Tamsui Oxford Journal of Information and Mathematical Sciences **27(2)** (2011) 183-195
Aletheia University

The Conditional Covering Problem on Interval Graphs with Unequal Costs *

Akul Rana†

Department of Mathematics, Narajole Raj College, Narajole, Paschim Medinipur- 721 211, India

Anita Pal‡

Department of Mathematics, National Institute of Technology Durgapur, Durgapur-713 209, India

and

Madhumangal Pal§

Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore-721 102, India

Received November 18, 2009, Accepted July 30, 2010.

Abstract

The conditional covering problem is an extension of classical set covering problem. The classical set covering problem finds a minimum cost covering set that covers all the vertices of the graph. The conditional covering problem has the same objective with an additional condition

*2000 *Mathematics Subject Classification*. Primary 05C85, 68Q25, 68W40.

†Corresponding author. E-mail: rana_akul@rediffmail.com

‡E-mail: anita.buie@gmail.com

§E-mail: mmpalvu@gmail.com

that every vertex in the covering set must be cover by at least one other vertex in the covering set. This problem is known to be NP-hard for general graphs. In some special cases, polynomial results are known. In this paper, an $O(n^2)$ time algorithm is presented to compute the minimum cost conditional covering set of an interval graph with n vertices. Here it is assumed that the costs of the vertices are unequal. The dynamic programming approach is used to solve the problem.

Keywords and Phrases: *Design of algorithms, Analysis of algorithms, Interval graph, Set covering, Conditional covering.*

1. Introduction

Let $G = (V, E)$ be a simple graph, i.e., finite, undirected, loop less without multiple edges, with vertex set $V = \{1, 2, \cdots, n\}$ and edge set E of size m. We use $d(x, y)$ to denote the *shortest distance* between every pair of vertices $x, y \in V$. A real number $c(v)$ is associated with each vertex v of V, called *cost* of the vertex v. The cost of a set of vertices is the sum of the costs of the vertices in the set. For any set X of vertices, the cost of X is denoted by $c(X)$, i.e., $c(X) = \sum_{v \in X} c(v)$. Each vertex $v \in V$ provides a positive real number $R(v)$ such that v can cover all vertices within the distance $R(v)$ except the vertex v. $R(v)$ is said to be the *coverage radius* of the vertex v. A vertex covers all vertices within its coverage radius except itself (no vertex can cover itself), i.e., the vertex $x \in V$ is covered by a vertex $y \in V (x \neq y)$, if $d(x, y) \leq R(y)$ holds. A set of vertices X is said to be a *covering set* if the set of vertices in X covers all the vertices of the graph G. A *minimum covering set* is a covering set of minimum cost. The conditional covering problem (CCP, for short) is to find a minimum cost covering set which covers all the vertices of the graph. We refer this minimum cost covering set as *minimum conditional covering set* (MCCS). From this definition, it is observed that the cardinality of MCCS is at least two.

Although, the CCP is NP-complete for general graphs, there exists some spacial graphs which can be solved in polynomial time. In this paper, we consider a special case of the CCP on interval graphs. Here, we consider the vertex costs are non-uniform, coverage radius is uniform and all edge weights are unity.

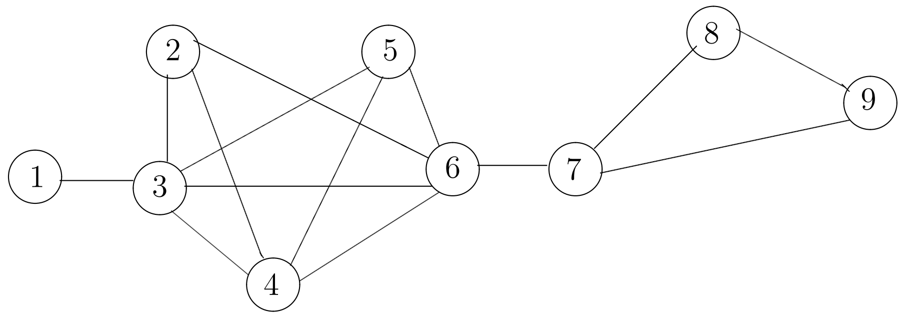

Figure 1: An interval graph.

1.1. Interval graph

An undirected graph $G = (V, E)$ is an interval graph if there exists a one-to-one correspondence between the vertex set V and a set of intervals I on the real line such that two vertices in V are adjacent in G if and only if their corresponding intervals intersect.

Let $I = \{I_1, I_2, \ldots I_n\}$ where $I_j = [a_j, b_j]$, $j = 1, 2, 3, \ldots n$; be the *interval representation* of a connected interval graph $G = (V, E)$, $V = \{1, 2, \ldots n\}$, a_j and b_j are respectively left and right end points of the interval I_j. The vertex j corresponds to the interval I_j. Without any loss of generality, we assume that each interval contains both its end points and that no two intervals share a common end point. Also, we assume that the intervals in I are indexed by increasing right end points, that is, $b_1 < b_2 < \cdots < b_n$. This indexing known as interval graph (IG) ordering. An interval $I_x = [a_x, b_x]$ is to the *left* of the interval $I_y = [a_y, b_y]$, if $b_x < b_y$ and I_x is to the *right* of the interval I_y, if $b_x > b_y$. If $b_x < b_y$, then we write $x < y$.

An interval graph and its interval representation are shown in Figure 1 and 2 respectively.

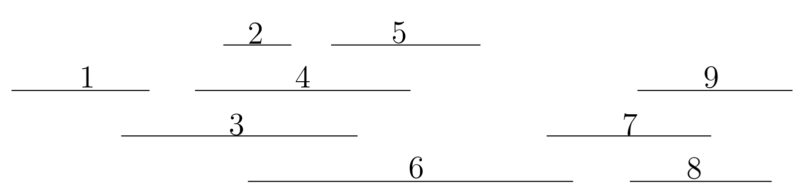

Figure 2: An interval representation of the interval graph of Figure 1.

Intervals and vertices of an interval graph are the same. The set I is called an *interval representation* of G. An interval representation with sorted end points of an interval graph can be constructed in $O(n+m)$ time [1]. Interval graphs are discussed extensively in [5].

In Figure 1, if the cost of the vertices 1, 2, 3, 4, 5, 6, 7, 8, 9 are 5, 4, 3, 8, 10, 5, 5, 7, 9 respectively and if, we take coverage radius $R = 2$, then it is easy to verify that the set $\{5, 9\}$ is a conditional covering set. The set $\{4, 7\}$ is also a conditional covering set, but none of them is a MCCS. We can verify that $\{3, 7\}$ is a MCCS. Also, $\{3, 6\}$ is another MCCS. From this example, it is observed that MCCS for a graph may not be unique.

1.2. Application of interval graph and CCP

The CCP occurs in several practical planning problems. One set of applications supports the situation when a facility experiences a failure and requires coverage from a backup facility. The application area of the CCP include locating facilities in distribution systems, emergency systems, communication systems and energy supply systems. Consider the problem of locating rescue centers in a country. Each potential site is associated with a covering radius and a cost of locating a facility there. In case of a disaster, no rescue center can help the site at which it is located. Hence every rescue center should be cover by another rescue center. This problem can be modelled as CCP on a graph. Another motivating example is the location of Weapons of Mass Destruction Civil Support Teams in the United States [11]. The teams are located to quickly respond to biological or chemical terrorist attacks in major cities. The teams does not necessarily cover the city at which a team is located since a biological or chemical terrorist attack may render the team incapable of performing its mission.

Interval graphs arise in the process of modelling of real life situations, specially involving time dependencies or other restrictions that are linear in nature. The graphs and various subclasses thereof arise in diverse areas such as archeology, molecular biology, sociology, genetics, traffic planning, VLSI design, circuit routing, psychology, scheduling, transportation and others.

1.3. Survey of the related works

In the literature, Moon and Chaudhry [13] were the first to address CCP as constrained facility location model. They present an integer programming model for this problem. This study was followed in a sequence of papers [2, 3, 10]. Moon and Papayanopoulos [14] consider one variation of CCP on trees and presented a linear time algorithm. These authors consider the graph that contains uniform facility cost, each vertex has a radius in which a facility must be located and potential facility locations can exist at places other than the vertices of the graph. Lunday *et al.* [11] introduced a modified version of the CCP. In modified version of CCP, a facility can cover all vertices within its coverage radius except itself and potential facility locations are confined to the vertices of the graph. In this paper, we study this modified version. For the CCP on a path with uniform coverage radii, Lunday *et al.* [11] present a linear time algorithm to optimally solve the unweighted cost CCP and an $O(n^2)$ time dynamic programming algorithm to solve the weighted cost CCP. Horne and Smith [7] studied the weighted cost CCP on path and extended star graphs with nonuniform coverage radius and developed an $O(n^2)$ time dynamic programming algorithm. In an another paper, Horne and Smith [8] consider weighted cost CCP on the tree graphs and presented an $O(n^4)$ time dynamic programming algorithm. Recently, Rana *et al.* [16] propose an $O(n)$ time algorithm to solve the CPP on interval graphs with uniform coverage radius. In another paper [17], these authors presented an $O(n^2)$ time algorithm for the CCP on interval graphs with non uniform coverage radius.

1.4. Our result

In this paper, we consider a special case of CCP on interval graphs where vertex weights are unequal, coverage radii are fixed positive integers and all edge weights are 1. For this case, we propose an $O(n^2)$ time algorithm for finding minimum cost conditional covering set in interval graphs where n indicates the number of vertices in the graph. We have used dynamic programming approach to solve the problem.

1.5. Organization of the paper

The rest of this paper is organized as follows. In section 2, preliminaries and notations are given. Also, some results are presented in this section which provides the basis for the algorithm. Section 3 develops an $O(n^2)$ time dynamic programming algorithm for solving CCP on an interval graph. The time complexity is also calculated in this section. Finally, in section 4, we give some concluding remarks.

2. Preliminary

Ramalingam and Pandurangan have proved the following useful lemma for an interval graph.

Lemma 1. *[15] A graph $G = (V, E)$ with $|V| = n$ is an interval graph iff its vertices can be numbered from 1 to n such that for $i < j < k$, if (i, k) is an edge in the graph then (j, k) is an edge in the graph, i.e., $(i, k) \in E \Rightarrow (j, k) \in E$.*

Using this lemma, we prove the following result.

Lemma 2. *Let j be a vertex such that $i < j < k$ and if $d(i, k) \leq R$ then $d(k, j) \leq R$.*

Proof. There are two cases may arise, $(i, k) \in E$ or $(i, k) \notin E$.

Case 1: $(i, k) \in E$. Then by Lemma 1, $(j, k) \in E$. Therefore $d(k, j) \leq R$.

Case 2: $(i, k) \notin E$. In this case, there must exist a sequence of vertices $t_1 < t_2 < \cdots < t_m$ in between i and k which forms the path $i \to t_1 \to t_2 \cdots \to t_m \to k$. There are two subcases may arise:

Subcase 1: $t_i = j$ for some $i = 1, 2, \ldots, m$. Then, obviously $d(k, j) \leq R$.

Subcase 2: $t_i \neq j$ for any $i = 1, 2, \ldots, m$. In this case, there must exist at least one t_i such that $(t_i, j) \in E$. Therefore, $d(k, j) \leq R$. □

Let us introduce some notations which are used in the rest of this paper.

Let A_i be the set of vertices within the coverage radius of the vertex i, i.e., $A_i = \{j : d(i, j) \leq R\}$ and $B_i = A_i \cup \{i\}$. Define *lower reach* $l(i)$ of the vertex i as the smallest index vertex that lies within the coverage radius of i, i.e., $l(i) = min\{j : j \in B_i\}$ and $ml(i) = max\{l(j) : l(i) \leq j \leq i\}$.

We construct two sets P_i and Q_i as follows

$P_i = \{ml(i), ml(i) + 1, \cdots, i\}$ and $Q_i = \{j : j > i \text{ and } d(i, j) \leq R\}$.

It should be noted that for $1 \leq i \leq n - 1$, Q_i is always non empty, but Q_n is empty.

For any non empty set X of vertices, Min$\{X\}$ denote a covering set of X that has minimum cost. Clearly, Min$\{X\}$ is a set of vertices which may or may not be a subset of X.

For any non empty set of vertices X, $mc(X)$ denote the cost of minimum cost covering set of X, i.e., $mc(X) = c(Min\{\text{X}\})$.

If $X = \phi$ then Min$\{X\}$ denote a set of infinite cost, i.e., if $X = \phi$ then $mc(X) = \infty$.

From above definitions, we observe that only the vertices of the set A_i can cover the vertex i.

Following the above definitions, we can prove the following two lemmas easily.

Lemma 3. *Let j be a vertex such that $l(i) \leq j \leq i$ and $ml(i) = l(j)$ then $B_j \subseteq P_i \cup Q_i$.*

Proof. We have, $B_j = \{j\} \cup A_j$ and A_j is the set of all vertices within the coverage radius of the vertex j. So, the minimum index vertex in the set B_j is $l(j)$ and maximum index vertex is greater than j and lies within the coverage radius of j. Here $ml(i) = l(j)$, so $P_i = \{l(j), l(j)+1, \cdots, i\}$. Therefore, P_i contains the vertex j. Q_i is the set of all vertices greater than i and lies within the coverage radius of i. Since $l(i) \leq j \leq i$, therefore Q_i contains the maximum index vertex of B_j. Hence $P_i \cup Q_i$ contains all the vertices of B_j, i.e., $B_j \subseteq P_i \cup Q_i$. □

Lemma 4. *For any two vertices $x, y \in P_i$, $d(x, y) \leq R$, for any i.*

Proof. Recall that $P_i = \{ml(i), ml(i)+1, \cdots, i\}$ and $ml(i) = max\{l(j) : l(i) \leq j \leq i\}$. It follows that, for any $x \in P_i$, $l(i) \leq x \leq i$, $ml(i) \geq l(x)$. Clearly, $d(x, ml(i)) \leq R$. Then by Lemma 2, $d(x, y) \leq R$ for any two vertices $x, y \in P_i$. □

3. The Algorithm

Let $V_i = \{1, 2, \cdots, i\}$ and C_i be a subset of V that covers V_i. MC_i denote the MCCS of V_i.

Lemma 5. *In each C_i, there must be at least one vertex from the set $P_i \cup Q_i$.*

Proof. We prove this lemma by contradiction. If possible, let there exists a C_i which does not contain any vertex of $P_i \cup Q_i$. Let j be a vertex such that $l(i) \leq j \leq i$ and $ml(i) = l(j)$. Then $j \in V_i$. The vertices which lie within the distance R can cover the vertex j, i.e., j can only be covered by the vertices of the set A_j. From Lemma 3, it follows that $A_j \subset P_i \cup Q_i$. Since C_i does not include any vertex of $P_i \cup Q_i$, it does not include any vertex of A_j. Then j remains uncovered. But $j \in V_i$ and C_i is a subset of V that covers V_i, which is a contradiction. Hence the lemma. □

From above lemma, it is clear that, there must be one vertex, say, k, in C_i, from the set $P_i \cup Q_i$. Then $k \in P_i$ or $k \in Q_i$.

If $k \in P_i$, let M_1C_i be the minimum cost covering set of V_i and if $k \in Q_i$, let M_2C_i be the minimum cost covering set of V_i.

If $k \in P_i$, then the vertex k covers the set $\{l(k), l(k)+1, \cdots, i\}$ except itself. Then to find a C_i, it is necessary and sufficient that the set $C_i/\{k\}$ covers the set $\{k\} \cup V_{l(k)-1}$.

Let D_i be a subset of V that covers $\{i\} \cup V_{l(i)-1}$ and MD_i be a minimum cost D_i.

The following two lemmas are the backbone of the algorithm presented here.

Lemma 6. *Let $k \in P_i$. If $k \in C_i$ then M_1C_i =Min$\{\{k\} \cup MD_k\}$ where MD_i =Min$\{\{j\} \cup MC_{min\{l(i)-1, l(j)-1\}} : j \in A_i\}$.*

Proof. Since $k \in P_i$ and $k \in C_i$, the set of vertices covered by k is a super set of $\{l(k), l(k)+1, \cdots, i\} - \{k\}$. Then to find a C_i, it is required to cover the set $\{1, 2, \cdots, l(k)-1\} \cup \{k\} = V_{l(k)-1} \cup \{k\}$. But, D_i is a subset of V that covers $V_{l(i)-1} \cup \{i\}$. Therefore, $C_i = \{k\} \cup D_k$ and hence for $k \in P_i$, M_1C_i =Min$\{\{k\} \cup MD_k\}$.

Now, D_i is a subset of V that covers $V_{l(i)-1} \cup \{i\}$, D_i must include some vertex t within the coverage radius of i, i.e., $t \in D_i$ such that $t \in A_i$. If the interval I_t lies to the left of the interval I_i then $l(i) > l(t)$ and to find a D_i, it is necessary and sufficient to cover the set $V_{l(t)-1}$. If the interval I_t lies to the right of the interval I_i, then it is required to cover the set $V_{l(i)-1}$. That is, $D_i = \{\{t\} \cup V_{min\{l(i)-1, l(t)-1\}} : t \in A_i\}$.

Therefore, we have the following recurrence relations

$MD_i = Min\{\{k\} \cup MC_{min\{l(i)-1, l(t)-1\}} : t \in A_i\}$ and

$M_1C_i = Min\{\{k\} \cup MD_k : k \in P_i\}$. □

Lemma 7. *Let $k \in Q_i$. If $k \in C_i$ then M_2C_i =Min$\{\{k\} \cup MC_{l(k)-1}\}$.*

Proof. If $k \in Q_i$, then the interval I_k lies to the right of the interval I_i, i.e., $k \notin V_i$. The vertex k will cover the set $\{l(k), l(k)+1, \cdots, i\}$. Since $k \notin V_i$, to find a C_i, it is needed that the set $C_i/\{k\}$ covers the set $\{1, 2, \cdots, l(k)-1\} = V_{l(k)-1}$. Therefore, for $k \in Q_i$, we have the following recurrence relation $M_2C_i = Min\{\{k\} \cup MC_{l(k)-1} : k \in Q_i\}$. □

The algorithm proceeds by covering the vertices of the graph from left to right in the interval representation. At each stage $i (1 \leq i \leq n-1)$, we compute M_1C_i and M_2C_i. For $1 \leq i \leq n-1$, if $mc(M_2C_i) > mc(M_1C_i)$, then we take $MC_i = M_1C_i$, otherwise $MC_i = M_2C_i$. For $i = n$, $MC_n = M_1C_n$.

A formal description of the algorithm is given below.

Algorithm MCCS
Input: A set of sorted intervals of an interval graph $G = (V, E)$.
Output: A minimum conditional covering set MC_n in G.
Initially $MC_0 = \Phi$ (empty set) and $mc(MC_0) = 0$.
Step 1: Compute the arrays $l(i), ml(i)$ for each vertex $i \in V$.
Step 2: Compute the sets P_i, Q_i for each vertex $i \in V$.
Step 3: **For** $i = 1$ to $n-1$ do

```
        cost1= ∞, cost2= ∞, cost3= ∞;
    // Compute MD_i by lemma 6 //
    For all j ∈ A_i
        If j ∈ A_i and c(j) + mc(MC_{min{l(j)−1,l(i)−1}}) < cost1 then
            cost1= c(j) + mc(MC_{min{l(j)−1,l(i)−1}});
            t = j;
        endif;
    endfor;
    MD_i = {t} ∪ MC_{min{l(t)−1,l(i)−1}};
    // Compute M_1C_i by lemma 6 //
    For all j ∈ P_i
        If j ∈ P_i and c(j) + mc(MD_j) < cost2 then
            cost2= c(j) + mc(MD_j);
            t = j;
        endif;
    endfor;
    M_1C_i = {t} ∪ MD_t;
    // Compute M_2C_i by lemma 7 //
```

For all $j \in Q_i$
  If $j \in Q_i$ and $c(j) + mc(MC_{l(j)-1}) < cost3$ then
    cost3= $c(j) + mc(MC_{l(j)-1})$;
    $t = j$;
  endif;
endfor;
$M_2C_i = \{t\} \cup MC_{l(t)-1}$;
If $mc(M_2C_i) > mc(M_1C_i)$ then
  $MC_i = M_1C_i$;
else
  $MC_i = M_2C_i$;
endif;
end for;

Step 4: MD_n =Min$\{\{j\} \cup MC_{min\{l(j)-1,l(n)-1\}} : j \in A_n\}$;
  MC_n =Min$\{\{j\} \cup MD_n : j \in P_n\}$;

end MCCS

To illustrate our methodology, we consider an interval graph of seven vertices with $R = 2$ shown in Figure 3. The numbers within the boxes denote the cost of the corresponding vertex.

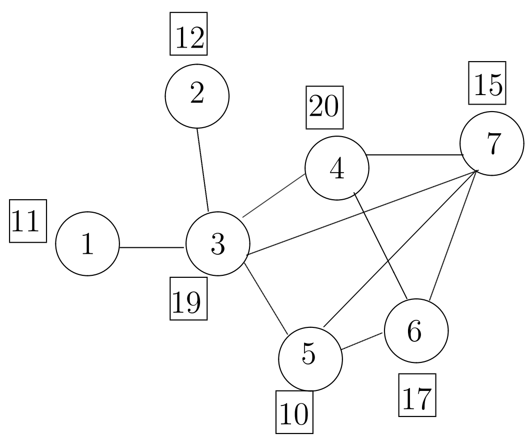

Figure 3: An example

Table 1: Illustration of the methodology on the CCP instance given in Figure 3.

	A_i	P_i	Q_i	MD_i	M_1C_i	M_2C_i	MC_i
$i=1$	$\{2,3,4,5,7\}$	$\{1\}$	$\{2,3,4,5,7\}$	$\{5\}$	$\{1,5\}$	$\{5\}$	$\{5\}$
$i=2$	$\{1,3,4,5,7\}$	$\{1,2\}$	$\{3,4,5,7\}$	$\{5\}$	$\{1,5\}$	$\{5\}$	$\{5\}$
$i=3$	$\{1,2,4,5,6,7\}$	$\{1,2,3\}$	$\{4,5,6,7\}$	$\{5\}$	$\{1,5\}$	$\{5\}$	$\{5\}$
$i=4$	$\{1,2,3,5,6,7\}$	$\{1,2,3,4\}$	$\{5,6,7\}$	$\{5\}$	$\{1,5\}$	$\{5\}$	$\{5\}$
$i=5$	$\{1,2,3,4,6,7\}$	$\{1,2,3,4,5\}$	$\{6,7\}$	$\{1\}$	$\{1,5\}$	$\{7\}$	$\{7\}$
$i=6$	$\{3,4,5,7\}$	$\{3,4,5,6\}$	$\{7\}$	$\{5\}$	$\{1,5\}$	$\{7\}$	$\{7\}$
$i=7$	$\{1,2,3,4,5,6\}$	$\{3,4,5,6,7\}$	ϕ	$\{5\}$	$\{1,5\}$	$-$	$\{1,5\}$

Therefore, the set $\{1,5\}$ is a MCCS of the graph of Figure 3.

4. Proof of Correctness of the Algorithm and its Time Complexity

We show by induction that MC_i being a minimum cost covering set of V_i.

As a basis step consider the computation of MC_1. Since, $V_1 = \{1\}$ and $P_1 = \{1\}$, any minimum cost covering set for V_1 must contain a vertex from Q_1. By choosing a vertex from Q_1 of minimum cost, the algorithm ensures that MC_1 is correctly computed.

Let us assume that MC_{i-1} be a minimum cost covering set of V_{i-1}. If MC_{i-1} covers the vertex i then the algorithm sets MC_i to MC_{i-1}. Since $V_{i-1} \subseteq V_i$, no minimum cost covering for V_i can have minimum cost than that for V_{i-1}. If the vertex i is not covered by the set MC_{i-1}, then we compute M_1C_i and M_2C_i. If $mc(M_2C_i) > mc(M_1C_i)$ then $MC_i = M_1C_i$, otherwise $MC_i = M_2C_i$. At nth stage, if MC_{n-1} covers the vertex n, then the algorithm sets MC_n to MC_{n-1}. Otherwise, since $Q_n = \phi$, $MC_n = M_1C_n$ is a minimum cost covering set for V_n. Therefore the algorithm has correctly computed MCCS for $V_n = G$. □

Theorem 1. *Algorithm MCCS finds a minimum cost conditional covering set on interval graph in $O(n^2)$ time.*

Proof. Computation of the sets A_i can be done in $O(n^2)$ time. Also, computation of arrays $l(i)$ and $ml(i)$ can be done in $O(n^2)$ time. The algorithm consist of n stages. Stage i required $O(A_i) + O(P_i) + O(Q_i) = O(B_i)$ time.

Therefore, the total time is $\sum_{i=1}^{n} \mid B_i \mid = \sum_{i=1}^{n} O(n) = O(n^2)$. Hence overall time complexity is $O(n^2)$. □

5. Concluding Remarks

In this article, we examine the CCP on interval graphs with non-uniform cost and present an algorithm which runs in $O(n^2)$ time. Here, we consider uniform edge weights, i.e., $d(u, v) = 1$ for all $(u, v) \in E$. Unfortunately, the algorithm presented here does not appear to be extendable to the CCP with non-uniform edge weights. It would be interesting to find a polynomial time algorithm for interval graphs with non uniform cost and edge weights. In our future study, we examine one of several practical variations of the CCP and examine the CCP on various other graphs other than the ones studied. □

References

[1] K. S. Booth and G. S. Lueker, Testing for the cosecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, *J. Comput. System Sci.*, **13**(1976), 191-199.

[2] S. S. Chaudhry, New heuristics for the conditional covering problem, *Opsearch*, **30**(1993), 42-47.

[3] S. S. Chaudhry, I. D. Moon, and S. T. McCormick, Conditional covering: Greedy heuristics and computational results, *Comput. Oper. Res.*, **14**(1987), 11-18.

[4] E. J. Cockayne, R. M. Dawes, and S. T. Hedetniemi, Total domination in graphs, *Networks*, **11**(1980), 211-219.

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

[6] S. M. Hedetniemi, E. J. Cockayne, and S. T. Hedtniemi, Linear algorithm for finding the Jordan centre and path centre of a tree, *Transportation Sci.*, **98**(1981), 98-114.

[7] J. A. Horne and J. C. Smith, Dynamic programming algorithm for the conditional covering problem on path and extended star graph, *Networks*, **46** no.4 (2005), 177-185.

[8] J. A. Horne and J. C. Smith, A dynamic programming algorithm for the conditional covering problem on tree graphs, *Networks*, **46** no.4 (2005), 186-197.

[9] J. M. Keil, Total domination in interval graphs, *Information Processing Letters*, **22**(1986), 171-174.

[10] V. Lotfi and I. D. Moon, Hybrid heuristics for conditional covering problem, *Int. J. Modeling Simulation*, **17**(1997), 185-190.

[11] B. J. Lunday, J. C. Smith, and J. B. Goldberg, Algorithms for solving the conditional covering problem on paths, *Naval Res. Logistics*, **52**(2005), 293-301.

[12] S. Mondal, M. Pal, and T. K. Pal, An optimal algorithm to solve 2-neighbourhood covering problem on interval graphs, *Intern. J. Computer Math.*, **79**(2002), 189-204.

[13] I. D. Moon and S. S. Chaudhry, An analysis of network location problems with distance constraints, *Mang. Sci.*, **30**(1984), 290-307.

[14] I. D. Moon and L. Papayanopoulos, Facility location on a tree with maximum distance constraints, *Comput. Oper. Res.*, **22**(1995), 905-914.

[15] G. Ramalingam and C. Pandu Rangan, A unified approach to domination problems in interval graphs, *Information Processing Letters*, **27**(1988), 271-274.

[16] A. Rana, A. Pal, and M. Pal, The conditional covering problem on unweighted interval graphs, *Journal of Applied Mathematics and Informatics*, **28**(2010), 1-11.

[17] A. Rana, A. Pal, and M. Pal, The conditional covering problem on interval graphs with non-uniform coverage radius, *communicated*.