Finding Connected p-Centers on Interval Graphs

William Chung-Kung Yen
Department of Information Management
Shih Hsin University, Taipei, Taiwan
Email: ckyen001@ms7.hinet.net

Shung C-S Yang
Department of Information Management
Shih Hsin University, Taipei, Taiwan
Email: shung007@hotmail.com

***Abstract*—Let $G(V, E)$ be an undirected and connected simple graph with n vertices. A positive weight $w_1(v)$ is associated to each vertex v and a positive length $w_2(e)$ is associated with each edge e. Given an integer $p \geq 1$, the fundamental p-Center problem is to locate a p-vertex set Q of G for the establishment of facilities. Minimizing the maximum weighted distance for each vertex v in $V - Q$ to its nearest facility site is the most important criteria. This paper focuses on the issue of finding connected p-centers on graphs, called the Connected p-Center problem (the CpC problem), which is a new practical variant from the p-Center problem. A p-center Q is connected if the subgraph induced by the vertices in Q is connected. Under the assumption that the clique path is given, this paper designs an $O(n)$-time algorithm for the CpC problem on interval graphs with $w_1(v) = 1$, for all vertices v, and $w_2(e) = 1$, for all edges e.**

***Index Terms*—connected p-centers, NP-Hard, interval graphs, clique path**

I. INTRODUCTION

Let $G(V, E)$ be an undirected and connected simple graph with n vertices. A positive weight $w_1(v)$ is associated to each vertex v and a positive length $w_2(e)$ is associated with each edge e. The notation $d(u, v)$ denotes the length of the shortest path from v to u, for all pairs of vertices u and v. For any subset Q of V, $d(Q, v) = \min\{d(u, v) \mid \text{for all } u \in Q\}$, for all $v \in (V - Q)$ and $\beta(Q) = \max\{w_1(v) * d(Q, v) \mid \text{for all } v \in (V - Q)\}$. Given an integer $p \geq 1$, the p-Center problem involves finding a set D with p vertices of $G(V, E)$ to minimize $\beta(D)$.

The p-Center problem has significant applications to wide-area of real-life problems. For example, issues related to find the best locations for placing various facilities, such as routers and Web proxy servers over the Internet, can be modeled by the p-Center problem. If the input graph can be any general graph, the p-Center problem is well-known to be NP-Hard [5]. Due to theoretical and practical importance, extensive research results exist for the p-Center problem as well as its related problems. Some literatures gave elegant and detail surveys. Please refer [2, 3, 9, 11] for understanding the newest results about models, generalizations, variations, algorithmic results, etc., for the p-Center problem.

Recently, the authors in [12] proposed a new practical variant of the p-Center problem, called the CpC problem. From the aspects of selection of backup sites of Web proxy servers or distribution of the workloads among the routers over a computer network, it is practical to require that the p-vertex center induces a connected subgraph. In such allocations, if a center vertex u fails or its load volume is too high, then at least one of other center vertex v can take or share the tasks of u without passing messages through any non-center vertices. This can greatly improve the quality and efficiency of backup and load balancing among the center vertices.

A *p-center H* will denote any set of p vertices of the input graph G herein. Furthermore, if the subgraph induced by H is connected, then H will be called a *connected p-center*. The following formally states the problem studied in this paper.

***The Connected p-Center Problem (The CpC problem)*:** Given an integer $1 \le p < n$, find a connected p-center D of the given graph $G(V, E)$ to minimize $\beta(D)$. D is called an *optimal connected p-center*, or simply an *optimal solution* of the C*p*C problem on G. Meanwhile, $\beta(D)$ is called the *optimal value* of G.

The paper is organized as follows. Section 2 will give basic concepts of interval graphs and show some preliminary results. Give the clique path with q cliques, Section 3 will design an $O(n)$-time algorithm for the C*p*C problem on interval graphs with $w_1(v) = 1$, for all vertices v, and $w_2(e) = 1$, for all edges e. Finally, Section 4 will discuss possible extensions and make conclusions.

II. PRELIMINARY RESULTS

Interval graphs form an important subclass of chordal graphs. For any graph G, let Ω be a cycle containing four or more vertices. A *chord* of Ω is an edge connecting any two nonadjacent vertices of Ω. G is a *chordal graph* iff each cycle Ω of G has a chord [4, 6, 10]. Meanwhile, let I be any set of n intervals on the real line. A graph $G(V, E)$ is an *interval graph* if there exists a 1-to-1 and onto function f from V to I, such that $(u, v) \in E$ iff the two intervals $f(u)$ and $f(v)$ intersect nonempty, and vice versa [7].

This paper uses an elegant representation of interval graphs, called clique path, to develop the algorithm. In [6], the author proposed a *clique tree T* to represent a chordal graph G, where each node of T corresponds to a maximal clique of G and each vertex v of G corresponds to the subtree formed by the clique nodes containing v. Given any chordal graph G, a clique tree T can be constructed in $O(m + n)$ time [10]. In addition, if the clique trees are restricted to paths, called *clique paths*, then it defines the class of interval graphs [4, 8].

For the traditional p-Center problem on interval graphs, the authors in [1] proved that it remains NP-Hard even $w_2(e)$ is either 1 or 2, for all edges e, and proposed an $O(n)$-time for the case $w_2(e) = 1$, for all edges e, if the endpoints of all intervals are sorted in advance. Though the following theorem can be easily obtained from the result in [1], we give a formal proof for the completeness of this paper.

Theorem 1: The C*p*C problem is NP-Hard on interval graphs satisfying the triangle inequality, i.e., $w_2(x, y) + w_2(y, z) > w_2(x, z)$, for all three distinct vertices x, y, z.

***Proof*:** Given a positive integer p and a graph G, the problem of finding a connected dominating set H of G with $|H| = p$ is a well-known NP-complete problem [5]. Given a graph $G(V, E)$, we add edges to G to obtain a complete graph G^* with vertex-set V. Meanwhile, in G^*, $w^*_2(e) = 1$, if e is an original edge of G, and $w^*_2(e) = 1.5$, if e is an added edge. Clearly, $w^*_2(x, y) + w^*_2(y, z) > w^*_2(x, z)$, for all three distinct vertices x, y, z. We can also ascertain that a p-vertex set Q is a connected dominating set of G iff Q is a connected p-center of G^* such that $\beta(Q) = 1$. Since a complete graph can be viewed as an interval graph with single clique, we have completed the proof. ■

III. THE C*p*C PROBLEM ON INTERVAL GRAPHS

An example instance of intervals on the real line, the corresponding interval graph and clique path are illustrated from Fig. 1 to Fig. 3, respectively. They are also used to explain the main idea of our algorithm throughout this section.

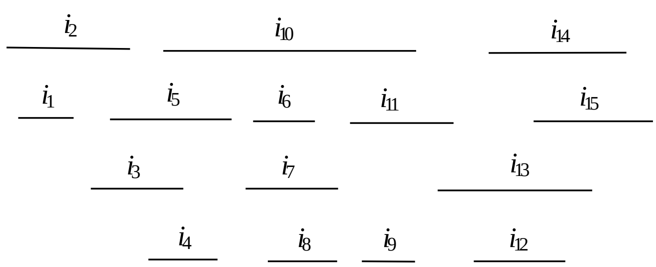

Figure 1: A set I of 15 intervals on the real line.

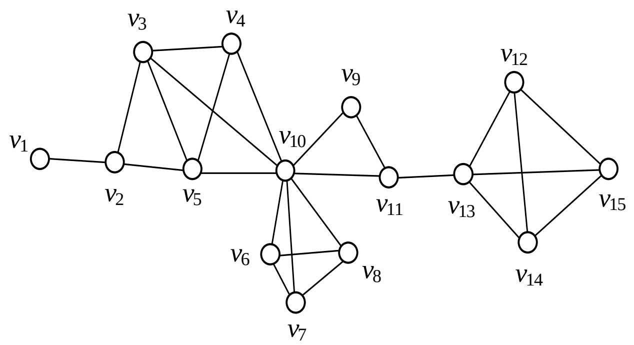

Figure 2: The interval graph corresponding to the set *I* in Fig. 1.

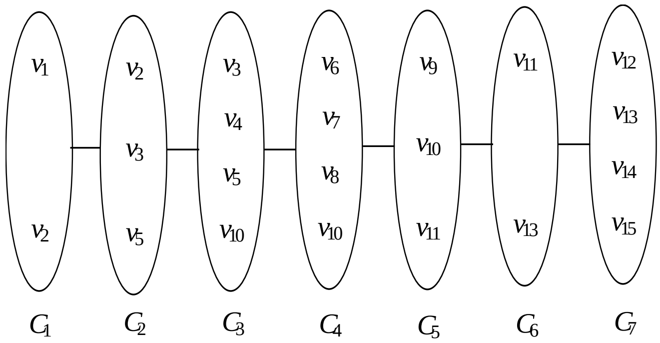

Figure 3: The clique path corresponding to the interval graph in Fig. 2.

Suppose that *P* is the clique path of the input interval graph *G* with cliques $C_1, \ldots, C_q$, where the cliques are numbered by 1, 2, …, *q* from left to right. For any two cliques C_i and C_j, we say that $C_i \le C_j$ iff $i \le j$. The notation $C_{[i,j]}$ represents the subpath *P* consisting of C_i, C_{i+1}, …, C_j, for all $i \le j$. The following properties can be easily derived from the definition of clique paths.

Property 1: For each vertex *v*, let $\Pi = \{C_t \mid v \in C_t, 1 \le t \le q\}$. Then, all $C_t \in \Pi$ form a subpath of *P*.

Property 2: $C_i \cap C_{i+1}$ is nonempty, and both $C_i - C_{i+1}$ and $C_{i+1} - C_i$ are nonempty, for all $1 \le i \le (q-1)$.

Property 3: If $v \in C_i - C_{i-1}$, then $v \notin C_{[1,i-1]}$, for all $2 \le i \le q$.

By the above properties, for each vertex *v*, define lc(*v*) and rc(*v*) to be the leftmost (smallest) clique and the rightmost (largest) clique containing *v*, respectively. Then, for any two distinct vertices *u* and *v*, we say that $u < v$ iff (rc(*u*) < rc(*v*)) or (rc(*u*) = rc(*v*) and lc(*u*) ≤ lc(*v*)). In the rest of the paper, we assume that the *n*-vertex of *G* are $v_1, \ldots, v_n$ such that $v_j < v_{j+1}$, $1 \le j \le (n-1)$. Now, the following definitions can be made.

Definition 1: For each clique C_j, the vertex f_j is defined as follows:

(1) If $j = 1$, then $f_j = v_1$.

(2) For each $2 \le j \le q$, let $\Pi = \{v \mid v \in C_j$ and lc(f_j) is the smallest$\}$. Then, f_j denotes the smallest vertex in Π.

Definition 2: For each clique $C_j - C_{j-1}$, the vertex l_j is defined as follows:

(1) If $j = q$, then $l_j = v_n$.

(2) For each $1 \le j \le (q-1)$, let $\Phi = \{v \mid v \in C_j$ and rc(l_j) is the largest$\}$. Then, l_j denotes the largest vertex in Φ

.

Definition 3: For each $2 \le j \le q$, let lc(f_j) = C_h, then define Previous(f_j) = f_h.

Definition 4: For each $1 \le j \le (q-1)$, let rc(l_j) = C_k, then define Next(l_j) = l_k. For any positive integer *p*, define $\text{Next}_{(p)}(l_j) = \{x_1, \ldots, x_p\}$ such that $x_1 = l_j$ and x_z = Next(x_{z-1}), $z = 2, \ldots, p$. For simplicity, we define $\text{Next}_{(1)}(l_j) = \{l_j\}$.

For the clique path in Fig. 3, f_j, Previous(f_j), l_j, and Next(l_j), $1 \le j \le 7$, are shown in Table 1. Note that Previous(f_4) = Previous(f_5) = f_3 = v_3. Meanwhile, $\text{Next}_{(3)}(l_2) = \{l_2, l_3, l_5\} = \{v_5, v_{10}, v_{11}\}$.

Definition 5:

(1) For each clique C_j, $2 \le j \le q$, the *backward distance* of f_j is defined as $d^B(f_j)$ = $\max\{d(f_j, y) \mid y \in C_{[1, j-1]}\}$.

(2) For each clique C_j, $1 \le j \le (q-1)$, the *forward distance* of l_j is defined as $d^F(l_j)$ = $\max\{d(l_j, x) \mid x \notin C_{[1, j-1]}\}$.

Table 1: f_j, Previous(f_j), l_j, and Next(l_j), $1 \le j \le 7$, for the clique path in Fig. 3.

j	1	2	3	4	5	6	7
f_j	v_1	v_2	v_3	v_{10}	v_{10}	v_{11}	v_{13}
Previous(f_j)	$\times$	f_1	f_2	f_3	f_3	f_5	f_6
l_j	v_2	v_5	v_{10}	v_{10}	v_{11}	v_{13}	v_{15}
Next(l_j)	l_2	l_3	l_5	l_5	l_6	l_7	$\times$

By the definition of Previous(f_j) and Next(l_j), it must have $d(f_j$, Previous(f_j)) = 1, for all $2 \le j \le q$, since f_j and Previous(f_j) belongs to the same clique. Similarly, $d(l_j$, Next(l_j)) = 1, for all $1 \le j \le (q-1)$. Meanwhile, by the definition of interval graphs, $d(f_2, v_1) = 1$ and $d(l_{q-1}, v_n) = 1$. These can easily establish the following Lemma 1.

Lemma 1: $d^B(f_j)$ and $d^F(l_j)$, for all $1 \le j \le q$, can be computed $O(q)$-time by scanning Table 1 from left to right once and from right to left once, respectively.

The following lemma can be easily verified.

Lemma 2: For each clique C_j, $2 \le j \le q$, $d^B(f_j) = d(f_j, v_1)$ and for each clique C_i, $1 \le i \le (q-1)$, $d^F(l_i) = d(l_i, v_n)$.

Now, it is the time to explain the idea of our algorithm.

Definition 6: For each connected p-center Q of G, define lc(Q) to be leftmost clique which contains at least one vertex in Q.

Definition 7: Let D_j be a connected p-center of G such that $\beta(D_j)$ is minimized among $\{Q \mid Q$ is a connected p-center of G such that lc(Q) = $C_j\}$, $1 \le j \le q$.

Definition 6 can imply the following lemma immediately.

Lemma 3:

(1) $D_j \cap C_{[1, j-1]} = \varnothing$ and D_j must contain at least one vertex in $(C_j - C_{j-1})$, $2 \le j \le q$.

(2) D_1 must contain at least one vertex in C_1.

Let $D_{\min} \in \{D_1, \ldots, D_q\}$ such that $\beta(D_{\min})$ is minimized. Definition 7 directly implies that $D_{\min}$ must be an optimal solution of G, i.e., $\beta(G)$ can be obtained in $O(q)$-time after $\beta(D_1), \ldots, \beta(D_q)$ have been obtained. The followings deal with the finding of $D_1, \ldots, D_q$.

• Finding D_1

The following two cases should be considered.

Case 1. There exists $1 \le k \le p$ such that $\text{Next}_{(k)} = \{x_1, \ldots, x_k\}$ and $x_k \in C_q$.

In this case, it is easy to see that C_j contains at least one vertex in $\text{Next}_{(k)}$, $1 \le j \le q$. Clearly, $D_1 = \text{Next}_{(k)} \cup H$, where H contains any $(p-k)$ vertices in $(V - \text{Next}_{(k)})$ and $\beta(D_1) = 1$ since we assume that $p \le n$.

Case 2. $\text{Next}_{(p)} = \{x_1, \ldots, x_p\}$ with $x_p \in C_t$ and $t < q$.

Let Q be any connected p-center of G such that Q contains at one vertex in C_1. The reasoning so far easily guarantees that $\beta(Q) \ge \beta(\text{Next}_{(p)}) = d^F(x_p)$. This means that $D_1 = \text{Next}_{(p)}$ and

$\beta(D_1) = d^F(x_p)$ in this case.

• Finding D_j, $2 \le j \le (q-1)$

The following two cases should be considered.

Case 1. $l_j \notin (C_j - C_{j-1})$

In this case, the induced subgraph by the vertices in $(C_j - C_{j-1})$ forms a connected component. Meanwhile, all vertices in $(C_j - C_{j-1})$ only belong to C_j. This implies that $Q \subseteq (C_j - C_{j-1})$, for all connected p-centers Q such that at least one vertex in $(C_j - C_{j-1})$ belongs to Q. For example, in Fig. 3, $l_4 = v_{10} \notin (C_4 - C_3) = \{v_6, v_7, v_8\}$ and v_6, v_7, and v_8 only belong to C_4. It is also easy to verify that H is a p-vertex subset of $(C_j - C_{j-1})$ iff H forms a connected p-center of G such that at least one vertex in $(C_j - C_{j-1})$ belongs to H. Meanwhile, their β values will be the same. Therefore, D_j is just any p-vertex subset of $(C_j - C_{j-1})$ and we can easily check that $\beta(D_j) = \max\{d^B(f_j), d^F(l_j)\} + 1$. If $p > |C_j - C_{j-1}|$, then no solution exist in this situation and we simply set D_j = NULL and assume that β(NULL) = ∞.

Case 2. $l_j \in C_j - C_{j-1}$

Using the similar reasoning used in the description of Finding D_1, the following two cases should be considered.

Case 2.1 There exists $1 \le k \le p$ such that $\text{Next}_{(k)} = \{x_1, \ldots, x_k\}$ and $x_k \in C_q$.

In this case, C_t contains at least one vertex in $\text{Next}_{(k)}$, $j \le t \le q$. Clearly, $D_j = \text{Next}_{(k)} \cup H$, where H contains any $(p-k)$ vertices in $(C_{[j,q]} - C_{j-1}) - \text{Next}_{(k)}$ and $\beta(D_j) = d^B(f_j) + 1$, if $(C_{[j,q]} - C_{j-1})$ contains at least p vertices. Otherwise, we simply set D_j = NULL.

Case 2.2 $\text{Next}_{(p)} = \{x_1, \ldots, x_p\}$ with $x_p \in C_t$ and $t < q$.

Let Q be any connected p-center of G such that Q contains at one vertex in C_j. Then, $\beta(Q) \ge \beta(\text{Next}_{(p)}) = \max\{d^B(f_j) + 1, d^F(x_p)\}$. This means that $D_j = \text{Next}_{(p)}$ and $\beta(D_j) = \max\{d^B(f_j) + 1, d^F(x_p)\}$ in this case.

• Finding D_q

This case means that $D_q \subseteq (C_q - C_{q-1})$. Obviously, D_q can be any p-vertex subset of $(C_q - C_{q-1})$ and we can easily check that $\beta(D_j) = d^B(f_j) + 1$. If $p > |C_q - C_{q-1}|$, then no solution exist and we set D_j = NULL.

The above reasoning can establish the following lemma consequently.

Lemma 4: All $\beta(D_j)$, $1 \le j \le q$, can be computed in $O(q)$-time, if $d^B(f_j)$ and $d^F(l_j)$, $1 \le j \le q$, have been computed.

Combining all reasoning so far, the following algorithm can be designed for solving the C*p*C problem on interval graphs with unit edge-lengths correctly.

Algorithm C*p*C_Interval

Input: A positive integer p and an interval graph G with $w_1(\text{v}) = 1$, for all $v \in V$ and $w_2(e) = 1$, for all $e \in E$.

Output: A connected p-Center D such that $\beta(D)$ is minimized.

Step 1: find the clique path P of G;

/* Assume that P has q clique nodes $C_1, \ldots, C_q$.

Step 2: compute $d^B(f_j)$ and $d^F(l_j)$, for all $1 \le j \le q$;

Step 3: compute $\beta(D_1), \ldots, \beta(D_q)$ to obtain $\beta(D_{\min})$;

Step 4: $D = D_{\min}$;

Step 5: return (D);

End Algorithm C*p*C_Interval

Theorem 2: The C*p*C problem on interval graphs with unit edge-lengths can be solved in $O(n)$-time, under the condition that the clique path (with q cliques) is given.

Proof: Step 1 and Step 2 can be viewed as the preprocessing phase of our algorithm and the phase can be done in $O(m + n)$ time. The reasoning so far implies that Step 3 can be done $O(q)$ time. Finally, Step 4 just take $O(p)$-time to generate D. Since $q \leq (n - 2)$, the worst case time-complexity of Step 3 and Step 4 is $O(p + q) = O(n)$. Thus, this theorem holds. ■

IV. CONCLUSIONS

This paper considered the Connected *p*-Center problem (the C*p*C problem), which is a variant of the *p*-Center problem with more practical applications. We first proved that the C*p*C problem is NP-Hard on interval graphs satisfying triangle inequalities. Then, an $O(n)$-time algorithm for the problem on interval graphs with unit vertex-weights and unit edge-lengths was proposed under the assumption that the clique path P (with q cliques) of the input interval graph G is given.

Consider a unit edge-lengths complete graph G with $w_1(v)$ can be any positive number, for all vertices v. Obviously, $D = \{x_1, \ldots, x_p\} \subseteq V$ forms an optimal solution in this case iff $w_1(x_1)$, …, $w_1(x_p)$ are the first p largest numbers among $\{w_1(v) \mid v \in V\}$. It seems that our algorithm can be easily extended to handle the C*p*C problem on interval graphs in the same situation. Designing approximation algorithms for the C*p*C problem on interval graphs with arbitrary edge lengths is also a very interesting and important research issue in the future.

ACKNOWLEDGEMENT

This research was supported by National Science Council, Taiwan, R.O.C., under the contract number NSC 97-2221-E-128-002

REFERENCE

[1] T. C. E. Cheng, L. Kang, and C. T. Ng, An improved algorithm for the *p*-center problem on interval graphs with unit lengths, *Computers & Operations Research*, Vol. 34, pp. 2215-2222, 2007.

[2] M. S. Daskin, What you should know about location modeling, *Naval Research Logistics*, Vol. 55, pp. 283-294, 2008.

[3] M. S. Daskin, *Networks and Discrete Location, Models, Algorithms, and Applications*, John Wiley & Sons, Inc., New York, 1995.

[4] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs, Pacific Journal of Mathematics, Vol. 15, pp. 835-855, 1965.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*, Bell Laboratories, Murray Hill, Freeman & Co., N. J, 1978.

[6] F. Gavril, The intersection graphs of subtrees in tree are exactly the chordal graphs, Journal of Combinatorial Theory Series B, Vol. 16, pp. 47-56, 1974.

[7] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[8] C. G. Kekkerkerker and J. CH. Boland, Representation of a finite graph by a set of intervals on the real line, *Fundament Mathematicae*, Vol. 51, pp. 45-64, 1962.

[9] C. S. ReVelle, H. A. Eiselt, and M. S. Daskin, A bibliography for some fundamental problem categories in discrete location science, *European Journal of Operational Research*, Vol. 184, pp. 817-848, 2008.

[10] D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM Journal on Computing*, Vol. 5, pp. 266-283, 1976.

[11] B. C. Tansel, R. L. Francis, and T. J. Lowe, Location on networks: a survey. Part I: The *p*-center and *p*-median problems, *Management Science*, Vol. 29, pp. 482-497, 1983.

[12] W. C-K Yen and C-T Chen, The *p*-center problem with connectivity constraint, *Applied Mathematical Sciences*, Vol. 1, pp. 1311-1324, 2007.