The Connected p-Center Problem on 3-Cactus Networks with Forbidden Nodes

William Chung-Kung Yen and Chien-Tsai Chen
Department of Information Management, Shih Hsin University, Taipei, Taiwan
Email: {ckyen001@ms7.hinet.net, hbk_156@hotmail.com}

ABSTRACT

Let $G(V, E, W)$ be a network with n-node-set V and m-link-set E, where each link e is associated with a positive distance $W(e)$. The traditional p-Center problem is to locate some type of facilities at p nodes to minimize the maximum distance between any node and its nearest facility. This paper proposes an additional practical constraint. We restrict that the subnetwork induced by the p facility nodes must be connected. The resulting problem is called the Connected p-Center problem (the CpC problem). This paper designs an $O(pn)$-time algorithm for the CpC problem on 3-cactus networks using a very elegant approach. Then, we extend the algorithm to 3-cactus networks with forbidden nodes, i.e. some nodes in V cannot be selected as centers, and the complexity is still $O(pn)$.

Keywords: connected p-center, induced subnetwork, cactus network, k-cactus network, network with forbidden vertices

1. INTRODUCTION

Client/server architecture has become a basis for almost all networks and distributed systems. Consider the following practical and interesting situation in many real client/server network environments and distributed systems. There are resources, e.g., servers, programs, routers, data objects, etc., to be established at some nodes to provide services requested by the clients over a computer network. If there is no facility at a node u, the clients at u need to route to a nearest facility node to meet its requirement. This type of application corresponds to the fundamental discrete location problem, called the p-Center problem.

Let $G(V, E, W)$ be a network with n-node-set V and m-link-set E, where each link e is associated with a positive distance $W(e)$. For any $Q \subseteq V$, the *distance between Q and any node $v \notin Q$*, is defined as $d(v, Q) = \min_{u \in Q} \{d(v, u)\}$, where $d(x, y)$ denotes the distance (length) of the shortest path between any pair of nodes x and y. Meanwhile, we define $\delta(Q) = \max_{v \in V-Q} \{d(v, Q)\}$. The p-Center problem can be defined formally as follows [7].

***The p-Center problem*:** Given a network $G(V, E, W)$ and a positive integer p, identify a subset $H = \{h_1, \ldots, h_p\}$ of V, called a p-center of G, such that $\delta(H)$ is minimized.

Fig. 1 shows an input network of the p-Center problem. In the case $p = 2$, it is easy to verify that $H = \{v_3, v_6\}$ is a 2-center of this network such that $\delta(H) = 8$ is minimized.

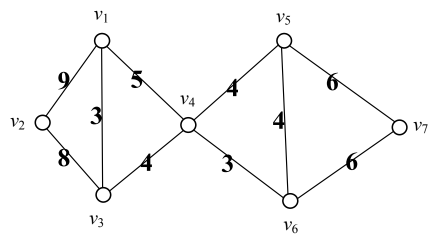

Figure 1. A network with distances on links.

2. MOTIVATIONS AND RELATED WORKS

The p-Center problem has very wide-area applications to many real-world systems and environments. Its application areas include finding the best locations of industrial plants, warehouses, distribution centers, and public service facilities in transportation networks, as well as locating various service facilities in telecommunication and computer networks. Indeed, extensive research effort has been done on this problem [1, 5, 7, 13]. The problem on general networks (graphs) is NP-Hard [9, 10]. In [18], the author provided an $O(n)$-time algorithm for the 1-Center problem on interval graphs. The author in [6] extended the result of [18] for the problem under the assumption that the endpoints of input intervals are sorted and the time-complexity is $O(n)$. Lan Y-F et al. proposed a linear-time algorithm for finding centers on weighted cactus graphs [16]. Frederickson solved the p-center problem on trees in linear-time (without necessarily restricting the location of the facilities to the vertices of the tree) using parametric search [8]. Bespamyatnikh et al. gave an $O(pn)$-time algorithm for the problem on circular-arc graphs [4]. Hsu et al. presented a general p-facility location problem on the real line with unimodal distance functions and an $O(pn^2)$-time algorithm was proposed [11]. Kariv and Hakimi addressed the p-center problem on general graphs [13]. In [20], Tamir proved that the p-center problems on weighted and unweighted networks can be solved in $O(n^p \; m^p \; \log^2 n)$-time and $O(n^{p-1} \; m^p \; \log^3 n)$-time, respectively. In addition, some research has been done on approximating the

p-center problem [3, 12].

This paper proposes a very practical additional constraint: the subnetwork induced by any p-center must be connected. This issue is very important and practical to real networks. Suppose that Fig. 1 represents some regional backbone network of Internet and each vertex denotes a web server. Now, we want to assign three servers as cache servers. Assume that $\{v_1, v_5, v_6\}$ are selected as cache servers. Consider the situation that v_1 receives a request from v_2. If the current load of v_1 is very heavy, then v_1 must determine a route such that it can pass the request to v_5 or v_6 for further processing to reduce its load. The dynamic routing overhead occurs because v_1 is not directly connected to either v_5 or v_6. Therefore, we prefer to allocate the cache servers so that their induced subnetwork is connected. This can reduce dynamic routing overhead for improving load balance among these cache servers. We call a p-center that induces a connected subnetwork as a *connected p-center* hereafter. From this point of view, it is reasonable and natural to assume that $p \geq 2$ and $n \geq 2$ in the rest of this paper.

The Connected p-Center Problem (The CpC problem): Given a network $G(V, E, W)$ and a positive integer $p \geq 2$, identify a connected p-center $Q = \{q_1, \ldots, q_p\}$ such that $\delta(Q)$ is minimized. We denote $\delta(G) = \delta(Q)$ and Q is called an *optimal solution* of the CpC problem on G hereafter.

Let us examine the network shown in Fig. 1 again and also consider the case $p = 2$. Verifying that $Q = \{v_3, v_4\}$ is a connected 2-center such that $\delta(Q) = 9$ is minimized is simple and we say that $\delta(G) = 9$.

The rest of this paper is organized as follows. Section 3 will design an $O(pn)$-time algorithm for the CpC problem on 3-cactus networks using a very elegant approach. Then, in Section 4, we extend the algorithm to 3-cactus networks with forbidden nodes, i.e. some nodes in V cannot be selected as centers, and the complexity is still $O(pn)$. Finally, the conclusion will be drawn in Section 5.

3 AN *O*(*pn*)-TIME ALGORITHM FOR THE C*p*C PROBLEM ON 3-CACTUS NETWORKS

A network is a *cactus network* if every link belongs to at most one cycle. Alternatively, a cactus network is connected network in which two cycles have at most one vertex in common [22]. A cactus network C can be constructed from a tree network T via replacing some links of T by cycles of arbitrary length greater than or equal to 3. The motivation that we consider the CpC problem on cactus networks can be described as follows. Firstly, cactus networks are often used to model real-world systems or environments when a tree network is inappropriate or is not enough. Typical examples arise in telecommunications when considering feeder networks for rural, suburban, and light urban regions [14, 15]. Moreover, the ring and bus are two very popular and essential structures used in local area networks. The combination of several local area networks forms a cactus network naturally [15]. Secondly, some literature studied the p-center and its related problems on cactus networks [2, 14, 15, 19, 22]. But, no research result deals with the CpC problem on cactus networks so far. In this section, we will propose an original study of the CpC problem on cactus networks. For any integer $k \geq 3$, a *k-cactus network* is a cactus network in which every cycle consists of at most k nodes. The section will solve the CpC problem on 3-cactus networks.

To obtain an optimal solution of the CpC problem on a 3-cactus network C, we randomly choose a node r as the *root*. Then, we partition all nodes v of C into different levels, denoted as $L(v)$, with the level of r is equal to 1, i.e., $L(r) = 1$, and the network will be denoted as $C(r)$ as shown in Fig. 2. After this preprocessing, we will deal with the CpC problem on $C(r)$ in the rest of this section.

Definition 1. For each node u, Children(u) = $\{v \mid L(v) = L(u) + 1$ and $(u, v) \in E\}$. If Children(u) is an empty set, then u is called a *leaf node*. Otherwise, u is called a *non-leaf node*.

Definition 2. For each non-root node u, Parent(u) is the node v such that $(u, v) \in E$ with $L(v) = L(u) - 1$.

Definition 3. For each pair of nodes u and v with $(u, v) \in E$ and Parent(u) = Parent(v), denote u = Brother(v) and v = Brother(u). The pair u and v are called a *B-pair*. In Fig. 2, b_j and f_j are B-pairs, i.e., Brother(b_j) = f_j and Brother(f_j) = b_j, for all $1 \leq j \leq \beta$.

Definition 4. For each non-root node u, the following definitions are made.

(1) Let P: $u = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_\rho = r$ be the path from u to r in which v_{j+1} = Parent(v_j), $1 \leq j < \beta$. Then, $v_2, \ldots, v_\rho$ are called the *ancestors* of u and denoted as Ancestors(u) = $\{v_2, \ldots, v_\rho\}$.

(2) $C(u)$ denotes the subnetwork rooted at node u as shown in Fig. 2.

Definition 5. For each B-pair u and v, $C(u, v)$ denotes the subnetwork formed by the union of the link (u, v), $C(u)$, and $C(v)$. See Fig. 2.

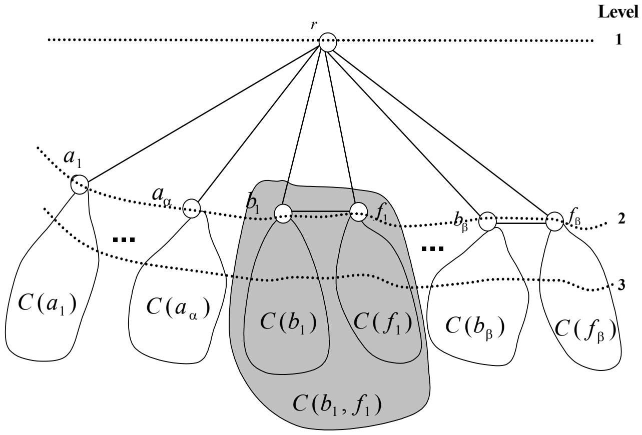

Figure 2. A 3-cactus network with$L(r)$ = 1.

In the rest of this paper, we will use $C(u)$ to represent the subnetwork rooted at any node u and the node-set of $C(r)$ alternatively if no confusion occurs. The same representation is also applied to $C(u, v)$ for any B-pair u and v.

Lemma 1. Suppose that H is any connected p-center of $C(r)$. Then, one of the following conditions holds.

1. There exists u such that $H \subseteq C(u)$ and $u \in H$.
2. There exists a B-pair u and v such that $H \subseteq C(u, v)$ and $u, v \in H$.

Proof: Let $M = \{y \mid y \in H$ and $L(y)$ is minimized$\}$. By the definition of levels of all nodes, we must have $|M|$ is equal to either 1 or 2. The proof can be completed just by examining the set M. ■

Let $Q^{(u)}$ be an optimal solution of the CpC problem on $C(u)$ with $u \in Q$, for each $C(u)$. Meanwhile, $Q^{(u,v)}$ denotes any optimal solution of the CpC problem on $C(u, v)$ such that $u, v \in Q$, for each $C(u, v)$.

Lemma 2. Let $\Omega = \{Q^{(u)} \mid u \in V\}$ and $\Psi = \{Q^{(u,v)} \mid u$ and v is a B-pair of $C(r)\}$. If $Q \in \Omega \cup \Psi$ and $\delta(Q) \le \delta(H)$, for all $H \in \Omega \cup \Psi$, then Q is an optimal solution of the CpC problem on $C(r)$.

Let $\Delta(Q^{(u)}) = \max\{d(y, Q^{(u)}) \mid y \in C(u) - Q^{(u)}\}$ and $\theta(Q^{(u)}) = \max\{d(y, Q^{(u)}) \mid y \in V - C(u)\}$, for each $Q^{(u)}$. Since $u \in Q^{(u)}$, we have $\theta(Q^{(u)}) = \max\{d(y, u) \mid y \in V - C(u)\}$.

$$\delta(Q^{(u)}) = \max\{\Delta(Q^{(u)}), \theta(Q^{(u)})\} \text{ --<1>}$$

The following formula for each $Q^{(u,v)}$ can be easily proved by the similar way, where $\Delta(Q^{(u,v)}) = \max\{d(y, Q^{(u,v)}) \mid y \in C(u, v) - Q^{(u,v)}\}$ and $\theta(Q^{(u,v)}) = \max\{d(y, \{u, v\}) \mid y \in V - C(u, v)\}$.

$$\delta(Q^{(u,v)}) = \max\{\Delta(Q^{(u,v)}), \theta(Q^{(u,v)})\} \text{ --<2>}$$

Definition 6. For each node u of $C(r)$, if u is a leaf node, then define MLen(u) = 0. Otherwise, let Children(u) = $\{x_1, \ldots, x_k, y_1, z_1, \ldots, y_q, z_q\}$, where y_j and z_j are B-pairs, $1 \le j \le q$. The following values are associated with u.

1. MLen(u) = $\max\{\max_{1\le i\le k}\{\text{MLen}(x_i) + d(x_i, u)\}, \max_{1\le j\le k}\{\text{MLen}(y_j) + d(y_j, u)\}, \max_{1\le j\le k}\{\text{MLen}(z_j) + d(z_j, u)\}\}$
2. MLen(u, $\overline{x_t}$) = $\max\{\max_{1\le i\ne t\le k}\{\text{MLen}(x_i) + d(x_i, u)\}, \max_{1\le j\le k}\{\text{MLen}(y_j) + d(y_j, u)\}, \max_{1\le j\le k}\{\text{MLen}(z_j) + d(z_j, u)\}\}$, $1 \le t \le k$.
3. MLen(u, $\overline{y_t}$) and MLen(u, $\overline{z_t}$), for all $1 \le t \le q$, can be defined in the same way.
4. MLen(u, $\overline{y_t}$, $\overline{z_t}$) = $\max\{\max_{1\le i\le k}\{\text{MLen}(x_i) + d(x_i, u)\}, \max_{1\le j\ne t\le k}\{\text{MLen}(y_j) + d(y_j, u)\}, \max_{1\le j\ne t\le k}\{\text{MLen}(z_j) + d(z_j, u)\}\}$, $1 \le t \le k$.

The following lemma can be proved by simple modification of the technique for proving Lemma 2 of [21]. We omit the details here.

Lemma 3. For each node u with Children(u) = $\{x_1, \ldots, x_k, y_1, z_1, \ldots, y_q, z_q\}$, if MLen($x_i$),$1 \le i \le k$, and MLen($y_j$) and MLen($z_j$), $1 \le j \le q$, have been computed, then MLen(u), MLen(u, $\overline{x_t}$), $1 \le t \le k$; MLen(u, $\overline{y_t}$) and MLen(u, $\overline{z_t}$), $1 \le t \le q$; and MLen(u, $\overline{y_t}$, $\overline{z_t}$), $1 \le t \le q$, can be obtained in $O(k + q)$-time.

It is trivial to see that $\theta(Q^{(r)}) = 0$. The correctness of the following formulas and Lemma 4 can be easily verified.

$\theta(Q^{(u)}) = d(r, u) + \text{MLen}(r, \overline{u})$, $L(u) = 2$ and u has no brother. --<3>

$\theta(Q^{(u)}) = d(\text{Parent}(u), u) + \max\{\theta(Q^{(\text{Parent}(u))}), \text{MLen}(\text{Parent}(u), \overline{u})\}$, $L(u) \ge 3$ and u has no brother. --<4>

$\theta(Q^{(u)}) = \max\{d(r, u) + \text{MLen}(r, \overline{u}), \text{MLen}(\text{Brother(u)}) + d(\text{Brother}(u), u)\}$, $L(u) = 2$ and u has a brother. --<5>

$\theta(Q^{(u)}) = \max\{d(\text{Parent}(u), u) + \max\{\theta(Q^{(\text{Parent}(u))}), \text{MLen}(\text{Parent}(u), \bar{u})\}, \text{MLen}(\text{Brother}(u)) + d(\text{Brother}(u), u)\}$, $L(u) \geq 3$ and u has a brother. --<6>

$\theta(Q^{(u,v)}) = d(r, \{u, v\}) + \text{MLen}(r, \bar{u}, \bar{v})$, $L(u) = L(v) = 2$ --<7>

$\theta(Q^{(u,v)})$
$= d(\text{Parent}(u), \{u, v\}) + \max\{\theta(Q^{(\text{Parent}(u))}), \text{MLen}(\text{Parent}(u), \bar{u}, \bar{v})\}$, $L(u) \geq 3$. --<8>

Lemma 4. $\theta(Q^{(u)})$, for all nodes u, and $\theta(Q^{(u,v)})$, for all B-pairs u and v, can be computed in $O(n)$-time.

Now, consider each node x with the brother node Brother(x). Define $\pi(x) = \max_{y \in C(x)}\{d(y, \{\text{Parent}(x), \text{Brother}(x)\})\}$. Physically, $\pi(x)$ is the distance of the longest shortest path between any vertex y in $C(x)$ and one of the two vertices in {Parent(x), Brother(x)}. It is easy to verify that $\pi(x) = \text{MLen}(x) + \min\{W(x, \text{Parent}(x)), W(x, \text{Brother}(x))\}$.

For each non-root node u, $\mu(u)$ is computed by the following rules.

1. $\mu(u) = d(u, \text{Parent}(u))$, if u is a leaf node and u has no brother.
2. $\mu(u) = \max_{y \in \text{Children}(u)}\{\mu(y)\} + W(u, \text{Parent}(u))$, if u is non-leaf and u has no brother.
3. $\mu(u) = \text{MLen}(u) + d(u, \text{Parent}(u))$ and $\mu(\text{Brother}(u)) = \pi(\text{Brother}(u))$, if u has a brother and $\pi(u) \geq \pi(\text{Brother}(u))$.
4. $\mu(u) = \pi(u)$ and $\mu(\text{Brother}(u)) = \text{MLen}(\text{Brother}(u)) + d(\text{Brother}(u), \text{Parent}(\text{Brother}(u)))$, if u has a brother and $\pi(u) < \pi(\text{Brother}(u))$.

Finally, the value $\mu(r) = \max_{y \in \text{Children}(u)}\{\mu(y)\}$.

Lemma 5 directly holds from the above computational results.

Lemma 5. For each node u, $\mu(u) \geq \mu(z)$, for all $z \in C(u)$. Meanwhile, for each non-root node x of $C(r)$, $\mu(y) > \mu(x)$, for all $y \in$ Ancestors(x).

Lemma 6. For any node u, let $H = \{h_1, \ldots, h_\alpha\}$ be a set of vertices in $C(u)$ such that $\mu(h_1), \ldots, \mu(h_\alpha)$ are the first α largest numbers among $\{\mu(v) \mid v \in C(u)\}$. Then, H forms a connected subnetwork of $C(u)$.

Proof: Lemma 5 implies that $h_1 = u$. Assume that the H does not induce a connected subnetwork of $C(u)$. Let j, $2 \leq j \leq \alpha$, be the smallest number such that $H^* = \{h_1, \ldots, h_j\}$ induces a connected subnetwork but h_{j+1} is not adjacent to any vertex in H^*. Since we only consider connected networks, it implies that Parent(h_{j+1}) $\notin H$ and Brother(h_{j+1}) $\notin H$. But Lemma 5 states that $\mu(\text{Parent}(h_{j+1})) > \mu(h_{j+1})$ and also implies that the first α largest numbers among $\{\mu(v) \mid v \in C(u)\}$ must include $\mu(\text{Parent}(h_{j+1}))$, i.e., Parent($h_{j+1}$) $\in H$. A contradiction occurs. ■

Lemma 7. For any node u, let $Q = \{q_1, \ldots, q_p\}$ be a set of vertices in $C(u)$ such that $\mu(q_1), \ldots, \mu(q_p)$ are the first p largest numbers among $\{\mu(v) \mid v \in C(u)\}$. Then, Q is an optimal connected p-center of $C(u)$ with $u \in Q$, i.e., Q can be an instance of $Q^{(u)}$ and we can derive the following formula, where $V(C(u))$ denotes the node-set of C(u).

$$\Delta(Q^{(u)}) = \begin{cases} \infty, |V(C(u))| < p \\ 0, |V(C(u))| = p \\ \lambda, \lambda \text{ is the } (p+1)^{\text{th}} \text{ largest number among } \{\mu(v) \mid v \in C(u)\} \end{cases}$$

--<7>

Lemma 8. For any B-pair u and v, let $Q = \{u, v, q_1, \ldots, q_{p-2}\}$ be a set of vertices in $C(u, v)$ such that $\mu(q_1), \ldots, \mu(q_{p-2})$ are the first $(p-2)$ largest numbers among $\{\mu(v) \mid v \in C(u) - \{u, v\}\}$. Then, Q is an optimal connected p-center of $C(u, v)$, i.e., Q can be an instance of $Q^{(u,v)}$ and we can derive the following formula, where $C(u, v)$ denotes the node-set of $C(u, v)$.

$$\Delta(Q^{(u,v)}) = \begin{cases} \infty, |V(C(u,v))| < p \\ 0, |V(C(u,v))| = p \\ \lambda, \lambda \text{ is the } (p-1)^{\text{th}} \text{ largest number among } \{\mu(v) \mid v \in C(u,v) - \{u,v\}\} \end{cases}$$

--<8>

Lemma 9. Suppose that u is any non-leaf node. Let $H^{(v)} = \begin{cases} Q^{(v)}, |V(C(v))| \geq p \\ V(C(v)), |V(C(v))| < p \end{cases}$, for each $v \in$ Children(u). Then, $Q^{(u)} = \{q_1, \ldots, q_p\}$ be a set of vertices in $C(u)$ such that $\mu(q_1), \ldots, \mu(q_p)$ are the first p largest numbers among $\{\mu(y) \mid y \in \{u\} \cup \bigcup_{v \in \text{Children}(u)}\{H^{(v)}\}\}$.

Lemma 10. Suppose that u and v is any B-pair. Let

$H^{(u)} = \begin{cases} Q^{(u)} - \{u\}, |V(C(u))| \geq p \\ V(C(u)) - \{u\}, |V(C(u))| < p \end{cases}$ and

$H^{(v)} = \begin{cases} Q^{(v)} - \{v\}, |V(C(v))| \geq p \\ V(C(v)) - \{v\}, |V(C(v))| < p \end{cases}$. Then, $Q^{(u,v)}$ = {u, v, q_1, …, q_{p-2}}, in which {q_1, …, q_{p-2}} is a set of vertices in $C(u, v)$ such that $\mu(q_1)$, …, $\mu(q_{p-2})$ are the first (p – 2) largest numbers among $H^{(u)} \cup H^{(v)}$.

Lemma 11. $\Delta(Q^{(u)})$, for all nodes u, and $\Delta(Q^{(u,v)})$, for all B-pairs u and v, can be computed in $O(pn)$-time.

***Proof*:** We can achieve all computations using Breadth-First-Search technique to scan each node u and each B-pair u and v from leaf nodes to r. Lemma 9 and Lemma 10 just imply that the major task is to find the p^{th} largest number among $p * |\text{Children}(u)|$ numbers, for each node u. It is well-known that finding the k^{th} largest number among n numbers can be done in $O(n)$-time, for any given k [17]. Therefore, the total time-complexity can be easily verified to be $O(pn)$. ■

Theorem 1. The CpC problem on 3-cactuc networks can be solved in $O(pn)$-time.

***Proof*:** This theorem follows directly from Lemma 2, Lemma 4, and Lemma 11. ■

4. EXTENSION TO 3-CACTUS NETWORKS WITH FORBIDDEN NODES

In real-world systems such as computer and telecommunication networks, some nodes may not be suitable to be selected as center nodes due to function failure or some practical constraints, such as capacity, processing ability, etc. We use F to represent the set of such nodes and called them *forbidden nodes*. The resulting problem can be now defined as follows:

***The Forbidden Connected p-Center Problem (The FCpC problem)*:** Given a network $G(V, E, W)$, a subset F of V, and a positive integer constant $p \geq 2$, identify a connected p-center Q = {q_1, …, q_p} of G such that $\delta(Q)$ is minimized under the restriction that $Q \cap F = \emptyset$.

The section will extend the results of the previous section to the FCpC problem and the time-complexity will remain $O(pn)$. In the rest of this section, if we call H a connected p-center, then it means that $H \cap F = \emptyset$.

Definition 7. For each node u of the cactus $C(r)$, if $u \in F$, then define $\Phi(u)$ = 0. Otherwise, define $\Phi(u)$ = $|\Pi(u)|$, where $\Pi(u) = \{y \mid y \in C(u) - F\}$.

It is easy to see that $\Phi(u)$, for each non-forbidden node u, i.e., $u \notin F$, can be computed using the following rules.

1. $\Phi(u)$ = 1, if u is a leaf node.
2. $\Phi(u) = \sum_{y \in (\text{Children}(u) - F)} \Phi(y)$ + 1, if u is not a leaf node.

After that, another value $\eta(u)$ for each node u are computed as follows:

1. $\eta(u) = \Phi(u)$, if $u \in F$ or u has no brother.
2. $\eta(u) = \Phi(u) + \Phi(\text{Brother}(u))$, if u belongs to a B-pair.

Lemma 12. After computing $\eta(u)$, for all nodes u, if $\eta(u) < p$, then u can not belong to any connected p-center H.

Lemma 12 implies that all nodes u with $\eta(u) < p$ can be viewed as additional forbidden vertices. Therefore, we can assume that $\eta(u) \geq p$ hereafter. As stated in previous section, we also randomly choose a node $r \notin F$ as the root.

Lemma 13. Suppose that H is any connected p-center of $C(r)$. Then, one of the following conditions holds.

1. There exists $u \notin F$ such that $H \subseteq C(u)$ and $u \in H$.
2. There exists a B-pair u and v, $u, v \notin F$, such that $H \subseteq C(u, v)$ and $u, v \in H$.

Lemma 14. Let $\Omega = \{Q^{(u)} \mid u \notin F\}$ and $\Psi = \{Q^{(u,v)} \mid u$ and v is a B-pair of $C(r)$ and $u, v \notin F\}$. If $Q \in \Omega \cup \Psi$ and $\delta(Q) \leq \delta(H)$, for all $H \in \Omega \cup \Psi$, then Q is an optimal solution of the FCpC problem on $C(r)$.

The remaining task can be achieved using the similar techniques in Section 2 and the following theorem can then be ascertained.

Theorem 2. The FCpC problem on 3-cactus networks can be solved in $O(pn)$-time.

5. CONCLUSIONS

This paper addressed the Connected p-Center problem (the CpC problem) on networks. This problem can be viewed as a more practical variant of the traditional p-Center problem. We proposed an $O(pn)$-time algorithm for the CpC problem on 3-cactus networks using dynamic programming strategy. Then, the algorithmic result was extended to the situation that the vertices in F of the input 3-cactus network are forbidden. The time-complexity is still $O(pn)$.

In the future, the first practical and meaningful issue is to extend our algorithms to k-cactus networks, $k \geq 4$, and cactus networks. Meanwhile, solving the CpC and FCpC problems on other classes of networks, such as planar networks and interval networks, is also a very typical research topic. Meanwhile, identifying other variants of the traditional p-Center problem is also a very important task. For example, requiring that the p-centers must be "total", i.e., the subnetwork induced by the p-centers has no isolated nodes, is another practical variant with applications to real networks.

ACKNOWLEDGEMENT

This research was supported by National Science Council, Taiwan, under the contract number NSC 94-2213-E-128-005.

REFERENCES

1. Abdelaziz F. (2006), 1-center problem on the plane with uniformly distributed demand points, *Operations Research Letters*, Vol. 34, Iss. 3, 264-268.
2. Averbakh I. and Berman O. (1995), Sales-delivery man problems on treelike networks, *Networks*, Vol. 25, 45-58.
3. Bar-Ilan J, Peleg D (1991), Approximation algorithms for selecting network centers. in *Proceedings of workshop on algorithms and data structures*, 343–354.
4. Bespamyatnikh S, Bhattacharya B, Keil M, Kirkpatrick D, Segal M. (2002), Efficient algorithms for centers and medians in interval and circular-arc graphs, *Networks*, Vol. 39, 144–152.
5. Burkard R. E. and Dollani Helidon (2003), Center problems with pos/neg weights on trees, *European Journal of Operational Research*, Vol. 145, Iss. 3, 483-495
6. Chenga T. C. E., Kang L., and Ng C. T. (to appear), An improved algorithm for the p-center problem on interval graphs with unit lengths, *Computers & Operation Research*.
7. Daskin M. S. (1995), *Networks and Discrete Location, Models, Algorithms, and Applications*, John Wiley & Sons, Inc., New York.
8. Frederickson G. (1991), Parametric search and locating supply centers in trees, in *Proceedings of workshop on algorithms and data structures*, 299–319.
9. Garey M. R. and Johnson D. S. (1978), *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*, Bell Laboratories, Murray Hill, Freeman & Co., N. J.
10. Golumbic M. C. (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, Inc., New York.
11. Hsu V. N., Lowe T. J., Tamir A. (1997), Structured p-facility location problems on the line solvable in polynomial time, *Operations Research Letter*s, Vol. 21, 159–164.
12. Hochbaum D, Shmoys D. B. (1986), A unified approach to approximation algorithms for bottleneck problems, *Journal of the ACM*, Vol. 33, 533–550.
13. Kariv O, Hakimi S. L. (1979), An algorithmic approach to network location problems I: the p-centers, *SIAM Journal of Applied Mathematics*, Vol. 37, 514–538.
14. Koontz W. L. G. (1980), Economic evaluation of loop feeder relief alternatives, *The Bell System Technical Journal*, Vol. 59, 277-281.
15. Lan Y-F, Wang Y-L (2000), An optimal algorithm for solving the 1-median problem on weighted 4-cactus graphs, *European Journal of Operational Research*, Vol. 122, 602-610.
16. Lan Y-F, Wang Y-L, Suzuki H. (1999), A linear-time algorithm for solving the center problem on weighted cactus graphs, *Information Processing Letters*, Vol. 71, 205-212.
17. Lee R. C. T., Chang R. C., Tseng S. S., and Tsai Y. T. (2002), *Introduction to the Design and Analysis of Algorithms*, Flag Publishing Company, Taipei, Taiwan.
18. Olariu S. (1990), A simple linear-time algorithm for computing the center of an interval graph, *International Journal of Computer Mathematics*, Vol. 24, 121–128.
19. Hedetniemi S. T., Laskar R., and Pfaff J. (1986), A linear-time algorithm for finding a minimum dominating set in a cactus, *Discrete Applied Mathematics*, Vol. 13, 287-292.
20. Tamir A. (1988), Improved complexity bounds for center location problems on networks by using dynamic data structures, *SIAM Journal of Discrete Mathematics*, Vol. 1, 377–396.
21. Yen W. C-K (2002), Bottleneck domination and bottleneck independent domination on graphs, *Journal of Information Science and Engineering*, Vol. 18, 311-331.
22. Zmazek B. and Ž erovnik J. (2004), The obnoxious center problem on weighted cactus graphs, *Discrete Applied Mathematics*, Vol. 136, 377-386.