J. Appl. Math. & Informatics Vol. 28(2010), No. 1 - 2, pp. 1 - 11
Website: http://www.kcam.biz

THE CONDITIONAL COVERING PROBLEM ON UNWEIGHTED INTERVAL GRAPHS

AKUL RANA, ANITA PAL AND MADHUMANGAL PAL*

ABSTRACT. The conditional covering problem is an important variation of well studied set covering problem. In the set covering problem, the problem is to find a minimum cardinality vertex set which will cover all the given demand points. The conditional covering problem asks to find a minimum cardinality vertex set that will cover not only the given demand points but also one another. This problem is NP-complete for general graphs. In this paper, we present an efficient algorithm to solve the conditional covering problem on interval graphs with n vertices which runs in $O(n)$time.

AMS Mathematics Subject Classification : 35J60, 35J40.
Key word and phrases : large solutions, blow-up rate, uniqueness; sub-supersolutions.

1. Introduction

The conditional covering problem (CCP, for short) is a facility location problem on a graph. Let $G = (V, E)$ be a graph where $V = \{1, 2, \ldots, n\}$ is the set of vertices and E is the set of edges. The vertex set V of the graph represents the set of demand points as well as the set of potential facility locations. A weight $w(e)$ is associated with every edge $e \in E$. The length of a path is the sum of the weights of the edges in the path. A path from the vertex x to the vertex y is a shortest path if there is no other path from x to y with lower length. We use $d(x, y)$ to denote such a shortest path. For a facility located at a vertex $x \in V$ requires a facility location cost $c(x)$ and provide a positive coverage radius $R(x)$. A facility can cover all vertices within its coverage radius except the vertex at which it is located, that is, a vertex $x \in V$ is covered by a facility located at a vertex $y \in V$ if $x \neq y$ and $d(x, y) \leq R(x)$. The CCP seeks to minimize facility

Received November 5, 2008. Revised July 8, 2009. Accepted July 28, 2009. *Corresponding author.

location cost such that the set of vertices in the graph must be covered by a facility and every facility should be covered by at least one another facility. Here each facility has a specified, possibly overlapping region to serve.

The graphs considered in this paper are finite, connected, undirected and have no self loop or parallel edges. In this paper, we take $c(x) = 1$ for all $x \in V$, $w(e) = 1$ for all $e \in E$ and $R(x) = R > 1$. Since each vertex has unit cost, our goal is to find a set of vertices with minimum cardinality which will cover all vertices and every selected vertex should be covered by another selected vertex. This set is called a conditional covering set (CCS for short). Note that, the CCS for a given graph may not be unique. It should be noted that if $R(x) = 1$ for all $x \in V$, then our problem is identical to the *total dominating set* problem.

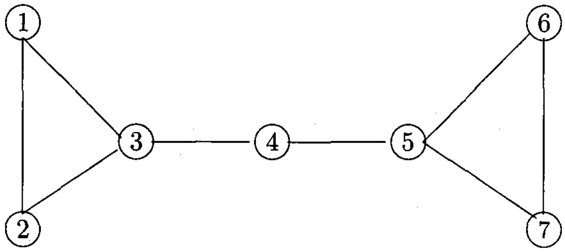

FIGURE 1. An example.

The CCP for an interval graph with seven vertices is demonstrated in Figure 1. We take uniform coverage radius $R = 2.5$. If we locate a facility at the vertex 4 then it will cover all demand points except itself. To cover the vertex 4, another facility is to be located at any other vertex. Again, if we select the vertices 3 and 5 as two facility locations then the set $\{3, 5\}$ is also a solution. So for this problem there are several solutions, viz., $\{4, 5\}$, $\{4, 6\}$, $\{4, 7\}$, $\{4, 3\}$, $\{4, 2\}$, $\{4, 1\}$ and $\{3, 5\}$.

Figure 2 shows a CCS for an interval graph which has unique solution $\{4, 6\}$ for $R = 2$.

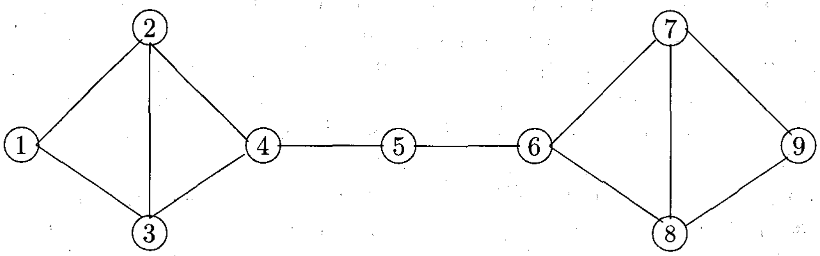

FIGURE 2.

Now, we give some basic definitions used in this paper.

Definition 1 (Interval Graph). An undirected graph $G = (V, E)$ is an interval graph if the vertex set V can be put into one to one correspondence with a set of intervals I on the real line R such that two vertices are adjacent in G if and only if their corresponding intervals have non empty intersection.

Intervals and vertices of an interval graph are the same. The set I is called an interval representation of G and G is referred to as the intersection graph of I. Interval graphs are discussed extensively in [4].

Definition 2 (Total Dominating Set). A set of vertices D in V is a total dominating set for a graph $G = (V, E)$ if every vertex in V is adjacent to a vertex in D.

1.1. Application of interval graph and CCP. Interval graphs arise in the process of modeling 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. The CCP occurs in several practical planning problems. The application area of the CCP include locating facilities in distribution systems, emergency systems, communication systems and energy supply systems. When it is possible that a facility experiences a failure, the existence of one or more other facilities that can be used as backups.

1.2. Survey of the related works. The problem studied in this paper is closely related to the total dominating set problem. In the total dominating set problem all edge weights, coverage radii and facility location costs are equal to 1. The total dominating set problem was introduced by Cockayane *et al.* [3]. They investigate properties of the smallest total dominating set and of the largest number of sets in a partition of the vertex set V into total dominating set. 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 problem has received strong interest since Moon and Chaudhry [13] published the first paper on this topic. This study was followed in a sequence of papers [1, 2, 9]. Moon and Papayanopoulos [14] consider one variation of CCP on trees. These authors consider the graph that contains uniform facility cost and each demand point has a radius in which a facility must be located. The CCP as defined in our work consider a set of facilities that covers all demand points within their coverage radius, except the vertex at which it is located. In our problem, potential facility locations are confined to the vertices of the graph. Complexity issues regarding the placement of facilities in CCP were considered by Lunday *et al.* [10] and Horne and Smith [6, 7]. For the CCP on a path graph with uniform coverage radii, Lunday *et al.* [10] present a linear time algorithm to optimally solve the unweighted cost CCP and an $O(n^2)$ dynamic programming algorithm to solve the weighted cost CCP. Horne and Smith[6] studied the weighted cost CCP on path and extended star graphs with nonuniform coverage radius and developed an $O(n^2)$ dynamic programming algorithm. In an another paper, Horne and Smith [7] consider weighted cost CCP on the tree graphs and presented an $O(n^4)$ dynamic programming algorithm.

1.3. Our result. To the best of our knowledge, no algorithm is available to solve the CCP on interval graphs. Since the total dominating set problem is NP-complete even on bipartite graphs [6], CCP is also NP-complete for general graphs. Using dynamic programming technique an $O(n)$ time algorithm is designed to solve the unweighted cost CCP with uniform coverage radius on interval graphs. A data structure called *interval tree* [15] is used to solve this problem.

1.4. Organization of the paper. The rest of this paper is organized as follows. In section 2, we discuss some important properties of interval graph and interval tree. In section 3, we prove some important results related to conditional covering set in interval graph for understanding the algorithm. Section 4 develops a dynamic programming algorithm for solving CCP on an interval graph. The time complexity is also calculated in this section. Finally, in section 5, we give some concluding remarks.

2. The interval tree

Let $I = \{I_1, I_2, \dots I_n\}$ where $I_j = [a_j, b_j]$, $j = 1, 2, 3, \dots n$; be the interval representation of a connected interval graph $G = (V, E)$, $V = \{1, 2, \dots n\}$, a_j and b_j are respectively left and right end points of 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.

For each vertex $u \in V$, let $H(u)$ represents the highest numbered adjacent vertex of u. If no adjacent vertex of u exists with higher number then $H(u)$ is assumed to be u. In other words, $H(u) = max\{v : (u, v) \in E \; and \; v \geq u\}$. For an interval graph $G = (V, E)$ let an interval tree (IT) with root n be defined as $T(G) = (V, E')$ where $E' = \{(u, v) : u \in V and \; v = H(u), u \neq n\}$. The IT is a spanning tree of the connected interval graph G. In [15], it is proved that for a connected interval graph there exist a unique interval tree.

For each vertex v of IT, we define level of v to be the number of edges from the vertex n to the vertex v in the interval tree, i.e.,

$$level(v) = \text{number of edges from } n \text{ to } v.$$

We define the height of the tree IT by

$$h = max\{level(v) : v \in V\}.$$

From definition of IT and its level it is clear that, level(1)= height of the tree IT. The path in interval tree from the vertex 1 to the root n is called the *main path*.

Let N_j be the set of vertices which are at a level j and N_0 is the singleton set $\{n\}$.

An interval graph and its interval representation are shown in the Figure 3.

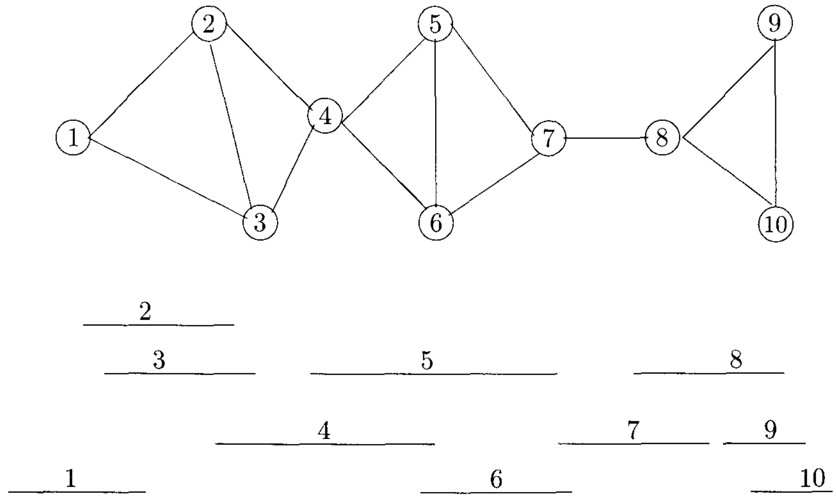

FIGURE 3. An interval graph G and its interval representation.

2.1. Some results on interval graphs and interval tree. In this section, we present some important results relating to the interval graph and interval tree. Ramalingam and Pandu Rangan [18] have shown the following result.

Lemma 1. [18] *If the vertices $u, v, w \in V$ are such that $u < v < w$ in the IG ordering and u is adjacent to w, then v is also adjacent to w.*

Lemma 2. *If the vertices $u, v, w \in V$ are such that $u > v > w$ in the IG ordering and u is not adjacent to v, then u is not adjacent to w.*

Proof. Given that, $u > v > w$ in the IG ordering and u is not adjacent to v. Therefore intervals I_u and I_v for the vertices u and v have empty intersection, i.e., $I_u \bigcap I_v = \Phi$. Since, $u > v > w$, $I_u \bigcap I_w = \Phi$. Therefore, there is no edge between u and w.

Some important properties of interval tree are studied in [15]. From the construction of IT, it is clear that if *level*(u) $>$ *level*(v) then $u < v$ in IG ordering. par

The vertices at any level of IT are consecutive integers. This is proved in [15] as the following lemma.

Lemma 3. [15] *The vertices of N_j are consecutive integers and if v is equal to $min\{u : u \in N_j\}$ then $max\{u : u \in N_{j+1}\}$ is equal to $v - 1$.*

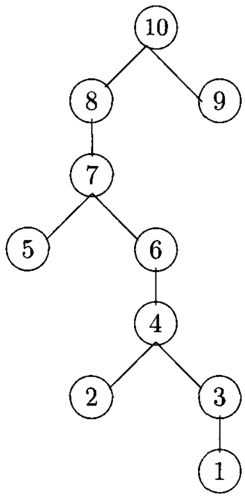

FIGURE 4. Interval tree of the interval graph of Figure 3.

If the level of a vertex v of IT is j then it should be adjacent only to the vertices at levels $j-1$, j and $j+1$ in G. This is proved in [15] as the following lemma.

Lemma 4. [15] *If $u, v \in V$ and $|level(u) - level(v)| > 1$ then there is no edge between the vertices u and v.*

The distance $d(u,v)$ between any two vertices u and v of same level is either 1 or 2, which is proved in [15] as follows.

Lemma 5. [15] *For $u, v \in V$ if $level(u) = level(v)$ then,*

$$d(u,v) = \begin{cases} 1, & (u,v) \in E. \\ 2, & otherwise. \end{cases}$$

We use the notation $u \to v$ to indicate that there is a path from the vertex u to the vertex v of length one. Now we introduce some notations which are used through out the remaining part of the paper.

l: an integer represents the level number at any stage,
u_l^*: the vertex on the main path at level l,
X_l: the set of vertices at level l of IT which are greater than u_l^*, i.e.,

$$X_l = \{v : v > u_l^* \text{ and } v \in N_l\},$$

Y_l: the set of vertices at level l of IT which are less than u_l^*, i.e.,

$$Y_l = \{v : v < u_l^* \text{ and } v \in N_l\}.$$

From the above definitions, we observed that

$$X_l \bigcap Y_l = \phi \quad \text{and} \quad N_l = X_l \bigcup Y_l \bigcup \{u_l^*\}.$$

Lemma 6. *If v be any member of X_l then $(v, u_{l-1}^*) \in E$.*

Proof. We know that, $X_l = \{v : v > u_l^* \text{ and } v \in N_l\}$. Let $v\prime$ be any member of X_l. Since, $u_l^* < v\prime < u_{l-1}^*$ for all $v\prime \in X_l$ and there exist an edge between u_l^* and u_{l-1}^*, then by Lemma 1, $(v\prime, u_{l-1}^*) \in E$. Hence the lemma.

Lemma 7. *If v be any member of Y_l then $(u_l^*, v) \in E$.*

Proof. We have, $Y_l = \{v : v < u_l^* \text{ and } v \in N_l\}$. Let $v\prime$ be any member of Y_l. Again, $u_{l+1}^* < v\prime < u_l^*$ for all $v\prime \in Y_l$. From the construction of IT, we see that there exist an edge between u_{l+1}^* and u_l^*. Hence by Lemma 1, $(v\prime, u_l^*) \in E$.

3. Some important results related to CCS

Let C be the conditional covering set and R be the uniform coverage radius of the given interval graph. Let $m = \lfloor R \rfloor$. To find a CCS, an IT of the given interval graph is to be constructed. If there exist at least one vertex of N_1 which is not adjacent to u_1^*, we select u_{m-1}^* as the first member of C, otherwise we select u_m^* as the first member of C which is proved in the following two lemmas.

Lemma 8. *If there is at least one vertex of N_1 which is not adjacent to u_1^* then u_{m-1}^* is the first member of C.*

Proof. Let v_1 be a member of N_1 which is not adjacent to u_1^*. Now, $N_1 = X_1 \cup Y_1 \cup \{u_1^*\}$ and every member of Y_1 is adjacent to u_1^* (by Lemma 7). Therefore, v_1 must be a member of X_1. Let v_i be any member of X_i, $i = 2, 3, \ldots, m-1$. Then $(v_i, u_{i-1}^*) \in E$ (by Lemma 6). Again, $H(u_i^*) < v_{i-1}$ for $i = 2, 3, \ldots, m-1$ which implies $(u_i^*, v_{i-1}) \notin E$. Therefore, $d(u_{m-1}^*, n) = m-1$ (as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_1^* \to n$) and $d(u_{m-1}^*, v_1) = m$ (as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_1^* \to n \to v_1$). For $i = 2, 3, \ldots, m-1$; if $(v_i, u_i^*) \in E$ for all $v_i \in X_i$, then $d(u_{m-1}^*, v_i) = m-i$(as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_i^* \to v_i$). If $(v_i, u_i^*) \notin E$ for at least $v_i \in X_i$, then $d(u_{m-1}^*, v_i) = m-i+1$ (as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_i^* \to u_{i-1}^* \to v_i$). Therefore, $d(u_{m-1}^*, v) \leq m$, for all $v \in \cup_{i=1}^{m-1} X_i$.

Let y_i be any member of Y_i, $i = 1, 2, \ldots, m-1$. Then $(y_i, u_i^*) \in E$ (by Lemma 7). If $(y_{i-1}, y_i) \in E$ then $(u_i^*, y_{i-1}) \in E$, since $y_i < u_i^* < y_{i-1}$. For $i = 1, 2, \ldots, m-1$, we have, $d(u_{m-1}^*, y_i) \leq m-i$ (as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_i^* \to y_i$ or $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_{i+1}^* \to y_i$). Also, u_i^*, $i = 1, 2, \ldots, m-2$; are covered by u_{m-1}^*, since $d(u_{m-1}^*, u_i^*) \leq m-i-1$ (as $u_{m-1}^* \to u_{m-2}^* \to \cdots \to u_{i+1}^* \to u_i^*$).

Thus all members of $\cup_{i=1}^{m-1} N_i$ are covered by u_{m-1}^* except itself.

Lemma 9. *If u_1^* is connected with all vertices of N_1 then u_m^* is the first member of C.*

Proof. Let v_i be any member of X_i, $i = 1, 2, \ldots, m$. Since all members of N_1 are connected to u_1^*, $d(u_m^*, v_1) = m$ (as $u_m^* \to u_{m-2}^* \to \cdots \to u_1^* \to v_1$). For $i = 2, 3, \ldots, m$; $d(u_m^*, v_i) \le m - i + 2$ (as $u_m^* \to u_{m-2}^* \to \cdots \to u_i^* \to v_i$ or $u_m^* \to u_{m-2}^* \to \cdots \to u_{i-1}^* \to v_i$). Therefore, $d(u_m^*, v_i) \le m$ for all $v_i \in X_i$, $i = 1, 2, 3, \cdots, m$.

Again, if y_i be any member of Y_i then $(u_i^*, y_i) \in E$. Also for $i = 2, 3, \ldots, m$; if $(y_i, y_{i-1}) \in E$ then $(u_i^*, y_{i-1}) \in E$. For $i = 1, 2, 3, \ldots, m$, $d(u_m^*, y_i) \le m + 1 - i$ (as $u_m^* \to u_{m-1}^* \to \cdots \to u_i^* \to y_i$ or $u_m^* \to u_{m-1}^* \to \cdots \to u_{i+1}^* \to y_i$). Therefore, $d(u_m^*, v) \le m$ for all $v \in N_i$, $i = 1, 2, 3, \ldots, m$.

Let the first selected facility be u_l^* (either u_{m-1}^* or u_m^*). After selection of the first member of C, we shall select the next facility at u_{l+m}^*. Then two selected facilities u_l^* and u_{l+m}^* are covered by each other and vertices of $\cup_{j=1}^{l+m} N_j$ are covered by these two facilities. This will be proved in the following two lemmas.

Lemma 10. *If v be any member of $\cup_{j=1}^{m} X_{l+j}$ then $d(u_l^*, v) \le m$.*

Proof. Let v_i be any member of X_{l+i}, $i = 1, 2, 3, \ldots, m$ then $(u_{l+i-1}^*, v_i) \in E$ (by Lemma 6). Then, $d(u_l^*, v_i) = i$ (as $u_l^* \to u_{l+1}^* \to \cdots \to u_{l+i-1}^* \to v_i$). Therefore, $d(u_l^*, v) \le m$ for all $v \in \cup_{j=1}^{m} X_{l+j}$.

Lemma 11. *If v be any member of $\cup_{j=1}^{m} Y_{l+j}$ then either $d(u_l^*, v) \le m$ or $d(u_{l+m}^*, v) \le m$.*

Proof. Let y_i be any member of Y_{l+i}, $i = 1, 2, 3, \ldots, m$ then $(u_{l+i}^*, y_i) \in E$ (by Lemma 7). Again, if $(y_i, y_{i-1}) \in E$ then $(u_{l+i}^*, y_{i-1}) \in E$, since $y_i < u_{i+1}^* < y_{i-1}$ for $i = 2, 3, \ldots, m$. There are two cases may arise.

Case 1: $H(y_1) = u_l^*$ and Case 2: $H(y_1) \ne u_l^*$.

Case 1: $H(y_1) = u_l^*$.

In this case, $d(u_l^*, y_i) = i \le m$, (as $y_i \to y_{i-1} \to \cdots \to y_1 \to u_l^*$ or $y_i \to u_{l+i-1}^* \to u_{l+i-2}^* \to \cdots \to u_{l+1}^* \to u_l^*$).

Case 2: $H(y_1) \ne u_l^*$.

In this case, $H(y_1) < u_l^*$, therefore $(u_l^*, y_1) \notin E$. Since $(u_l^*, H(y_1)) \in E$, $d(u_l^*, y_1) = 2$ (as $u_l^* \to u_{l+1}^* \to y_1$ or $y_1 \to H(y_1) \to u_l^*$). Also for $i = 1, 2, 3, \ldots, m - 1$; $d(u_l^*, y_i) = i \le m$ (as $u_l^* \to u_{l+1}^* \to \cdots \to u_{l+i}^* \to y_i$ or $u_l^* \to u_{l+1}^* \to \cdots \to u_{l+i-1}^* \to y_{i-1} \to y_i$ or $\cdots$ or $u_l^* \to H(y_1) \to y_2 \cdots \to y_i$) and $d(u_{l+m}^*, y_m) = 1$ (by Lemma 5).

Hence the lemma.

4. Algorithm and its time complexity

Lemmas 7, 8, 9 and 10 are the backbones of the dynamic programming algorithm to be developed. From the above Lemmas it is observed that if u_l^* is selected as a member of C at any stage, then we are to go to the level $l + m$. If

$l + m \geq h$ then u^*_{l+m} will be the next member of C. If $l + m < h$ then u^*_i, for any $i \in \{l+1, l+2, \ldots, h\}$ will be the next member of C if and only if u^*_l is the first member of C otherwise C is the required conditional covering set.

Now we are in the stage to present the complete algorithm to find the conditional covering set on interval graphs.

Algorithm CCS

Input: An interval graph $G = (V, E)$ with interval representation and uniform coverage radius R.

Output: A conditional covering set C in G.
Initially $C = \Phi$ (empty set) and let $m = \lfloor R \rfloor$.

Step 1: Construct the interval tree IT of the given interval graph.

Step 2: Compute the vertices on the main path of the tree IT and let they be u^*_i, $i = 1, 2, \ldots, h$, h is the height of the tree IT.

Step 3: Compute the sets $X_i, Y_i, i = 1, 2, \ldots, h$.

Step 4: If $h \leq m$, then select any two members on the main path as members of C.

Step 5: If $m < h < 2m - 1$ and u^*_l is the first selected facility, then select second facility at u^*_i where $i = l+1, l+2, l+3, \ldots, h$.

Step 6: If $(u^*_l, v) \in E$ for all $v \in N_1$ then $//h \geq 2m - 1//$
$C = C \cup \{u^*_m\}$ otherwise $C = C \cup \{u^*_{m-1}\}$.

Step 7: Compute $L = l + m$//Select other members of C//
If $L \leq h$ then
  $C = C \cup \{u^*_L\}$
  compute $L = L + m$
endif.

end CCS

For the interval graph considered in Figure 3, the height of the interval tree IT is $h = 6$. Here, $X_1 = \{9\}$, $Y_1 = \phi$, $X_2 = \phi$, $Y_2 = \phi$, $X_3 = \phi$, $Y_3 = \{5\}$, $X_4 = \phi$, $Y_4 = \phi$, $X_5 = \phi$, $Y_5 = \{2\}$, $X_6 = \phi$ and $Y_6 = \phi$. If we take uniform coverage radius $R = 2.1$ then by Algorithm CCS, we have $C = \{7, 4\}$ which is the only solution.

The vertices of IT are the vertices of G. The sets $N_i, i = 1, 2, \ldots, h$ are mutually exclusive and the vertices of N_i are consecutive integers. Again, the sets X_i and $Y_i, i = 1, 2, \ldots, h$ are also mutually exclusive. The vertices of each X_i and Y_i are also consecutive integers. So, we will store only the lowest and highest numbered vertices corresponding to the sets $X_i, Y_i, N_i, i = 1, 2, \ldots, h$ instead of all vertices. If any set is empty then we will take the lowest and highest numbered vertices as 0 and 0. It is obvious that, $| \cup_{i=1}^{h} N_i | = n$.

Theorem 1. *The time complexity to find the CCS on unweighted cost interval graph is $O(n)$, where n is the number of vertices of the given interval graph.*

Proof. For a given interval representation of an interval graph, the unique interval tree IT can be constructed in $O(n)$ time [15]. Since the main path starting from the vertex n and ending at the vertex 1, all the vertices u_i^*, $i = 1, 2, \ldots, h$; on the main path can be identified in $O(n)$ time. Step 7 will be repeated for $\mid C \mid$ times and each time the value of l is increased by m. Hence overall time complexity of algorithm CCS is $O(n)$.

Theorem 2. *The space complexity of algorithm CCS is* $O(n)$.

Proof. By storing the endpoints of intervals one can store an interval graph using $O(n)$ space. The tree IT, sets X_i, Y_i and the vertices $u_i^*, i = 1, 2, \ldots, h$ can be stored using $O(n)$ space. Also in worst case, $\mid C \mid$ may equal to $O(n)$. Hence the total space complexity is $O(n)$.

5. Concluding remarks

In this article, we examine the CCP on unweighted interval graphs with uniform coverage radius. We used the data structure interval tree to solve the problem and present an algorithm which runs in $O(n)$ time. Our framework for designing algorithm for the conditional covering set seems effective on classes of graphs which have similar structure with tree. Unfortunately, the algorithm for CCP on unweighted interval graph presented here does not appear to be extendable to weighted cost CCP. We will continue our study to derive an algorithm for the CCP on weighted interval graphs. A future study can examine one of several practical variations of the CCP or can examine the CCP on various other graphs other than the ones studied.

References

1. S.S. Chaudhry, *New heuristics for the conditional covering problem*, Opsearch, **30** (1993), 42–47.
2. S.S. Chaudhry, I.D. Moon, S.T. McCormick, *Conditional covering: Greedy heuristics and computational results*, Comput. Oper. Res., **14** (1987), 11–18.
3. E.J. Cockayne, R.M. Dawes, S.T. Hedetniemi, *Total domination in graphs*, Networks, **11** (1980), 211–219.
4. M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
5. S.M. Hedetniemi, E.J. Cockayne, S.T. Hedetniemi, *Linear algorithm for finding the Jordan centre and path centre of a tree*, Transport Sci., **98** (1981), 98–114.
6. J.A. Horne, J.C. Smith, *Dynamic programming algorithm for the conditional covering problem on path and extended star graph*, Networks, **46(4)** (2005), 177–185.
7. J.A. Horne, J.C. Smith, *A Dynamic programming algorithm for the conditional covering problem on tree graphs*, Networks, **46(4)** (2005), 186–197.
8. J.M.Keil, *Total domination in interval graphs*, Information Processing Letters, **22** (1986), 171–174.
9. V. Lotfi, I. D. Moon, *Hybrid heuristics for conditional covering problem*, Int. J. Modeling Simulation, **17** (1997), 185–190.
10. B.J. Lunday, J.C. Smith, J.B. Goldberg, *Algorithms for solving the conditional covering problem on paths*, Naval Res. Logistics, **52** (2005), 293–301.

11. S. Mondal, M. Pal, T.K. Pal, *An optimal algorithm to solve 2-neighbourhood covering problem on interval graphs*, Intern. J. computer Math., **79** (2002), 189–204.
12. I.D. Moon, *Conditional covering: worst case analysis of greedy heuristics*, J. Korean OR/MS Society, **15** (1990), 97–104.
13. I.D. Moon, S.S. Chaudhry, *An analysis of network location problems with distance constraints*, Management Sci., **30** (1984), 290–307.
14. I.D. Moon, L. Papayanopoulos, *Facility location on a tree with maximum distance constraints*, Comput. Oper. Res., **22** (1995), 905–914.
15. M. Pal, G.P. Bhattacharjee, *A data structure on interval graphs and its applications*, J. Circuits, Systems and Computers, **7(3)** (1997), 165–175.
16. M. Pal, G.P. Bhattacharjee, *Optimal sequential and parallel algorithms for computing the diameter and center of an interval graph*, Intern. J. Computer Math., **59** (1995), 1–13.
17. M. Pal, G.P. Bhattacharjee, *The parallel algorithms for determining edge packing and efficient edge dominating set in interval graphs*, Parallel Algorithms and Applications, **7** (1995), 193–207.
18. G. Ramalingam, C. Pandu Rangan, *A unified approach to domination problems in interval graphs*, Information Processing Letters, **27** (1988), 271–274.

Akul Rana received his M. Sc. from Vidyasagar University, India in 2001. Since 2006 he has been at Narajole Raj College, India. He has started his Ph. D work at National Institute of Technology, Durgapur on 2007. His research interests include computational graph theory and data structure.

Department of Mathematics, Narajole Raj College, Narajole-721 211, India
e-mail: rana_akul@rediffmail.com

Anita Pal received her M. Sc. and Ph. D from Vidyasagar University, India in 2001 and 2005 respectively. Since 2007 she has been at National Institute of Technology, Durgapur, India. His research interests include computational graph theory, parallel algorithms and data structure.

Department of Mathematics, National Institute of Technology, Durgapur-713 209, India
e-mail: anita.buie@gmail.com

Madhumangal Pal received his M. Sc. from Vidyasagar University, India and Ph. D from Indian Institute of Technology, Kharagpur, India in 1990 and 1996 respectively. He is engaged in research since 1991. In 1996, he received Computer Division Award from Institution of Engineers (India), for best research work. During 1997 to 1999 he was a faculty member of Midnapore College and since 1999 he has been at the Vidyasagar University, India. His research interests include computational graph theory, parallel algorithm, data structure, combinatorial algorithms, genetic algorithms, fuzzy sets and intuitionistic fuzzy sets. His is the Editor-in-Chief of 'Journal of Physical Sciences' and member of the editorial boards of the journals 'International Journal of Fuzzy Systems & Rough Systems', 'International Journal of Computer Science, Systems Engineering & Information Technology' and 'Advanced Modeling and Optimization'.

Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore-721 102, India
e-mail: mmpalvu@gmail.com