The Tree Center Problems and the Relationship with the Bottleneck Knapsack Problems

Akiyoshi SHIOURA * and Maiko SHIGENO*

Abstract

The tree center problems are to find a subtree minimizing the maximum distance from any vertex. This paper shows that these problems in a tree network are related to the bottleneck knapsack problems, and presents linear-time algorithms for the tree center problems by using the relation.

1 The Tree Center Problems

Let $T = (V, E)$ be a tree network with a vertex set $V = \{v_1, \cdots, v_n\}$ and an edge set $E = \{e_2, \cdots, e_n\}$, where an edge e_i is incident to the vertex v_i and on the path connecting v_i and v_1. We denote by N the index set $\{2, \cdots, n\}$ of edges. For each $i \in N$, the edge e_i has a positive length l_i and a positive weight w_i. In this paper, we assume that T is drawn in the Euclidean plane so that each edge e_i is a line segment with length l_i, and regard T as a closed and connected subset of points in the Euclidean plane. For two points p, q on an edge e_i, a *partial edge connecting p and q* is the set of points lying between p and q on e_i. The length of the partial edge is defined by the Euclidean distance l_{pq} of p and q, and the weight is $w_i(l_{pq}/l_i)$.

A subset S of T is called a *subtree* if it is closed and connected. A subtree S is decomposed into several partial edges such that the intersection of any pair of distinct partial edges is empty or a vertex. We call a subtree *discrete* when its boundary points are vertices of T.

In a tree network T, the *distance between two points* $p, q \in T$, denoted by $d(p, q)$, is the sum of the lengths of partial edges on the unique path connecting p and q. The *distance between a point p and a subtree S* is given by $d(p, S) = \min\{d(p, q) \mid q \in S\}$. For each subtree S, the *eccentricity* $\mathrm{ecc}(S)$ is the maximum distance from S to any vertex in T, i.e., $\mathrm{ecc}(S) = \max\{d(v_i, S) \mid v_i \in V\}$. The *size of a subtree* S is defined as the sum of the weights of partial edges in S and denoted by $\mathrm{size}(S)$.

*Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-okayama, Meguro-ku, Tokyo 152, Japan. shioura@is.titech.ac.jp, maiko@is.titech.ac.jp

This paper deals with two types of the *tree center problems*, which find a subtree in T minimizing the eccentricity under the size constraint. The continuous version is described as

$$\text{CTCP} \left| \begin{array}{ll} \text{Minimize} & \text{ecc}(S) \\ \text{subject to} & \text{size}(S) \leq W, \\ & S \text{ is a subtree of } T, \end{array} \right.$$

and the discrete version is

$$\text{DTCP} \left| \begin{array}{ll} \text{Minimize} & \text{ecc}(S) \\ \text{subject to} & \text{size}(S) \leq W, \\ & S \text{ is a discrete subtree of } T, \end{array} \right.$$

where $0 \leq W < \text{size}(T)$. We call optimal solutions of these problems the *continuous tree center* and *discrete tree center*, respectively. These tree center problems and their variations were discussed by many researchers. When $W = 0$, the tree center problems find the so-called *absolute center* and *vertex center*, for both of which $O(n)$-time algorithms were already presented [4, 5]. For the case $w_i = l_i$ $(i \in N)$, Minieka [7] and Hakimi, Schmeichel and Labbé [3] presented simple greedy algorithms for the continuous and discrete tree center problems, respectively, both of which run in $O(n^2)$ time by naive implementation.

Our main aim is to investigate the relationship between the tree center problems and the bottleneck knapsack problems. We shall formulate the tree center problems as linear and integer programming problems. These formulations give the relation that the tree center problems are the bottleneck knapsack problems with the "subtree" constraint. We show that the additional subtree constraint is satisfied by optimal solutions of the bottleneck knapsack problems. The linear-time solvability of the bottleneck knapsack problems enables us to find optimal subtrees in $O(n)$ time.

A closely related work was done by Tamir [8] for the tree median problem, which finds a subtree minimizing the sum of the distances from all vertices. He derived the relationships with the knapsack problems, and presented exact and approximation algorithms for the tree median problems.

For convenience of the description, assume that T is a rooted tree with root v_1. For any two vertices v_i, v_j, if the path connecting v_1 and v_j contains v_i, then v_i is an *ancestor* of v_j, and v_j is a *descendant* of v_i. The *parent* of v_i, denoted by $f(v_i)$, is the unique ancestor of v_i connected by an edge e_i. For any $i \in N$ and a real number r with $0 \leq r \leq 1$, (r, i) represents the point on e_i with the Euclidean distance $l_i r$ from $f(v_i)$.

2 Main Results

We first show that the continuous tree center problem (CTCP) is reducible to the continuous bottleneck knapsack problem.

Simple observation gives the following property.

Lemma 2.1 [7] *The continuous tree center contains the absolute center.*

Therefore, it is sufficient to consider continuous subtrees containing the absolute center. We assume without loss of generality that the absolute center of T is a vertex in T. (Otherwise, we augment the absolute center to V.) We also assume that v_1 is the absolute center. Any subtree containing v_1 can be represented by using a vector $(y_i)_{i\in N}$ with $0 \le y_i \le 1$, where each y_i is associated with the edge e_i. If there exists a partial edge of e_i connecting $f(v_i)$ and (r,i), then we set $y_i = r$, otherwise $y_i = 0$. Such a vector satisfies the condition:

$$\text{if } y_j > 0 \text{ then } y_i = 1 \qquad (j \in N, v_i = f(v_j)). \tag{1}$$

On the other hand, if $(y_i)_{i\in N}$ satisfies the condition (1), then the point set

$$\bigcup_{i\in N,\ y_i>0} \{(r,i) \mid 0 \le r \le y_i\} \cup \{v_1\}$$

induces a subtree containing v_1.

For each $i \in N$, we define b_i as

$$b_i = \max\{d(v_i, v_j) \mid v_j \text{ is a descendant of } v_i\}.$$

It satisfies the recursive equation:

$$b_i = \begin{cases} \max\{l_j + b_j \mid j \in N, f(v_j) = v_i\} & (\exists v_h \text{ s.t. } f(v_h) = v_i), \\ 0 & (\text{otherwise}). \end{cases}$$

The eccentricity is rewritten as follows.

Lemma 2.2 *Suppose that S is a subtree containing v_1 with $S \neq T$ and that $(y_i)_{i\in N}$ is a vector associated with S. Then, it holds that*

$$\mathrm{ecc}(S) = \max\{l_i(1-y_i) + b_i \mid i \in N, y_i < 1\}.$$

From the above discussion, we can restate the CTCP as

$$\text{CTCP1} \left| \begin{array}{lll} \text{Minimize} & \max\{l_i(1-y_i)+b_i \mid i\in N, y_i<1\} & \\ \text{subject to} & \sum\{w_i y_i \mid i \in N\} \le W, & \\ & \text{if } y_j > 0 \text{ then } y_i = 1 & (j \in N, v_i = f(v_j)), \\ & 0 \le y_i \le 1 & (i \in N), \end{array} \right.$$

which is equivalent to

$$\text{CTCP2} \left| \begin{array}{lll} \text{Minimize} & \max\{l_i x_i + b_i \mid i\in N, x_i > 0\} & \\ \text{subject to} & \sum\{w_i x_i \mid i \in N\} \ge W', & (2.3) \\ & \text{if } x_j < 1 \text{ then } x_i = 0 \quad (j \in N, v_i = f(v_j)), & (2.4) \\ & 0 \le x_i \le 1 \qquad (i \in N). & (2.5) \end{array} \right.$$

Here $x_i = 1 - y_i$ for each $i \in N$ and $W' = \sum\{w_i \mid i \in N\} - W$. Elimination of the condition (2.4) from the CTCP2 yields a continuous bottleneck knapsack problem, referred to as the CTCP3. (Note that the CTCP3 can be seen as a variant of the continuous minimax resource allocation problem in [6].)

Lemma 2.3 *The optimal solution of the CTCP3 satisfies the condition (2.4).*

Proof. For a given scalar parameter t, define

$$x_i(t) = \begin{cases} 1 & (l_i + b_i \le t), \\ (t - b_i)/l_i & (b_i \le t < l_i + b_i), \\ 0 & (b_i > t). \end{cases}$$

Let $t^* = \min\{t \mid \sum w_i x_i(t) \ge W'\}$. Then, the optimal solution of the CTCP3 is given by $(x_i(t^*))_{i \in N}$. When $v_i = f(v_j)$, we have $b_i \ge b_j + l_j$, and consequently $x_i(t^*) = 0$ if $x_j(t^*) < 1$. Therefore, $(x_i(t^*))_{i \in N}$ satisfies (2.4). ∎

With a slight modification, the algorithm proposed in [2] searches the parameter t^* of the above proof in linear time.

Theorem 2.4 *The continuous tree center problem is reducible to the continuous bottleneck knapsack problem in* $O(n)$ *time, and solvable in* $O(n)$ *time.*

Next, we shall show that the discrete tree center problem (DTCP) can be reduced to the 0-1 bottleneck knapsack problem.

We assume that the root v_1 is a vertex center. The following property is implicitly used in Hakimi, Schmeichel and Labbé [3].

Lemma 2.5 *There exists a discrete tree center containing a vertex center.*

Thus, we have only to consider discrete subtrees containing v_1. The following is a corollary of Lemma 2.2.

Corollary 2.6 *Let* S *be any discrete subtree containing* v_1. *If* $S \ne T$ *then*

$$\mathrm{ecc}(S) = \max\{l_i + b_i \mid i \in N,\ e_i \subseteq T \setminus S\}.$$

By using a 0-1 valued vector $(y_i)_{i \in N}$, we can reformulate the DTCP to

DTCP1

Minimize	$\max\{(l_i + b_i)(1 - y_i) \mid i \in N\}$	
subject to	$\sum\{w_i y_i \mid i \in N\} \le W$,	
	if $y_j > 0$ then $y_i = 1$	$(j \in N, v_i = f(v_j))$,
	$y_i \in \{0, 1\}$	$(i \in N)$.

Putting $W' = \sum\{w_i \mid i \in N\} - W$ and $x_i = 1 - y_i$ for each $i \in N$, we obtain

DTCP2

Minimize	$\max\{(l_i + b_i)x_i \mid i \in N\}$		
subject to	$\sum\{w_i x_i \mid i \in N\} \ge W'$,		(2.6)
	if $x_j < 1$ then $x_i = 0$	$(j \in N, v_i = f(v_j))$,	(2.7)
	$x_i \in \{0, 1\}$	$(i \in N)$.	(2.8)

By removing the condition (2.7) from the DTCP2, we obtain a 0-1 bottleneck knapsack problem. For such a problem, the 0-1 valued vector $(x_i^*)_{i \in N}$ given by

$$x_i^* = \begin{cases} 1 & (l_i + b_i \leq l_k + b_k), \\ 0 & (l_i + b_i > l_k + b_k) \end{cases}$$

is optimal, where $k \in N$ satisfies

$$\sum\{w_i \mid i \in N,\ l_i + b_i < l_k + b_k\} < W' \leq \sum\{w_i \mid i \in N,\ l_i + b_i \leq l_k + b_k\}.$$

Since $l_i + b_i > l_j + b_j$ for any $i, j \in N$ with $v_i = f(v_j)$, the condition (2.7) holds for $(x_i^*)_{i \in N}$. We can find the index k in $\mathrm{O}(n)$ time by the median finding algorithm [1].

Theorem 2.7 *The discrete tree center problem is reducible to the 0-1 bottleneck knapsack problem in* $\mathrm{O}(n)$ *time, and solvable in* $\mathrm{O}(n)$ *time.*

Acknowledgement

We are grateful to Takeaki Uno of Tokyo Institute of Technology for his valuable comments, and to anonymous referees for their suggestions and relevant reference [8].

References

[1] M. Blum, R. W. Floyd, V. Pratt and R. L. Rivest. Time Bounds for Selection. *J. Comput. System Sci.* **7** (1973) 448–461.

[2] P. Brucker. An $\mathrm{O}(n)$ Algorithm for Quadratic Knapsack Problems. *Oper. Res. Lett.* **3** (1984) 163–166.

[3] S. L. Hakimi, E. F. Schmeichel and M. Labbé. On Locating Path- or Tree-Shaped Facilities on Networks. *Networks* **23** (1993) 543–555.

[4] G. Y. Handler. Minimax Location of a Facility in an Undirected Tree Graph. *Transportation Sci.* **7** (1973) 287–293.

[5] S. M. Hedetniemi, E. J. Cockayne and S. T. Hedetniemi. Linear Algorithms for Finding the Jordan Center and Path Center of a Tree. *Transportation Sci.* **15** (1981) 98–114.

[6] T. Ibaraki and N. Kato. *Resource Allocation Problems: Algorithmic Approaches.* The MIT Press, Massachusetts (1988).

[7] E. Minieka. The Optimal Location of a Path or Tree in a Tree Network. *Networks* **15** (1985) 309–321.

[8] A. Tamir. Fully Polynomial Approximation Schemes for Locating a Tree-Shaped Facility: A Generalization of the Knapsack Problem. Technical Report, Tel Aviv University (1993).