Theoretical Computer Science 607 (2015) 296–305

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Two paths location of a tree with positive or negative weights

Jianjie Zhou [a], Liying Kang [a], Erfang Shan [b,*,1]

[a] Department of Mathematics, Shanghai University, Shanghai 200444, P.R. China
[b] School of Management, Shanghai University, Shanghai 200444, P.R. China

ARTICLE INFO

Article history:
Received 17 February 2015
Received in revised form 16 August 2015
Accepted 24 August 2015
Available online 29 August 2015

Keywords:
Facility location
Path
Tree
Semi-obnoxious

ABSTRACT

This paper studies two problems of locating two paths in a tree with positive and negative weights. The first problem has objective to minimize the sum of minimum weighted distances from every vertex of the tree to the two paths, while the second is to minimize the sum of the weighted minimum distances from every vertex of the tree to the two paths. We develop an $O(n^2)$ algorithm based on the optimal properties for the first problem, and also an $O(n^3)$ algorithm for the second problem.

1. Introduction

Network location problem is to find the optimal locations of service facilities in a network. The location problem usually has important applications in transportation and communication and thus has received much attention from researchers. According to the realistic demands, the shapes of the facilities can be points, paths, or trees. In the past, a variety of network location problems have been defined and studied in the literature [1,7–12,14,19].

The core of tree is defined as a path in the graph with minimum sum of the distances of all vertices of the graph from the path (see [14] for this definition). Morgan and Slater [14] studied a linear time algorithm for finding the core of a tree. Becker and Perl [2] presented two algorithms for the problem of finding the two-core of a tree. The first requires $O(n^2)$ time and the second requires $O(dn)$ time, where d is the maximum number of edges of any simple path in the tree. Wang [21] reduced the computational complexity of this problem to linear time.

Sometimes facilities may be temporality unavailable to provide service to customers due to unknown or unpredictable situations. Perto et al. [16] considered the reliability problems in multiple path-shaped facility location on networks. They gave an $O(n^2)$ time complexity algorithm when the graph is a tree and also proved that the problem is NP-hard even when the graph is a cactus. In some cases, there is a bound on the total length of the located paths. Becker et al. [4] considered the problem of locating central-median paths with bounded length on trees, they presented two problems and gave an $O(n \log^2 n)$ divide-and-conquer algorithm [4]. It was shown that the problem of locating a median path of length at most l is NP-complete on the simple classes of cactus and grid graphs [3,13,18].

In reality, due to the complexity of the world, some vertices are desirable and others are undesirable, the problem is referred to as the semi-obnoxious location problem. To model this problem, Burkard et al. [6] considered 2-medians problems in trees with positive or negative (for simplicity we write pos/neg) weights, they formulated two objective functions and

* Corresponding author.
E-mail address: efshan@shu.edu.cn (E. Shan).

1 This research was partially supported by the National Natural Science Foundation of China (Nos. 11471210 and 11171207).

http://dx.doi.org/10.1016/j.tcs.2015.08.025
0304-3975/

gave the corresponding algorithms. Zaferanieh and Fathali [22] presented that the core of a tree with pos/neg weights can be found in linear time.

In this paper, we consider the problem of locating two paths of a tree with pos/neg weights. We formulate two different objective functions: (1) the sum of the minimum weighted distances of the type vertex-path over all vertices; (2) the sum of the weighted minimum distances of the type vertex-path over all vertices. In Section 2, we give a formal formulation of two problems. Section 3 analyzes the optimal properties of the first problem and gives an $O(n^2)$ algorithm. In Section 4, we show that the second problem has some optimal properties and present an $O(n^3)$ algorithm for the problem. Last section gives some conclusions.

2. Problem formulation

Let $T=(V,E)$ be a tree with $|V|=n$. Each edge $e=(v_i,v_j)$ of T has a positive length $l(e)=l(v_i,v_j)$ and each vertex $v_i\in V$ has a real wight $w(v_i)$. When the tree is rooted at r it is denoted by T_r. For any vertex v_i, let T_{v_i} be the subtree of T_r rooted at vertex v_i, $S(v_i)$ the set of the children of v_i in T_r and $p(v_i)$ the parent of v_i in T_r.

Let $d(v,u)$ be the shortest distance between points v and u, where "points" can be vertices or points on an edge. The length of the shortest path between vertex v and path P is $d(v,P)$ defined as $d(v,P)=\min\limits_{u\in P} d(u,v)$. A path is discrete if both its endpoints are vertices of T, otherwise it is continuous.

We consider the location of two path shaped facilities P_1, P_2. For a given location $P=\{P_1,P_2\}$, the sum of the minimum weighted distances and the sum of weighted minimum distances are given as follows:

$$F_1(P_1,P_2)=\sum_{i=1}^{n}\min_{j=1,2} w(v_i)d(v_i,P_j), \tag{2.1}$$

$$F_2(P_1,P_2)=\sum_{i=1}^{n} w(v_i)\min_{j=1,2} d(v_i,P_j). \tag{2.2}$$

The corresponding two optimization problems are:

(L_1) Find a location of two path P_1 and P_2 in T such that (2.1) is minimized.
(L_2) Find a location of two path P_1 and P_2 in T such that (2.2) is minimized.

Both problems L_1 and L_2 can be considered as extension of the pos/neg core problem defined in [22]. A solution $P^*=\{P_1,P_2\}$ is called a *minimal optimal solution* of problem L_1 (resp. L_2) if P^* minimizes (2.1) (resp. (2.2)) and there is no sub-path $P'_1\subseteq P_1$ and $P'_2\subset P_2$ or $P'_1\subset P_1$ and $P'_2\subseteq P_2$ such that $P'=\{P'_1,P'_2\}$ has the same objective function value as P^*. There is a tie when the two paths P_1,P_2 are equal distance from vertex v_i. In this case we have the following assumption: in problem L_1, if $w(v_i)\ge 0$, it is served by P_1, otherwise it is served by P_2; in problem L_2, it is served by P_1. For any $v\in V(T)$, if v is served by P_i ($i=1$ or $i=2$) and there exists a point x in P_i such that $d(v,P_i)=d(v,x)$, we call v is *served* by x.

3. Properties and an $O(n^2)$ algorithm for the problem L_1

3.1. Optimal properties of problem L_1

Assume $P^*=\{P_1,P_2\}$ is a minimal optimal solution of problem L_1. It is easily seen that if the weight of vertex v is nonnegative, it is served by the closer path from vertex v; otherwise it is served by the farther path from vertex v.

Proposition 3.1. *(1) If $P_1\cap P_2=\emptyset$ and $x\in P_1$, $y\in P_2$ are the points such that $d(x,y)=d(P_1,P_2)$, then the vertices of T with negative weights are served by x or y.*
(2) If $P_1\cap P_2\neq\emptyset$ and x,y are the end points of $P_1\cap P_2$, then the vertices of T with negative weights are served by x, y or vertices in $V(P_1\cap P_2)$.

Proof. (1). For any $v\in V(T)$ with negative weight, if $d(v,P_1)=\min\{d(v,P_1),d(v,P_2)\}$, then v is served by P_2. Since $d(x,y)=d(P_1,P_2)$, $d(v,P_2)=d(v,y)$. Then v is served by y. If $d(v,P_2)=\min\{d(v,P_1),d(v,P_2)\}$, then v is served by P_1. Since $d(x,y)=d(P_1,P_2)$, $d(v,P_1)=d(v,x)$. Then v is served by x.

(2). The proof is similar to the proof of (1), we omit it. □

Proposition 3.2. *In a minimal optimal solution of problem L_1 the two optimal paths are always discrete.*

Proof. Let $P^*=\{P_1=P(v_i,x),P_2=P(v_s,v_t)\}$ be a minimal optimal solution for problem L_1 and assume that v_i, v_s and v_t are vertices of T, while the other endpoint x belongs to the interior of an edge (v_r,v_q), where $v_r\in V(P_1)$. Let V_x be set of