*Journal of Mathematics and Informatics*
*Vol. 8, 2017, 45-55*
*ISSN: 2349-0632 (P), 2349-0640 (online)*
*Published 31 July 2017*
*www.researchmathsci.org*
*DOI: http://dx.doi.org/10.22457/jmi.v8a6*

Journal of
**Mathematics and Informatics**

# The 2-Tuple Domination Problem on Circular-arc Graphs

***Angshu Sinha*[1], *Akul Rana*[2] and *Anita Pal*[3]**
[1]Department of Mathematics,
NSHM Knowledge Campus, Durgapur-713212, INDIA
e-mail:angshusinha20@gmail.com

[2]Department of Mathematics, Narajole Raj College
Narajole, Paschim Medinipur- 721211, INDIA
e-mail: arnrc79@gmail.com
[3]Department of Mathematics, National Institute of Technology Durgapur
Durgapur-713209, INDIA
Corresponding author. e-mail: anita.buie@gmail.com

*Received 20June 2017; accepted 26 July 2017*

***Abstract.*** Given a simple graph $G=(V,E)$ and a fixed positive integer $k$. In a graph $G$, a vertex is said to dominate itself and all of its neighbors. A set $D \subseteq V$ is called a $k$-tuple dominating set if every vertex in $V$ is dominated by at least $k$ vertices of $D$. The $k$-tuple domination problem is to find a minimum cardinality $k$-tuple dominating set. This problem is NP-complete for general graphs. In this paper, the same problem restricted to a class of graphs called circular-arc graphs are considered. In particular, we presented an $O(n^2)$-time algorithm to solve the 2-tuple domination problem on circular-arc graphs, one of the non-tree type graph classes.

***Keywords:*** *Design of algorithms, analysis of algorithms, circular-arc graph, domination,* $2$*-tuple domination.*

***AMS Mathematics Subject Classification (2010):***05C69, 05C78

## 1.Introduction

Circular arc graphs are intersection graphs of arcs of a circle. These graphs have been reported since 1964 and they have been received considerable attention for a series of papers by Tucker in the year 1970. Various subclasses of circular-arc graphs have been also studied. Among these are the proper circular-arc graphs, unit circular-arc graphs, Helly circular-arc graphs and co-bipartite circular-arc graphs. Several characterizations and recognition algorithms have been formulated for circular-arc graphs and its subclasses. In particular, it should be mentioned that linear time algorithms are known for all these classes of graphs.

A graph $G=(V,E)$ is called an intersection graph for a finite family $F$ of non empty sets if there is a one-to-one correspondence between $F$ and $V$ such that two sets in $F$ have a non empty intersection if and only if their corresponding vertices in $V$ are adjacent to each other. $F$ is called an intersection model of $G$ and $G$ is called the intersection graph of $F$. If $F$ is a family of arcs around a circle then $G$ is called a circular-arc graph. If $F$ is a family of line segments on the real line, then $G$ is called an interval graph. An interval graph is a special case of circular-arc graphs; it is a circular-arc graph that can be represented by arcs that do not cover the entire circle. Some circular-arc graphs do not have such a representation, so the class of interval graphs is a proper subclass of the class of circular-arc graphs.

In general, problems for circular-arc graphs tend to be more difficult than for interval graphs. One of the reasons is that intervals of a real line satisfy the Helly property, while arcs of a circle do not necessarily satisfy it. This implies that the maximal cliques of an interval graph can be associated to chosen points of the line. The latter means that an interval graph can have no more maximal cliques than vertices. In contrast, circular-arc graphs may contain maximal cliques which do not correspond to points of the circle.

The domination in graph theory is a natural model for many location problems in operations research. In a graph $G$, a vertex is said to dominate itself and all of its neighbors. A dominating set of $G=(V,E)$ is a subset $D$ of $V$ such that every vertex in $V$ is dominated by at least one vertex in $D$. The domination number $\gamma(G)$ is the minimum cardinality dominating set of $G$.

For a fixed positive integer $k$, a $k$-tuple dominating set of a graph $G=(V,E)$ is a subset $D \subseteq V$ such that every vertex in $V$ is dominated by at least $k$ vertex in $D$. The $k$-tuple domination number $\gamma_{\times k}(G)$ is the minimum cardinality of a $k$-tuple dominating set of $G$. The special case when $k=1$ is the usual domination. The case when $k=2$ was called double domination or 2-tuple domination in [12]. A 2-tuple dominating set $D$ is said to be minimal, if there does not exist any $S \subset D$ such that $S$ is a 2-tuple dominating set of $G$. A 2-tuple dominating set $D$, denoted by $\gamma_{\times 2}(G)$, is said to be minimum, if it is minimal as well as it gives 2-tuple domination number. Since every vertex in $V$ is dominated by at least 2 vertices in $D$, therefore $D$ contains at least two members, i.e., $|D| \geq 2$. The case when $k=3$ was called triple domination in [16]. This problem is NP-hard for general graphs. There exist polynomial time algorithms to solve 2-tuple domination problem for some special classes of graphs [1, 13, 15, 17].

### 1.1.Review of previous work

Eschen and Spinrad [6] presented an $O(n^2)$-time algorithm for recognizing a circular-arc graph and constructing a circular-arc model. There are several results on the circular-arc graphs. We only mention results relevant to the class of domination problems studied in this paper. Domination and its variations have been extensively studied in the literature, see [3, 9, 10]. Chang [3] presented a unified approach to design efficient $O(n)$ or $O(n \log\log n)$ algorithms for the weighted domination problem and the weighted independent, connected, and total domination problems on interval graphs, and extended the algorithms to solve the same problems on circular-arc graphs in $O(n+m)$ time. Among the variations of domination, the $k$-tuple domination was introduced in [12], also see [9]. While determining the exact value of $\gamma_{\times k}(G)$ for a graph $G$ is not easy, many studies focus on its upper bounds. Gagarin and Zverovich presented an upper bound for general graphs in [7], and later Chang [5] improved it. In [15], Pramanik, Mondal and Pal solved 2-tuple domination problem on interval graphs using $O(n^2)$ time. Recently, Barman, Mondal and Pal [1] solved 2-tuple domination problem on permutation graphs in $O(n^2)$ time.

### 1.2.Application

A circular-arc graph is a general form of interval graph and it is one of the most useful discrete mathematical structures for modeling problems arising in the real world. Circular-arc graphs arise in genetics, traffic control, computer compiler design, scheduling problems and other combinatorial problems.

Domination in graphs has many applications in several fields. Domination arises in facility location problems, where the number of facilities (e.g., hospitals, fire stations etc.) is fixed and one attempts to minimize the distance that a person needs to travel to get to the closest facility. A similar problem occurs when the maximum distance to a facility is fixed and one attempts to minimize the number of facilities necessary so that everyone is serviced. Concepts from domination also appear in problems involving finding sets of representatives, in monitoring communication or electrical networks, and in land surveying (e.g., minimizing the number of places a surveyor must stand in order to take height measurements for an entire region). An important application for network purposes of $k$-tuple domination is for fault tolerance or mobility.

### 1.3.Main result

Given the sorted array of end points of the arcs in the intersection model of the circular-arc graph. An algorithm is designed to solve the 2-tuple domination problem on the circular-arc graph with running time $O(n^2)$, where $n$ is the number of arcs.

### 1.4.Organization of the paper

The remainder of this paper is organized as follows. In Section 2 we introduce the notations and definitions used throughout the paper. In Section 3 we study the approaches towards solving 2-tuple domination problem for Circular arc graphs and present some intermediate results for the same. In Section 4, we present an $O(n^2)$ time algorithm for the 2-tuple domination problem in circular-arc graphs. Section 5 contains some concluding remarks.

## 2.Notations and preliminaries

Let $A = \{A_1, A_2, ..., A_n\}$ be the circular arc family of circular-arc graph $G = (V, E)$. The family of circular-arcs are located around a circle $C$. While going in a clockwise direction, the point at which we first encounter the arc $A_i$ will be called the left point of the arc $i$ and is denoted by $l_i$. Similarly, the point at which we leave the arc $A_i$ will be called the right point of the arc $i$ and is denoted by $r_i$. Every arc can be represented by their two endpoints, e.g., $A_i$ can be represented as $[l_i, r_i]$ where $l_i$ is the left point and

$r_i$ is the right point of the arc $A_i$ on the circle $C$. A ray is a straight line from the centre of the circle $C$ passing through the right end point of any arbitrary chosen arc. We label this arc by $n$, then start anticlockwise traversal from the ray. We label $n-1$ to the arc with next successive right end point. In this process, we label all the remaining arcs.

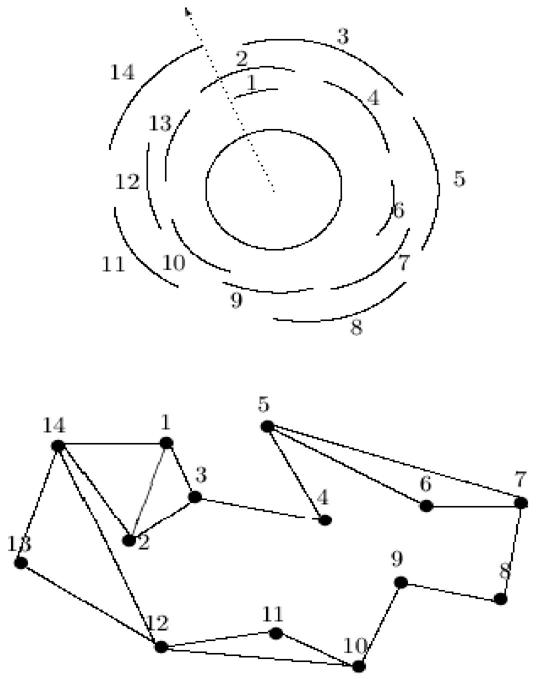


**Figure 1:** A circular-arc graph and its circular-arc representation

Without loss of generality, we assume the following :

1. An arc contains both its end points and that no two arcs share a common end point.
2. The graph $G$ is connected and the list of sorted endpoints is given.
3. No single arc in $A$ cover the entire circle $C$.
4. Arcs and vertices of a circular-arc graph are same.
5. The endpoints of the arcs in $A$ are sorted according to the order in which they are visited during the anticlockwise traversal along circle by starting at an arbitrary arc called $A_n$.
6. The arcs are sorted in increasing values of $r_i$ i.e., $r_i < r_j$ for $i < j$.
7. $\bigcup_{i=1}^{n} A_i = C$ (otherwise, the problem becomes one on interval graph).

We henceforth assume that the arcs are given sorted by their right endpoints. In this paper, we consider only finite, undirected, connected graphs without self loops or multiple edges. From now on, when there is no ambiguity we use the term "arcs" and "vertices" interchangeably. We use the same labels to refer to the vertices in a circular-arc graph as well as the corresponding arcs in its circular-arc representation that is being considered.

The family of arcs $A$ is said to be canonical, if $l_i$ and $r_i$ for all $i = 1,2,...,n$ are distinct integers between 1 to $2n$.

If $A$ is not canonical, using sorting one can construct a canonical family of arcs using $O(n \log n)$ time.

An arc $i$ is contained in arc $j$, if every point of the arc $i$ is in the arc $j$. Let $N(i)$ be the set of arcs intersecting the arc $i$ and $N[i] = N(i) \cup \{i\}$.

A vertex $u$ is 1-dominated by itself and all its neighbors. Unless stated, we use the vertex $u$ dominate the vertex $v$ to mean $u$ 1-dominate the vertex $v$. A vertex is said to be 2-dominated if it is dominated by at least two vertices.

The span of a vertex $u$ is defined to be the highest arc which is dominated by the vertex $u$ and is denoted by span$(u)$.

That is, span$(u) = max\,\{x : x \in N[u]\}$.

The continuous part of the circle that starts at the point $l$ and terminates at the point $r$ is denoted by seg$(l\,,\; r)$.

A point $x$ on the circle is said to be in the arc $A_i$ if it lies within the interior of the seg$(l_i\,,\; r_i)$.

For each $i$, define $next_1(i)$ to be the arc in $N[i]$ whose right end point is last encountered in a clockwise traversal from $l(i)$.

The array $next_2(i)$ to be the arc in $N[i]$ whose right end point is first encountered in anticlockwise traversing from the point $r(next_1(i))$. From this definition, it follows that $r(next_1(i)) > r(next_2(i))$.

**3.Some results**

Instead of presenting the algorithm on the circular-arc graph $G$, we will be working with the intersection model of $G$. Observe that, the vertex $next_1(i)$ dominates the arc $i$ and all arcs which overlap with the arc $next_1(i)$. Also the vertex $next_2(i)$ dominates the arc $i$ and all arcs which overlap with the arc $next_2(i)$.

**Lemma 1.***The vertex* $next_1(i)$ *has maximum span 1-dominating the vertex* $i$.

**Proof:** From definition of $next_1(i)$, it follows that the vertex $next_1(i)$, 1-dominates the vertex $i$. Also the vertex $next_1(i)$ has maximum right end point among the vertices of $N[i]$, that is, $next_1(i)$ dominates the highest arc. Hence the result.

It is easy to see that, the vertex $next_2(i)$ has next to maximum span 1-dominating the vertex $i$.

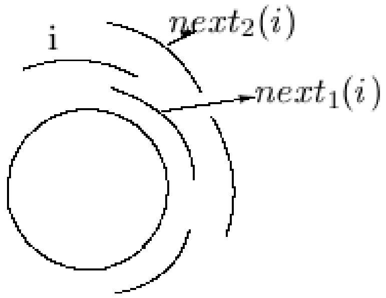


**Figure 2:** A part of circular-arc representation

The following lemma plays an important role for the development of the algorithm.

**Lemma 2.***Let* $m(i) = max\{l(next_1(i)), l(next_2(i))\}$. *The vertices* $next_1(i)$ *and* $next_2(i)$ *2-dominates all arcs in* $seg(m(i), r(next_2(i))$.

**Proof:** Let $j$ be any arc which overlaps with $seg(m(i), r(next_2(i))$. Then there must be a point $x$ of the arc $j$ such that $l(next_1(i)) < x < r(next_1(i))$ and $l(next_2(i)) < x < r(next_2(i))$. Since $m(i) = max\{l(next_1(i)), l(next_2(i))\}$ and $r(next_2(i)) < r(next_1(i))$, therefore $m(i) < x < r(next_2(i))$. Then the point $x$ is common to both $next_1(i)$ and $next_2(i)$. This implies every point of $seg(m(i), r(next_2(i))$ overlap with both the arcs $next_1(i)$ and $next_2(i)$ .i.e., the arc $j$ is 2-dominated by both $next_1(i)$ and $next_2(i)$.

Let $D$ be the 2-tuple dominating set. Initially, $D = \emptyset$ and $i = 1$. Firstly $next_1(i)$ and $next_2(i)$ selected as members of $D$. Then we select vertex $p$ whose right end point is first encountered in clockwise traversal from $r(next_2(i))$ such that $l(p) > r(next_2(i))$. Two cases may arise: $p$ intersect $next_1(i)$ or $p$ does not intersect $next_1(i)$.

If $p$ intersect $next_1(i)$ then the vertex $p$ is 1-dominated by $next_1(i)$. Since $next_1(p)$ has a maximum span among all vertices which dominates the vertex $p$, $next_1(p)$ will be selected as the next member of dominating set. There are two sub cases: $next_1(p) \in N(next_1(i))$ or $next_1(p) \notin N(next_1(i))$.

For the first sub case among the vertices $next_1(i)$ and $next_1(p)$, select the vertex whose right end point is minimum and replace that vertex by $next_2(i)$. If $next_1(p) \notin N(next_1(i))$, $next_2(p)$ will be selected as a member of $D$ to 2-dominate the arc $p$ and replace $next_2(i)$ by $next_2(p)$.

If $p$ does not intersect $next_1(i)$, then $i$ is replaced by $p$ and the above process is repeated.

This procedure is continued until $next_1(i) = n$ or $p = n$.

Note that, $r(next_1(i)) > r(next_2(i))$ for every vertex $i$. Therefore $next_2(i) \neq n$ for any $i \in V$.

**Lemma 3.** *If $next_1(i) = n$ for some $i \in V$ then $D = D \cup \{next_2(i), n\}$. Moreover, if there is no vertex $k$ such that $k \in seg(r(1), r(next_2(1))$ then $next_2(1)$ is deleted from $D$.*

**Proof:** Recall that, initially the vertices $next_1(1)$ and $next_2(1)$ selected as a member of $D$ to 2-dominate the vertex 1. Now, the vertices 1 and $n$ are adjacent. Since the vertex $next_1(i) = n$ is a member of $D$, the vertex 1 is 2-dominated by $next_1(1)$ and $n$. If there is no vertex $k$ such that $k \in seg(r(1), r(next_2(1))$ then $next_2(1)$ does not dominate any vertex other than $1$. The deletion of $next_2(1)$ from $D$ reduces the cardinality of $D$ as well as resulting set $D$ is a 2-tuple dominating set. Since our aim is to compute minimum cardinality 2-tuple dominating set, $next_2(1)$ is deleted from $D$. Hence the lemma.

**Lemma 4.** *If $p = n$ then the dominating set may be changed or unaltered.*

**Proof:** If $p = n$ then there are four cases may arise.

***Case 1:*** $p \in N(next_1(1))$ and $p \in N(next_2(1))$.

In this case the vertex $p$ is 2-dominated $N(next_1(1))$ and $N(next_2(1))$ and $D$ remains unaltered.

***Case 2:*** $p \in N(next_1(1))$ or $p \in N(next_2(1))$ and $p \in N(q)$ for any $q \in D \setminus \{next_1(1), next_2(1)\}$.

In this case also all the arcs are covered and $D$ remains unaltered.

***Case 3:*** $p \in N(next_1(1))$ or $p \in N(next_2(1))$ or $p \in N(q)$ for some $q \in D \setminus \{next_1(1), next_2(1)\}$.

In this case $p$ is 1-dominated by $next_1(1)$ or $next_2(1)$ or $q$ for some $q \in D \setminus \{next_1(1), next_2(1)\}$. To 2-dominate the vertex $p$, $next_1(p)$ is selected as a member of $D$. Then the vertex $1$ is adjacent to $next_1(p)$. The vertex $1$ is 2-dominated by $next_1(1)$ and $next_1(p)$. If there is no vertex $k$ such that

$$k \in seg(r(1), r(next_2(1))$$

then $next_2(1)$ does not dominate any vertex other than $1$. Hence, in this case $next_2(1)$ will be deleted from $D$. Otherwise, $D$ remains unaltered.

*Case 4:* $p \notin N(next_1(1))$ or $p \notin N(next_2(1))$ or $p \notin N(q)$ for some

$q \in D \setminus \{next_1(1), next_2(1)\}$.

In this case to 2-dominate the vertex $p$, $next_1(p)$ and $next_2(p)$ is selected as member a of $D$. Then the vertex 1 is adjacent to $next_1(p)$. The vertex 1 is 2-dominated by $next_1(1)$ and $next_1(p)$. If there is no vertex $k$ such that $k \in seg(r(next_1(p)), r(next_2(1))$ then $next_2(1)$ does not dominate any vertex other than 1. Hence $next_2(1)$ will be deleted from $D$ if $k \notin seg(r(next_1(p)), r(next_2(1))$. Otherwise, $D$ remains same.

Hence the lemma follows.

## 4.The algorithm

Based on the above results and discussion a formal description of the algorithm is given below.

**Algorithm 2TDP**

**Input:** A set of sorted arcs of a circular-arc graph $G = (V, E)$.

**Output:** A minimum cardinality 2-tuple dominating set of $G$.

Initially $D = \Phi$ (empty set).

**Step 1:** Compute $N[i]$, $next_1(i)$ and $next_2(i)$ for each vertex $i \in V$.

**Step 2:** Initialize $i = 1$. $D = D \cup \{next_1(i), next_2(i)\}$.

**Step 3:** Select the arc whose right end point is first encountered in clockwise traversal starting from $r(next_2(i))$. Let it be $p$.

**Step 4:** If $p \in N(next_1(q))$ for some $q \in D$ then $D = D \cup \{next_1(p)\}$.

If $next_1(p) \in N(next_1(i))$ then among the arcs $next_1(i)$ and $next_1(p)$, select the arc whose right end point is minimum, goto Step 3 and replace that vertex by $next_2(i)$.

else [$next_1(p) \notin N(next_1(i))$ then]

If $i \notin seg(l(p), r(next_1(p))$ then

$D = D \cup \{next_1^2(p)\}$, goto Step 3 and replace $next_2(i)$ by $next_1(p)$.

else

$D = D \cup \{next_2(p)\}$, goto Step 3 and replace $i$ by $p$.

endif

endif

**Step 5:** If $p \notin N(next_1(q))$ for any $q \in D$, replace $i$ by $p$ and goto Step 2.

**Step 6:** If $p = n$ then

If $p \in N(next_1(1)) \cap N(next_2(1))$ or if $p \in N(q)$ for some $q \in D \setminus \{next_1(1), next_2(1)\}$ and $p \in N(next_1(1)) \cup N(next_2(1))$ then $D$ is unaltered

elseif $p \in N(next_1(1))$ or if $p \in N(next_2(1))$ or if $p \in N(q)$ for some

$q \in D \setminus \{next_1(1), next_2(1)\}$
$D = D \cup \{next_1(p)\}$
If $\exists$ some arc $k$ such that $k \in seg(r(next_1(p)), r(next_2(1)))$ then
$D$ remains unaltered
else
$D = D \setminus \{next_2(1)\}$
else
$D = D \cup \{next_1(p), next_2(p)\}$
If $\exists$ some arc $k$ such that $k \in seg(r(next_1(p)), r(next_2(1)))$ then
$D$ remains unaltered
else
$D = D \setminus \{next_2(1)\}$
endif
**Step 7:** If $next_1(i) = n$ then
$D = D \cup \{next_1(i), next_2(i)\}$
If $\exists$ any arc $k$ such that $k \in seg(r(1), r(next_2(1)))$ then
$D$ remains unaltered
else
$D = D \setminus \{next_2(1)\}$
endif
**end 2TDP**

**Lemma 5.** *The set $D$ is a minimum 2-tuple dominating set.*
**Proof:** It is easy to see that, the set $D$ obtained by the Algorithm 2TDP is a 2-tuple dominating set. Now we shall show that $D$ is the minimum 2-tuple set. For this purpose, we discuss those steps of the Algorithm 2TDP where we select the member(s) of the set $D$ in the following:
Step 2: In this step we include the elements $next_1(1)$ and $next_2(1)$ as member of $D$. Two members of $D$ are necessary to 2-dominate the vertex $1$. Also $next_1(1)$ and $next_2(1)$ has maximum span and next to maximum respectively dominating the vertex 1. Therefore $next_1(1)$ and $next_2(1)$ are the best selection to achieve a maximum span.
Step 4: Let $p$ be the arc whose right end point is first encountered in clockwise traversal from $next_2(1)$. Then $p$ is not dominated by $next_2(1)$. If $p \in N(next_1(q))$ for some $q \in D$ then $p$ is dominated by $q$. To 2-dominate the vertex $p$, $\{next_1(p)\}$ is selected as a member of $D$, since it has a maximum span dominating the vertex $p$.

If $next_1(p) \in N(next_1(1))$ then the vertex $next_1(p)$ is 2-dominated by $next_1(1)$ and $next_1(p)$. To 2-dominate the other vertices clockwise traversal is done from $r(next_1(1))$, if $r(next_1(1)) < r(next_1(p))$. Otherwise, clockwise traversal is done from $r(next_1(p))$.

If $next_1(p) \notin N(next_1(1))$ then the vertex $next_1(p)$ is dominated by itself only.

If there is no arc in $seg(l(p), r(next_1(p))$ then the vertex $next_1^2(p)$ is added to $D$ and clockwise traversal is done from $r(next_1(p))$. Otherwise, the vertex $next_2(p)$ added to $D$ and clockwise traversal is taken from $r(next_2(p))$.

Finally, when $p = n$ or $next_1(i) = n$, the dominating set is modified in Step 6 and Step 7 (Lemma 3 and 4).

In the Step 2 and Step 4 of the algorithm, the elements of $D$ are selected (or deleted) in such a way that maximum span is achieved as well as each arc of the graph $G$ is 2-dominated by minimum number of vertices. Therefore $D$ is a minimum 2-tuple dominating set.

**Theorem 1.***Algorithm 2TDP finds a 2-tuple dominating set on circular-arc graphs in* $O(n^2)$ *time.*

**Proof:** The time complexity of Algorithm 2TDP is caused mainly by the computation of $N(i)$. For each $i \in V$, calculation of $N(i)$ requires $O(n)$ time where $n$ is the total number of arcs.This is repeated for $n$ times. Therefore total time to compute $N(i)$ is $O(n^2)$. The remaining part of the algorithm requires $O(n)$ steps Thus the overall time complexity is $O(n^2) + O(n) = O(n^2)$.

## 5.Concluding remarks

In this paper, we developed an efficient algorithm that solves the $2$-tuple domination problem on circular-arc graphs using $O(n^2)$ time. The same algorithm can be applied to a subset of circular-arc graphs known as interval graphs and the time complexity remains unchanged. A future study can investigate to design a polynomial time algorithm to solve $k$-tuple domination problem for $k > 2$.

## REFERENCES


1. S.Barman, S.Mondal and M.Pal, Minimum 2-tuple dominating set of permutation graphs, *J Appl Math Comput.*, 43(1-2) (2013) 133-150.
2. M.A.Bonuccelli, Dominating sets and domination number of circular-arc graphs, *Discrete Applied Mathematics,* 12 (1985) 203-213.
3. M.S.Chang, Efficient algorithms for the domination problems on interval and circular-arc graphs, *SIAM Journal on Computing*, 27(1998) 1671-1694.
4. G.J.Chang, Algorithmic aspects of domination in graphs,*Handbook of Combinatorial Optimization (D.-Z. Du and P. M. Pardalos eds.)*, 3 (1998) 339-405.
5. G.J.Chang, The upper bound on $k$-tuple domination numbers of graphs, *Europian Journal of Combinatorics)*, 29(5) (2008) 1333-1336.
6. E.M.Eschen and J.P.Spinrad, An $O(n^2)$ algorithm for circular-arc graph recognition, in *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms,*

*Austin, Texas*, (1993) 128-137.

7. A.Gagarin and V.E.Zverovich, A generalised upper bound for the $k$-tuple domination number, *Discrete Mathematics*, 308 (5) (2008) 880-885.
8. M.C.Golumbic, Interval graphs and related topics, *Discrete Mathematics*, 55 (1985) 113- 121.
9. T.W.Haynes, S.T.Hedetniemi and P.J.Slater, *Domination in Graphs: The Theory*, Marcel Dekker Inc. New York (1998).
10. T.W.Haynes, S.T.Hedetniemi and P.J.Slater, *Domination in Graphs: Advanced Topics*, Marcel Dekker Inc. New York (1998).
11. J.Harantand M.A.Henning, On double dominations in graphs, *Discuss. Math. Graph Theory*, 25 (2005) 29-34.
12. F.Harary and T.W.Haynes, Double domination in graphs, *Ars Combin.*, 55 (2000) 201-213.
13. C.S.Liaoand G.J.Chang, Algorithmic aspect of $k$-tuple domination in graphs, *Taiwanese J. Math.*, 6 (2002) 415-420.
14. C.S.Liao and G.J.Chang, $k$-tuple domination in graphs, *Inform. Proces. Letters*, 87 (2003) 45-50.
15. T.Pramanik, S.Mondal and M.Pal, Minimum 2-tuple dominating set of an interval graph, *International Journal of Combinatorics*, vol. 2011, Article ID 389369, 14 pages, 2011. doi:10.1155/2011/389369
16. D.Rautenbach and L.Volkman, New bounds on the $k$-domination number and the $k$-tuple domination number, *Applied Math. Letters*, 20 (2007) 98-102.
17. A.K.Sinha, A.Rana and A.Pal, The 2-tuple domination problem on trapezoid graphs, *Annals of Pure and Applied Mathematics*, 7(1) (2014) 71-76.
18. M.Hota, M.Pal and T.K.Pal, An efficient algorithm for finding a maximum weight k-independent set on trapezoid graphs, *Computational Optimization and Applications*, 18 (1) (2001) 49-62.
19. M.Hota, M.Pal and T.K.Pal, An efficient algorithm to generate all maximal independent sets on trapezoid graphs, *International Journal of Computer Mathematics*, 70 (4) (1999) 587-599.
20. S.Mondal, M.Pal and T.K.Pal,An optimal algorithm for solving all-pairs shortest paths on trapezoid graphs, *International Journal of Computational Engineering Science*, 3 (2)(2002) 103-116.
21. D.Bera, M.Pal and T.Pal, An efficient algorithm to generate all maximal cliques on trapezoid graphs, *International Journal of Computer Mathematics*, 79 (10) (2002) 1057-1065.
22. S.Mondal and M.Pal, A sequential algorithm to solve next-to-shortest path problem on circular-arc graphs, *J. Phys. Sci*., 10 (2006) 201-217.
23. A.Saha, M.Pal and T.K.Pal, Selection of programme slots of television channels for giving advertisement: A graph theoretic approach, *Information Sciences*, 177 (12) (2007) 2480-2492.