Reverse 1-median Problem with Constraint in Trees

Yang Jianfang
Institute of Operational Research and
Cybemetics, Hangzhou Dianzi
University, Hangzhou 310018, China
jianfangyang@126.com

Jiang Juan
Institute of computer application,
Hangzhou DianZi university,
Hangzhou 310018,China
jiangjuan@hdu.edu.cn

***Abstract*-Different from classical location problem, the reverse problem is how to improve the network as efficient as possible within a given budget when the facilities have already been located in a network and cannot be moved to a new place. This paper concerns the reverse 1-meidan problem with constraint in tree network. It is shown that the model can be divided into two equivalent subproblems. The subproblems will be solved respectively by minimum cut algorithm and greedy algorithm. Finally, an example is set to verify the feasibility of these algorithms.**

KeyWords-Facility location, Tree network, Reverse problem, Minimum cut, Greedy algorithm

I. INTRODUCTION

Location theory in network deals with the problem of finding locations of facilities in a network. As we know, the purpose of classical location problems is to find the "best" position for a facility in a network such that the distance from the facility to the farthest vertices (or all vertices) of the network is minimized. However, it is also possible to meet a reverse case in practice. The facilities may already exist and cannot be moved to a new place. Instead of finding optimal locations the goal is to improve the given locations as much as possible by modifying certain parameters within a given budget. This kind of improvement problem is called reverse problem.

Usually MinSum or MinMax is adopted as objectives for measuring the efficiency of facility locations, which is named respectively median problem and center problem. Reverse median and reverse center problems have already been investigated by Berman et al. [1,2]. Since then several authors studied reverse problems with considerable interest. The reverse 1-median problem as well as the reverse 1-center problem in general network are known to be NP-hard [2,3]. Therefore, some special networks have been studied. Zhang et al. [4] considered the reverse center location problem in a tree network where all vertices have equal weight. Burkar et al. gave a linear time algorithm for the reverse 1-median problem on a cycle with linear cost functions[3]. In 2008, Burkar et al. also concerned the reverse 2-median problem in trees and the reverse 1-median problem in graphs that contain exactly one cycle[5].

There is another concept, inverse problem, which is strongly related to reverse problem. The goal of an inverse location problem is to modify the parameters (lengths, cost coefficients, weights, etc) at minimum cost so that a given solution can become an optimal solution. There are many papers discussing inverse optimization problems [6-10].

In this paper, we consider a network improvement problem which is to modify the lengths of edges in a network so that after modification, the distances between a facility vertex and other vertices do not exceed a given upper bound and the total modification cost is minimum.

This article is organized as follows: In Section 2 the reverse 1-median problem with constraint in the tree network is introduced. In Section 3, we prove that the problem in trees can be transformed into two subproblems: reverse 1-median problem and reverse 1-center problem. Furthermore, a solution method which leads a polynomial algorithm is suggested. Based on the algorithm of Section 3, an example is illustrated in Section 4.

II. PROBLEMF FORMULATION

In this article, we investigate reverse 1-median problem with constrains in tree networks. Let $T=(V,E,s;h,l,a,w;B,p)$ be a tree network with a set of vertices V, $|V|=n$ and a set of edges E, $|E|=m$. Suppose there is one facility in T that is located at a given point s. Associated with each vertex i ,let $h_i(h_i \geq 0)$ denote the demand weight of vertex i ($i=1,2,\ldots,n$). l,a,w are functions from E to R^+. For each edge $(i,j)\in E$, l_{ij} is its current length, a_{ij} is its minimum permissible length, and w_{ij} is the cost of reducing the length by one unit. Furthermore, a budge denoted by $B>0$ and the length constraint denoted by $p>0$ is known. The task is to use the budget to change the length of some edges such that the over-all sum of the weighted distances to the prespecified vertex s becomes as small as possible, at the same time the length of the shortest path from v_i to s must be no more then the constant p. Using the notation introduced above, the problem can formally be stated as

$$\begin{aligned} (P)\quad & \min \sum_{i=1}^{n} h_i d(v_i,s) \\ \text{s.t.}\quad & \max_{i\in V} d(v_i,s) \leq p \\ & \sum_{(i,j)\in E} w_{ij}x_{ij} \leq B \\ & 0 \leq x_{ij} \leq l_{ij} - a_{ij},\ (i,j)\in E \qquad (1) \end{aligned}$$

Where x_{ij} represents the length to be shortened on edge (i,j) and $d(v_i,s)$ is the length of shortest path from v_i to s in the resulting network.

Published by Atlantis Press, Paris, France.

In order to simplify the model (P) , we will deal with it as [4]. We may assume that all $a_{ij}=0$, because if $a_{ij}>0$ for some edge $(i,j)\in E$, then the edge (i,j) can be replaced by two edge and $(i^{'},j)$, and define $l_{ii^{'}}=l_{ij}-a_{ij}$, $l_{i^{'}j}=a_{ij}$, $w_{ii^{'}}=w_{ij}$ and $w_{i^{'}j}=\infty$. Therefore (P) becomes the following form:

$$
\begin{aligned}
&(P^{'}) \quad \min \sum_{i=1}^{n} h_i d(v_i,s)\\
&\text{s.t.} \quad \max_{i\in V} d(v_i,s)\le p\\
&\sum_{(i,j)\in E} w_{ij}x_{ij} \le B\\
&\quad 0\le x_{ij}\le l_{ij},\ (i,j)\in E \qquad (2)
\end{aligned}
$$

and the corresponding network is denoted by $T=(V,E,s;h,l,w;B,p)$.

III. ALGORITHM ANALYSIS

The model can be divided into two parts: the first part is to control the longest distance from all verctices to the facility s , which meas $d(v_i,s)$ must be no more then an upper bound p for every vertex i on the modified network. The aim is to balance the benefit of all customers. In this problem we want to change the edge lengths at minimun cost. The second part is to optimize the minisum obejective value. The task is to use surplus budget in order to change the length of some edges such that the overall sum of the weighted distance of the vertice to the prespecified vertex becomes as small as possible.

Therefore, the question ($P^{'}$) would be divided into two subproblems, which be stated as:

(a) We want to change the edge lengths at minimun cost so that the length of the longest path from the place of location s to each vertex must not exceed an upper bound p . The problem can formally be stated as follows:

$$
\begin{aligned}
&(P_1) \quad \min \sum_{(i,j)\in E} w_{ij}x_{ij}\\
&\text{s.t.} \quad \max_{1\le i\le n} d(v_i,s)\le p\\
&\quad 0\le x_{ij}\le l_{ij},\ (i,j)\in E \qquad (3)
\end{aligned}
$$

Where x_{ij} represents the length to be shortened on edge (i,j) in the problem(P_1).

(b) We suppose the objective function value of the problem (P_1) is $B^{'}$ ($B^{'}=\min\sum_{(i,j)\in E} w_{ij}x_{ij}$). So the next step is to modify the edge lengths within the rest of the budget $B-B^{'}$. The model can formally be written as follows:

$$
\begin{aligned}
&(P_2) \quad \min \sum_{i=1}^{n} h_i d(v_i,s)\\
&\text{s.t.} \quad \sum_{(i,j)\in E} w_{ij}x_{ij}^{'} \le B-B^{'}\\
&\quad 0\le x_{ij}^{'}\le l_{ij}^{'},\ (i,j)\in E \qquad (4)
\end{aligned}
$$

Where $x_{ij}^{'}$ represents the length to be shortened again on edge (i,j) in the problem(P_2), and $l_{ij}^{'}=l_{ij}-x_{ij}$ represents the length of edge (i,j) after the first clip in the problem (P_1).

Lemma Every feasible solution of the problem ($P^{'}$) is feasible solution of the problem (P_1) and (P_2), and vice versa.

Proof. Suppose $x_{ij}^{(1)}$ ($(i,j)\in E$)is a feasible solution of the problem (P_1), $B^{(1)}$ is objective function value of the solution $x_{ij}^{(1)}$, $x_{ij}^{(2)}$ is a feasible solution of the problem (P_2), x_{ij} can be defined by $x_{ij}=x_{ij}^{(1)}+x_{ij}^{(2)}$.

$$
\begin{aligned}
\sum_{(i,j)\in E} w_{ij}x_{ij} &= \sum_{(i,j)\in E} w_{ij}(x_{ij}^{(1)}+x_{ij}^{(2)})\\
&= \sum_{(i,j)\in E} w_{ij}x_{ij}^{(1)} + \sum_{(i,j)} w_{ij}x_{ij}^{(2)}\\
&\le B^{(1)}+(B-B^{(1)})=B
\end{aligned}
$$

Because $x_{ij}^{(1)}$ is a feasible solution of the problem (P_1), we get an inequality $\max_{1\le i\le n} d(v_i,s)\le p$ after the first change. And because $x_{ij}^{(1)}+x_{ij}^{(2)}\ge x_{ij}^{(1)}$ is correct, the inequality $\max_{1\le i\le n} d(v_i,s)\le p$ is also satified in the problem ($P^{'}$).

Suppose x_{ij} ($(i,j)\in E$)is a feasible solution of the problem ($P^{'}$), we define $x_{ij}^{(1)}=x_{ij}$ and $x_{ij}^{(2)}=0$. Obviously the solutions $x_{ij}^{(1)}$, $x_{ij}^{(2)}$ defined as above is respctively a feasible solution of the problem (P_1) and (P_2). □

It follows directly frome above lemma, we can get:

Theorem 1. The problem ($P^{'}$) is equivalent to the problem (P_1) and (P_2).

Therefore we can solve the problem ($P^{'}$) by converting it into two subproblems problem (P_1) and (P_2).

For the problem (P_1), the maximum flow of minimum cut algorithm is proposed as [4]. Let V_1 be the set of all end vertices of V .We'll calculate the length of the shortest path from s to vertex i in the network, denoted as $d(i)$ ($i\in V_1$). Define

$d=\max\{d(i)\mid i\in V_1\}$.

Introduce a new vertex t and define

$V^{*}=V\cup\{t\}$, $E_1=\{(i,t)\mid i\in V_1\}$, $E^{*}=E\cup E_1$;

$$
\begin{cases}
l_{ij}^{*}=l_{ij} & \mathit{for}\ \ (i,j)\in E\\
l_{it}^{*}=d-d(i) & \mathit{for}\ \ (i,t)\in E_1.
\end{cases}
$$

$$
\begin{cases}
w_{ij}^{*}=w_{ij} & \mathit{for}\ \ (i,j)\in E\\
w_{it}^{*}=\infty & \mathit{for}\ (i,t)\in E_1\ \mathit{and}\ d-d(i)=0\\
w_{it}^{*}=0 & \mathit{for}\ (i,t)\in E_1\ \mathit{and}\ d-d(i)>0.
\end{cases}
$$

We can transform the network T into the new network $T^{*}=(V^{*},E^{*},s;h,l^{*},w^{*};B,p;d)$. An algorithm would be presented as follow.

Algorithm 1.

Step 1: Put $k\leftarrow 0, T^{k}\leftarrow T^{*}, l_{ij}^{k}\leftarrow l_{ij}^{*}, w_{ij}^{k}\leftarrow w_{ij}^{*}, d^{k}\leftarrow d$;

Step 2: Regard w_{ij}^{k} as the capacity of edge (i,j) . Let R^{k} be the minimum $s-t$ cut set of T^{k} , which can be found by algorithm for maximum flow problems in a $s-t$ planar network. The capacity of R^{k} (cost) is denoted by $w^{k}(R^{k})=\sum_{(i,j)\in R^{k}} w_{ij}^{k}$.

Step 3: If $w^{k}(R^{k})=\infty$ and $d^{k}\le p$, which means the constraint condition is established. Stop, an optimum solution has been obtained.

Otherwise, calculate

$\delta_1^{k}=d^{k}-p, \delta_2^{k}=\min\{l_{ij}^{k}\,\big|\,(i,j)\in R^{k}\}$, and decide $\delta^{k}=\min\{\delta_1^{k},\delta_2^{k}\}$.

Step 4: Let

$$
l_{ij}^{k+1}=\begin{cases}
l_{ij}^{k}-\delta^{k}, & \mathit{if}\ (i,j)\in R^{k},\\
l_{ij}^{k}, & \mathit{otherwise}.
\end{cases}
\quad \text{and put } d^{k+1}\leftarrow d^{k}-\delta^{k}.
$$

Step 5: If $\delta^{k}=\delta_1^{k}$, stop, the constraint condition is satisfied and an optimum solution has been obtained. Otherwise we define

$$w_{ij}^{k+1} = \begin{cases} \infty, & if\ l_{ij}^{k+1} = 0, \\ w_{ij}^{*}, & otherwise. \end{cases}$$

Let $T^{k+1} = (V^{*}, E^{*}, s; h, l^{k+1}, w^{k+1}; B, p; d^{k+1})$ and put $k \leftarrow k+1$, return to step 2.

Remark 1. At step 2, the condition $w^{k}(R^{k}) = \infty$ means there is at least one edge with the infinite cost of reducing the length by one unit, which implies we can not do any improvement. Therefore if $d^{k} > p$ is established, this problem is not the feasible solution.

For the problem (P_2), the greedy alogrithm would be used within the rest of the budget $B - B^{'}$. In order to solve this problem, there are several definitions and theorems must be introduced[1].

Considering any edge $(i,j) \in E$, a cut of (i,j) partitions the tree network $T = (V, E)$ into two subtrees T_i and T_j, then let s be in T_i. Suppose we reduce the length of (i,j) by x_{ij}. Because distances to vertices in T_i will not be affected, the total improvement of the objective function value $Z_m = \sum_{i=1}^{n} h_i d(v_i, s)$ is $I_{ij} = x_{ij} \sum_{k \in T_j} h_k = x_{ij} H_j$. We call I_{ij} the "marginal contribution" of edge (i,j) to Z_m. We can transform the problem (P_2) into a maximun problem as follow:

$$\begin{aligned} \max \quad & I = \sum_{(i,j)\in E} x_{ij} H_j \\ \text{s.t.} \quad & \sum_{(i,j)\in E} w_{ij} x_{ij} \le B - B^{'} \\ & 0 \le x_{ij}^{'} \le l_{ij}^{'},\ (i,j) \in E \end{aligned} \tag{5}$$

Theorem 2[1]. The edges that have the largest marginal contribution to Z_m are incident to s.

We will first consider to prune the edge adjacent to s by theorem 2.

For each edge (i,j), we define $r_{ij} = H_j / w_{ij}$. The corresponding network is denoted by $T = (V, E, s; h, l^{'}, w; B - B^{'})$ after the problem(P_1) had been solved. The following algorithm would be used to solve the problem similar to[1].

Algorithm 2.

Step 1: Put $k \leftarrow 0, T^{k} \leftarrow T, B^{k} \leftarrow B - B^{'}, r_{ij}^{k} \leftarrow r_{ij}$.

Step 2: Calculate r_{ij}^{k} for each edge (i,j) which is incident to s and order the different r_{ij}^{k} in no increasing order and choose the edge with the largest r_{ij}^{k} denoted as (t,s).

Step 3: For the selected edge (t,s), define:

$$x_{ts}^{k} = \begin{cases} l_{ts}^{'} & if \quad w_{ts} l_{ts}^{'} < B^{k} \\ B^{k} / w_{ts} & if \quad w_{ts} l_{ts}^{'} \ge B^{k}. \end{cases}$$

$$B^{k+1} = \begin{cases} B^{k} - w_{ts} l_{ts}^{'} & if \quad w_{ts} l_{ts}^{'} < B^{k} \\ 0 & if \quad w_{ts} l_{ts}^{'} \ge B^{k}. \end{cases}$$

Step 4: If $B^{k+1} = 0$, stop, otherwise delect the edge (t,s) from E, let $\{s\} = \{s\} \cup \{t\}$ and let $T^{k+1} = (V, E, s; h, l^{k+1}, w^{k+1}; B^{k+1}, p;)$ and put $k \leftarrow k+1$, return to step 2.

Remark 2. At step 4, when the condition $B^{k+1} = 0$ is satisfied, the Algorithm 2 should be stopped. Vice versa, when the Algorithm 2 is stopped, the budget is perhaps not equal to zero. Because in some special cases, the network should not be improved further.

We can draw a conclusion that the reverse 1-median problem with constraints in tree network would be solved in polynomial time as analysis of [1,4].

IV. AN EXAMPLE

To verify the above algorithms, we give an example. A given tree network T is shown in Figure.1, where the vertex s is the given location of the facility, and the pairs of numbers beside each edge (i,j) represent (l_{ij}, w_{ij}). Let $B = 17$, $p = 9$, $h = \{2,3,5,2,3,6,2,1,2,2,6,3\}$.

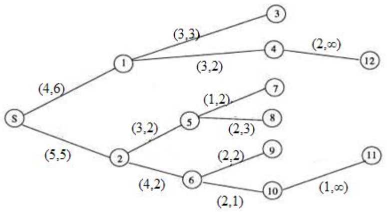

Figure 1. A tree network

Using the above algorithms, we can get the optimal solution $x_{2,5}^{*} = x_{6,10}^{*} = 1$; $x_{2,6}^{*} = x_{2,s}^{*} = 2$, $x_{ij} = 0$ for any other edge (i,j). The improved tree network is shown in Figure. 2.

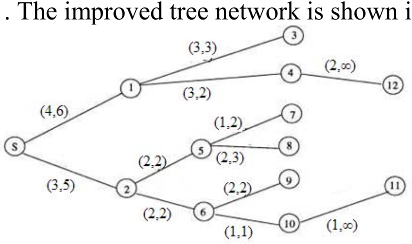

Figure 2. The improved tree network

V. SUMMARIES

In the previous sections, we considered the reverse 1-median problem with constraints in tree network. First, the model could be divided into two subproblems:reverse 1-meidian problem and reverse 1-center problem without constraint. Then we outlined a polynomial algorithm for the problem. Last an example illustrated the effectiveness and feasibility of the model and algorithm.

Since the reverse problem is NP-hard on general graphs even without constraints, different other models on some special graph should be investigated. This is an interesting task to be addressed in future.

ACKNOWLEDGMENT

National Natural Science Foundation of China（11071219）National Natural Science Foundation of China, Tian Yuan Special Foundation (.11026107); Zhejiang Natural Science Foundation (Y6090080，Y1090465);

Supported by Foundation of ZheJiang Educational Committee（Y201016901）.

REFERENCES

[1] Berman O; Ingco D I; Odoni A R. Improving the location of minsum facilities through network modification. Annals of Operations Research[J],1992,40,PP: 1-16.

[2] Berman O; Ingco D I; Odoni A R. Improving the location of minmax facilities through network modification. Networks[J], 1994, 24,PP: 31-41.

[3] Burkard R E; Gassner E; Hatzl J. A linear time algorithm for the reverse 1-median problem on a cycle. Networks[J], 2006, 48,PP:16-23.

[4] Zhang J Z; Liu Z H; Ma Z F. Some reverse location problems. European Journal of Operational Research[J], 2000,124,PP: 77–88.

[5] Burkard R E; Gassner E; Hatzl J. Reverse 2-median problem on trees. Discrete Applied Mathematics[J],2008, 156,PP: 1963- 1976.

[6] Cai M C; Yang X G; Zhang J. The complexity analysis of the inverse center location problem. J. Global Opt[J], 1999,15,PP:213–218.

[7] Yang C; Zhang J. Two general methods for inverse optimization problems. Applied Mathematics Letters[J], 1999,12,PP: 69-72.

[8] Burkarda R E; Carmen P; Zhang J Z. Inverse median problems. Discrete Optimization[J],2004,1,PP:23 – 39.

[9] Heuberger C. Inverse optimization: A survey on problems, methods, and results. Journal of Combinatorial Optimization[J],2004,8:329–361.

[10] Burkarda R E; Carmen P; Zhang J Z. The inverse 1-median problem on a cycle. Discrete Optimization[J], 2008,5,PP:242–253.