorithm

Let us consider $LP(l^+)$. Denote $L(l) = d_w(v(l_s), l)$ for $l \in L(T_1)$, and $L(l^+) = d_w(s, l^+)$. Since we only need to decrease the weights on T_1 and increase the weights on $P(s, l^+)$, for notation simplicity, we can ignore the superscript $\bullet^+$, $\bullet^-$ of c^+, c^-, b^+, b^- and x^+, x^-, and re-write $LP(l^+)$ as follows:

$$\begin{aligned} \min\ & \sum_{e \in T_1} c(e)x(e) + \sum_{e \in P(s,l^+)} c(e)x(e) \\ \text{s.t.}\ & R + \sum_{e \in P(v(l_s),l)} x(e) \ge L(l), && l \in L(T_1), \\ & -R + \sum_{e \in P(s,l^+)} x(e) \ge -L(l^+), \\ & 0 \le x(e) \le b(e), && e \in T_1, \\ & 0 \le x(e) \le b(e), && e \in P(s, l^+), \end{aligned} \tag{11}$$

where the value of variable R in the optimal solution corresponds to $d_{w^*}(v(l_s), l_s) = d_{w^*}(s, l^+)$.

Note that by the tree structure, computing all $\{d_w(v(l_s), l) \mid l \in L(T_1)\} \cup \{d_w(s, l) \mid l \in L(T_2)\}$ and $\Big\{ \sum_{e \in P(v(l_s),l)} b(e) \mid l \in L(T_1) \Big\} \cup \Big\{ \sum_{e \in P(s,l)} b(e) \mid l \in L(T_2) \Big\}$ can be done in linear time. For instance, setting $\pi(s) = 0$, and letting $\pi(v) = \pi(u) + w([u, v])$ from root s to leaves in T_2, $\pi(v)$ then corresponds to the distance from s to v under w, and $\pi(l) = d_w(s, l)$ for $l \in L(T_2)$.

It is straightforward to prove that (11) is feasible if and only if

$$\max \Big\{ L(l) - \sum_{e \in P(v(l_s),l)} b(e) \mid l \in L(T_1) \Big\} \le L(l^+) + \sum_{e \in P(s,l^+)} b(e). \tag{12}$$

Therefore, checking the feasibility of all $LP(l^+)$ for $l^+ \in L(T_2)$ can be done in linear time. Below we only consider problems $LP(l^+)$ which have feasible solutions.

Consider the dual problem of linear program (11). It can be written as

$$\begin{aligned}
\min \quad & \sum_{e \in T_1 \cup P(s,l^+)} b(e)z(e) + L(l^+)y(l^+) - \sum_{l \in L(T_1)} L(l)y(l) \\
\text{s.t.} \quad & \sum \{y(l) \mid P(v(l_s), l) \ni e, \quad l \in L(T_1)\} - z(e) \le c(e), \qquad \forall e \in T_1, \\
& y(l^+) - z(e) \le c(e), \qquad e \in P(s, l^+), \\
& \sum_{l \in L(T_1)} y(l) - y(l^+) = 0, \\
& y(l) \ge 0, \qquad \forall l \in L(T_1) \cup \{l^+\}, \\
& z(e) \ge 0, \qquad \forall e \in T_1 \cup P(s, l^+).
\end{aligned} \tag{13}$$

Now construct an auxiliary directed network $N = (V^+, A^+, c^+, k^+)$. Let $V^+ = V(T_1) \cup V(P(s, l^+)) \cup \{t\}$ be the vertex set of N, where t is an extra transit vertex. For each $[u, v] \in T_1$, we assign two arcs $e^1 = [u, v]$ and $e^2 = [u, v]$, both oriented from $v(l_s)$, i.e., u is nearer to $v(l_s)$ than v. For each $[u, v] \in P(s, l^+)$, we assign two arcs $e^1 = [u, v]$ and $e^2 = [u, v]$, too, and both oriented from l^+, i.e., u is nearer to l^+ than v. From each $l \in L(T_1)$, we assign an arc $e = [l, t]$. From t we assign an arc $e = [t, l^+]$. Namely, the arc set of N can be written as

$$A^+ = \{e^1 = [u, v], \ e^2 = [u, v] \mid [u, v] \in T_1 \cup P(s, l^+)\} \cup \{[l, t] \mid l \in L(T_1)\} \cup \{[t, l^+]\}.$$

c^+ and k^+ are capacity vector and cost vector of N, respectively, which are defined as follows:

$$c^+(e) = \begin{cases} c([u, v]), & e = e^1 = [u, v] \in A^+, \\ +\infty, & \text{otherwise.} \end{cases}$$

$$k^+(e) = \begin{cases} 0, & \forall e = e^1 = [u, v] \in A^+, \\ b([u, v]), & \forall e = e^2 = [u, v] \in A^+, \\ -L(l), & \forall e = [l, t] \in A^+, \\ L(l^+), & e = [t, l^+]. \end{cases}$$

Assertion 5 *Linear program* (13) *is equivalent to finding a minimum cost flow from* $v(l_s)$ *to* s *on* N.

Proof First, suppose f be a minimum cost flow from $v(l_s)$ to s on N. Define $y(l) = f(l, t)$ for any $l \in L(T_1)$, and $z(e) = f(e^2)$ for any $e \in T_1 \cup P(s, l^+)$. It is easy to verify that (y, z) is a feasible solution of (13).

Moreover, the minimum cost is

$$\sum_{e \in A^+} k^+(e)f(e) = \sum_{e \in T_1 \cup P(s,l^+)} b(e)z(e) + L(l^+)y(l^+) - \sum_{l \in L(T_1)} L(l)y(l),$$

which is just the objective value of (13) at the feasible solution (y, z).

Conversely, if (y, z) is an optimal solution of (13), we have

$$z(e) = \begin{cases} \max\left\{0, \ \sum\{y(l) \mid P(v(l_s), l) \ni e, \ l \in L(T_1)\} - c(e)\right\}, & e \in T_1, \\ \max\{0, \ y(l^+) - c(e)\}, & e \in P(s, l^+). \end{cases}$$

Define

$$\begin{aligned}
&f(e^1) = \min\{c(e),\ \textstyle\sum\{y(l) \mid P(v(l_s), l) \ni e,\ l \in L(T_1)\}\}, && \forall e \in T_1,\\
&f(e^1) = \min\{c(e),\ y(l^+)\}, && \forall e \in P(s, l^+),\\
&f(e^2) = z(e), && \forall e \in T_1 \cup P(s, l^+),\\
&f(l, t) = y(l), && \forall l \in L(T_1),\\
&f(t, l^+) = y(l^+).
\end{aligned}$$

As for any $e \in T_1 \cup P(s, l^+)$, $f(e^1) \leq c(e)$, the capacity requirement is met. Also, by the definition on f, the flow satisfies

$$f(e^1) + f(e^2) = \sum\{y(l) \mid P(v(l_s), l) \ni e,\ l \in L(T_1)\}, \quad e \in T_1,$$

$$f(e^1) + f(e^2) = y(l^+), \quad e \in P(s, l^+),$$

from which it is easy to see that f is a feasible flow on N, and the total cost of the flow f is equal to the minimum objective value of (13).

Hence, we conclude that solving (13) is equivalent to finding a minimum cost flow from $v(l_s)$ to s on N. ▮

Notice that N belongs to the class of two-terminal series parallel graphs. Applying Booth and Tarjan's algorithm[6], the minimum cost flow problem can be solved in $O(|V^+| \log |V^+|) = O(|V| \log |V|)$ times.

Now we discuss how to recover the optimal solution of $LP(l^+)$ from an optimal solution of (13).

Let (y^*, z^*) be the optimal solution of (13). We claim that $y^*(l^+) > 0$ first. Otherwise, we can deduce that $y^*(l) = 0$ for all $l \in L(T_1)$, and $z^*(e) = 0$ for all $e \in T_1 \cup P(s, l^+)$. This means that the optimal value of (13) (hence $LP(l^+)$) is 0, and thus, we have $d_w(v(l_s), l_s) \leq d_w(s, l^+)$, which means that s is already a center, contradicting the assumption that s is not a center.

By the complementary slackness theorem of linear programming, (R^*, x^*) is an optimal solution of (11) if and only if

$$-R^* + \sum_{e \in P(s, l^+)} x^*(e) = -L(l^+), \tag{14}$$

$$R^* + \sum_{e \in P(v(l_s), l)} x^*(e) = L(l), \quad \forall l \in L(T_1) \quad \text{and} \quad y^*(l) > 0, \tag{15}$$

$$R^* + \sum_{e \in P(v(l_s), l)} x^*(e) \geq L(l), \quad \forall l \in L(T_1) \quad \text{and} \quad y^*(l) = 0, \tag{16}$$

$$0 \leq x^*(e) \leq b(e), \qquad \forall e \in T_1 \cup P(s, l^+), \tag{17}$$

$$x^*(e) = 0, \qquad \forall e \in T_1 \quad \text{and} \quad \sum_{P(v(l_s), l) \ni e} y^*(l) - z^*(e) < c(e), \tag{18}$$

$$x^*(e) = 0, \qquad \forall e \in P(s, l^+) \quad \text{and} \quad y^*(l^+) - z^*(e) < c(e), \tag{19}$$

$$x^*(e) = b(e), \qquad \forall e \in T_1 \cup P(s, l^+) \quad \text{and} \quad z^*(e) > 0. \tag{20}$$

Now we consider how to solve the above inequality system. First, we process the edges satisfying (18), (19), and (20). If e satisfies (18) and (19), set $x^*(e) = 0$; if e satisfies (20), set $x^*(e) = b(e)$. Then we contract these edges and modify $L(l^+)$ and $L(l)$ for each $l \in L(T_1)$ accordingly. For instance, update $L(l^+)$ by $L(l^+) + \sum\{b(e) \mid z^*(e) > 0, \text{and}\, e \in P(s, l^+)\}$, and update $L(l)$ by $L(l) - \sum\{b(e) \mid z^*(e) > 0, \text{and}\, e \in P(v(l_s), l)\}$ for all $l \in L(T_1)$.

Second, denote $Q'' = \{l \in L(T_1) \mid y^*(l) > 0\}$ and $T_1^* = \{e \in T_1 \mid \text{there exists } l \in Q'' \text{ such that } P(v(l_s), l) \ni e\}$.

For each $e \in T_1 \setminus T_1^*$, it is clear that e belongs to only one $P(v(l_s), l)$ for some $l \in L(T_1)$ with $y^*(l) = 0$. We may set $x^*(e) = b(e)$ to guarantee (16). Hence, without loss of generality, we can assume $T_1 = T_1^*$.

After the above two steps, the inequality system has been transformed into a reduced one as follows:

$$-R^* + \sum_{e \in P(s,l^+)} x^*(e) = -L(l^+), \tag{21}$$

$$R^* + \sum_{e \in P(v(l_s),l)} x^*(e) = L(l), \qquad \forall l \in L(T_1), \tag{22}$$

$$0 \le x^*(e) \le b(e), \qquad \forall e \in T_1 \cup P(s, l^+), \tag{23}$$

where T_1 and $P(s, l^+)$ are modified by deleting some edges in the above two steps.

Combining (21) and (22), we obtain

$$\sum_{e \in P(s,l^+)} x^*(e) + \sum_{e \in P(v(l_s),l)} x^*(e) = L(l) - L(l^+), \qquad \forall l \in L(T_1). \tag{24}$$

Therefore, we have $L(l) \ge L(l^+)$ for all $l \in L(T_1)$, and $\sum_{e \in P(s,l^+)} x^*(e) \le \min\{L(l) - L(l^+) \mid l \in L(T_1)\} := H$.

If $H \ge \sum_{e \in P(s,l^+)} b(e)$, let $x^*(e) = b(e)$ for $e \in P(s, l^+)$, and $R^* = L(l^+) + \sum_{e \in P(s,l^+)} b(e)$. Otherwise from l^+ to s, edge by edge, set $x^*(e) = \min\{b(e), H\}$, and update H by $H - x^*(e)$, until H becomes zero. Let $R^* = L(l^+) + H$.

By the definition of H, it is not difficult to see that $\sum_{e \in P(v(l_s),l)} b(e) \ge L(l) - R^*$ if the system of (21)–(23) has a feasible solution.

The remaining unknowns are $\{x^*(e) \mid e \in T_1\}$ satisfying

$$\sum_{e \in P(v(l_s),l)} x^*(e) = L(l) - R^*, \qquad \forall l \in L(T_1), \tag{25}$$

$$0 \le x^*(e) \le b(e), \qquad \forall e \in T_1. \tag{26}$$

Sort all vertices in $L(T_1)$ in the nondecreasing order of $L(l)$. Define a potential function as follows: $p(v(l_s)) = 0$, and scan each $l \in L(T_1)$ by the just determined order. Suppose that v' is the last vertex which has already been assigned a potential $p(v')$ on $P(v(l_s), l)$. Let $p(v) = \min\{p(u) + b([u, v]), L(l) - R^*\}$ for the successive vertices from v' to l, where u is the vertex preceding v.

It is clear that $p(l) = L(l) - R^*$ for all $l \in L(T_1)$. Let $x^*([u, v]) = p(v) - p(u)$ for all $[u, v] \in T_1$. It is straightforward to check that $\{x^*(e) \mid e \in T_1\}$ satisfies (25) and (26).

Since the dominating computation in the recovering procedure is to sort vertices in $L(T_1)$, the computational complexity of recovering procedure is $O(|V| \log |V|)$. Note that finding a minimum cost flow from $v(l_s)$ to s on N can be done in $O(|V| \log |V|)$ time. So, we conclude that solving each $LP(l^+)$ can be done in $O(|V| \log |V|)$ time.

Since the number of problems $LP(l^+)$ is $|L(T_2)| < |V|$, we obtain the following theorem.

Theorem 3 *The inverse center location problem on a tree can be solved in* $O(|V|^2 \log |V|)$ *time.*

Remark 1 If T_1 is a path, we can solve $LP(l^+)$ directly. In fact, we can sort the edges in $P(v(l_s), l_s) \cup P(s, l^+)$ in the nondecreasing order of $c(e)$. Let $H_1 = d_w(v(l_s), l_s)$, and $H_2 = d_w(s, l^+)$. Starting from the first edge, if the current edge $e \in P(v(l_s), l_s)$, let $w^*(e) = w(e) - \min\{b(e), H_1 - H_2\}$ and update H_1 by $H_1 - \min\{b(e), H_1 - H_2\}$, otherwise $w^*(e) = w(e) + \min\{b(e), H_1 - H_2\}$ and update H_2 by $H_2 + \min\{b(e), H_1 - H_2\}$.

4 Unit Modification Cost Case

In this section, we consider a simple case that the unit modification costs are identical. Without loss of generality, we assume that $c(e) \equiv 1$.

Since $P(s, l^+)$ is a path, T_1 is a tree, and $c \equiv 1$, it is cheaper to increase the weights of edges on $P(s, l^+)$ first than to decrease the weights of edges on T_1 in order to solve $L(l^+)$. Hence, if $\sum\limits_{e \in P(s,l^+)} (w(e) + b(e)) \geq d_w(v(l_s), l_s)$, we only need to increase the length of $P(s, l^+)$ to $d_w(v(l_s), l_s)$, and $L(l^+)$ is solved.

If $\sum\limits_{e \in P(s,l^+)} (w(e) + b(e)) < d_w(v(l_s), l_s)$, we set $w^*(e) = w(e) + b(e)$ for $e \in P(s, l^+)$, and go further to reduce the weights of edges on T_1 such that $d_{w^*}(v(l_s), l) \leq d_{w^*}(s, l^*) := L^*$.

To this end, it is intuitive that shortening an arc close to $v(l_s)$ is better than shortening an arc which is far from $v(l_s)$ on T_1. Therefore, we should decrease a close arc as much as possible when a reduction is needed.

Let $\Delta(l) = \max\{d_w(v(l_s), l) - L^*, 0\}$ for all $l \in L(T_1)$. For each $v \in V(T_1) \setminus (L(T_1) \cup \{v(l_s)\})$, there is only one arc $[u, v]$ adjacent to v. Let $\Delta(v) = \max\{\Delta(l) \mid P(v(l_s), l) \ni [u, v],\ l \in L(T_1)\}$. Note that $\Delta(v)$ is the largest required reduction of the paths $P(v(l_s), l)$ passing by v. We can compute all $\Delta(v)$ in linear time by assigning $\Delta(v) = \max\{\Delta(v') \mid [v, v'] \in T_1\}$ from leaves in $L(T_1)$ to the root $v(l_s)$.

Denote by $\delta(v)$ the accumulating reduction on the path from $v(l_s)$ to v, and set $\delta(v(l_s)) = 0$. Then from $v(l_s)$ to leaves, for each arc $[u, v] \in T_1$, we set $\Delta(v) = \max\{\Delta(v) - \delta(u), 0\}$, $w^*([u, v]) = w([u, v]) - \min\{\Delta(v), b([u, v])\}$, and $\delta(v) = \delta(u) + \min\{\Delta(v), b([u, v])\}$. Obviously, the computation of w^* and δ runs in the linear time, too. Therefore, we obtain the following assertion.

Assertion 6 *If $c \equiv 1$, $LP(l^+)$ can be solved in linear time.*

Moreover, by the above analysis, we can also obtain the following properties:

- Let $w'(e) = w(e) + b(e)$ for $e \in T_2$. If $Q := \{l \in L(T_2) \mid d_{w'}(s, l) \geq d_w(v(l_s), l_s)\} \neq \emptyset$, we only need to find the longest path from s to Q, and extend this path to $d_w(v(l_s), l_s)$.
- If $Q = \emptyset$, let $Q' := \{l \in L(T_2) \mid d_{w'}(s, l) = \max\{d_{w'}(s, l) \mid l \in L(T_2)\}\}$ and let $l^+ = \arg\max\{d_w(s, l) \mid l \in Q'\}$. Then $L(l^+)$ solves the inverse center location problem.

Based on the above two properties, we don't need to solve each $LP(l^+)$ individually. We only need to determine Q or Q' (in case $Q = \emptyset$), and find the longest path from s to Q (or Q'). Thus, we need to solve at most one $LP(l^+)$. Hence, we get the following theorem.

Theorem 4 *When modification costs for every edge are equal, the inverse center location problem can be solved in $O(|V|)$ time.*

5 Concluding Remarks

In the above discussion, we assume that weights of edges cannot be reduced to zero. If we relax this restriction by allowing weights to be reduced to zero if necessary, we need more computation to find the optimal solution. The inverse center location problem with different unit modification costs can be solved in $O(|V|^3 \log(|V|))$ time, while the inverse center location problem with equal unit modification costs can be solved in $O(|V|^2)$ time.

A vertex in a network is called a general center if the distance from the vertex to the remotest point (not only among vertices, but also among all points in the edges) in the network is minimum. When the network is a tree, a center is exactly a general center, and vice versa. Therefore, an inverse general center location problem on a tree is exactly an inverse center location problem.

References

[1] N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.

[2] J. R. Evans and E. Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, Inc., New York, 1992.

[3] C. Heuburger, Inverse optimization, a survey on problems, methods, and results, *Journal of Combinatorial Optimization*, 2004, **8**(3): 329–361.

[4] M. Cai, X. Yang, and J. Zhang, The complexity analysis of the inverse center location problem, *Journal of Global Optimization*, 1999, **15**(4), 213–218.

[5] E. Tardos, A strongly polynomial algorithm to solve combinatorial linear programs, *Operations Research*, 1986, **34**(2): 250–256.

[6] H. Booth and R. E. Tarjan, Finding the minimum-cost maximum flow in a serial-parallel network, *Journal of Algorithms*, 1993, **15**(3): 416–446.