Discrete Mathematics, Algorithms and Applications
(2023) 2350015 (12 pages)

DOI: 10.1142/S1793830923500155

Edge-vertex domination on interval graphs

Amita Samanta Adhya*,§, Sukumar Mondal†,¶
and Sambhu Charan Barman‡,‖

**Research Centre in Natural and Applied Sciences*
(Department of Computer Science)
Raja N. L. Khan Women's College (Autonomous), India

†*Department of Mathematics*
Raja N. L. Khan Women's College (Autonomous), India

‡*Department of Mathematics*
Shahid Matangini Hazra Government General
Degree College for Women, India
§*amitasamanta1@gmail.com*
¶*sukumarmondal@rnlkwc.ac.in*
‖*barman.sambhu@gmail.com*

Received 28 July 2022
Revised 21 August 2022
Accepted 3 February 2023
Published 28 March 2023

Communicated by Weili Wu

For an undirected as well as connected graph $G = (V, E)$, a node point $w \in V(G)$ is edge-vertex dominated by an edge $e \in E(G)$ if w is incident to e or w is incident to an adjacent edge of e. A set $D_{\text{EV}} \subseteq E$ is called an edge-vertex dominating set of G if every node point of G is edge-vertex dominated by at least one edge of D_{EV}. The minimum cardinality among all edge-vertex dominating sets is the edge-vertex domination number, symbolled by $\gamma_{\text{EV}}(G)$. Here, we propose an algorithm that runs in $O(n)$-time for determining a minimum-cardinality D_{EV} of interval graph with n nodes. We also study some properties relating to the edge-vertex dominating set of interval graphs.

Keywords: Domination; edge-vertex domination number; interval graph.

Mathematics Subject Classification 2020: 05C30, 05C12, 68R10

1. Introduction

In a graph structure $G = (V, E)$, V represents the node point set/vertex set and E represents the link set/edge set. We also assume that G is connected and has neither self-loop nor parallel edges. We use the symbol $N_G(w)$ to represent the

‖Corresponding author.

open neighborhood of the node $w \in V(G)$, that is $N_G(w) = \{u : (u, w) \in E(G)\}$. Also, we use the notation $N_G[w]$ to indicate the closed neighborhood of the node point w, where $N_G[w] = N_G(w) \cup \{w\}$.

Domination is always an interesting and vital topic to researchers. Claude Berge, in his book [5], defined the basic idea of the domination number (D-number) $\gamma(G)$ of G for the first time. The term dominating set (D-set) and D-number were first mentioned by Ore [39]. About 20 years ago, Haynes *et al.* [23] wrote a revolutionary book on the fundamentals of domination, and they recorded over 12,000 research papers on domination and its different parameters in graphs. Sampathkumar *et al.* [49] first introduced the term "connected domination number". Later, Dorbec *et al.* [13] established independent domination in cubic graphs. Variations of domination like connected domination [3], edge domination, perfect domination, secured domination, k-tuple domination and k-hop domination [14, 32, 51], weighted domination, paired-domination, independent-domination, roman domination [35], inverse roman domination [33], total domination [50], isolate domination [4, 21], k-efficient domination [35], restrained domination [15], independent line-set domination [20] have been briefly discussed in the literature [9, 23, 29, 44]. Slater renames a k-hop D-set as a k-basis [51]. Besides these, many researchers studied various properties of graphs concerning domination [10, 12, 23–27, 52]. Like most of the general graph problems, minimum k-hop connected D-set problem is NP-Complete for arbitrary graphs [54], even in unit disc graphs (UDG), this problem is NP-Complete [36]. On planar graphs, Demaine *et al.* [14] proposed an algorithm that takes $O(n^4)$ time to determine k-hop D-set with minimum cardinality. Natarajan *et al.* [37] have done some fundamental works on hop D-number on some special class of graphs. Also, on permutation graphs, Rana *et al.* [43] set up an efficient algorithm to find a distance-k D-set. Later, Ayyaswamy *et al.* [2] worked on the upper and lower limits of the hop D-number on trees. Besides these, on UDGs and arbitrary graphs, many researchers set up some algorithms for solving k-hop connected D-set problem [11, 18, 45, 55]. Also, in [7], Basuchowdhuri *et al.* determined influential vertices for traditional communication networks using k-hop D-set. Kundu *et al.* [32] set up an optimal algorithm for determining a k-hop D-set on trees. It is a general work of the prior result to find a 1-hop D-set (with minimum cardinality) of a tree [10]. Favaron *et al.* [16] established that the diameter of the domination k-critical graphs is at most $2(k-1)$ while $k \geq 2$. Later, Ramy [46] showed the bounds for the 2-D-number of toroidal grid graphs. Also, Barman *et al.* [6] formulated an algorithm (runs in $O(n)$ time) to compute a d-hop D-set with minimum cardinality on interval graphs. In 2021, Adhya *et al.* [1] presented an algorithm (runs in $O(n)$ time) for determining a k-hop connected D-set having the least node points on permutation graphs.

For a connected as well as undirected graph $G = (V, E)$, a node point $w \in V(G)$ is edge-vertex dominated (EV-dominated) by an edge $e \in E(G)$ if w is incident to e or w is incident to an adjacent edge of e. A set $D_{\mathrm{EV}} \subseteq E$ is called an edge-vertex dominating set (EVDS) of G if every node point of G is edge-vertex dominated by at least one edge of D_{EV}. The minimum cardinality among all edge-vertex dominating sets is the edge-vertex domination number (EVDN). We denote it by the symbol $\gamma_{\text{EV}}(G)$. If D_{EV} is an EVDS and every edge in D_{EV} has an endpoint incident to another edge in D_{EV}, then D_{EV} is called the total EVDS. The minimum cardinality of total EVDS is called the total edge-vertex domination number, and we denote it by γ^t_{EV}. In the survey, like the EVDS there is another type of dominating set called vertex-edge dominating set [41]. In this paper, we focus only on EVDS.

1.1. *Interval graph*

An interval graph (IG) is a well-known intersection graph of the intervals drawn on a real line. Basically, we can build an undirected interval graph $G(V, E)$ from a collection of intervals I_j, $j = 0, 1, 2, \ldots, n$, by generating a node point w_i for each interval I_i, and linking two node points v_i and v_j by an edge whenever the corresponding two intervals intersect each other. Therefore, $E(G) = \{(v_i, v_j) \mid I_i \cap I_j \neq \emptyset\}$. Figure 1 shows an IG and its corresponding interval representation.

In operations research and scheduling theory, we usually utilize IGs for the representation of resource assignment problems. In these cases, each request, like a link in a network or a room assigned for a class in school or college or a processor in compiling system, etc., represents an interval for a particular time. Besides these, many researchers found applications of IG in computer science, genetics, bioinformatics, file arrangements [8], job assigning [8], protein sequencing [28], DNA mapping [19, 28], mechanism of macro substitution [17], circuit routine [38], temporal reasoning in AI, routing between two points nets [22], etc.

A very important result on IG $G = (V, E)$ is stated as follows.

Lemma 1 ([40]). *If the node points $v_i, v_j, v_k \in V(G)$, $i < j < k$ and v_i is adjacent with v_k, then v_j is adjacent with v_k but v_j may or may not adjacent with v_i.*

1.2. *Survey of the related works*

Peters *et al.* [41] first introduced EVDS and vertex-edge D-set of graphs. The EVDS problems are NP-complete. This problem is NP-complete for bipartite graphs [34] also. The double version of EVDS was studied by Sahin *et al.* [47]. They also

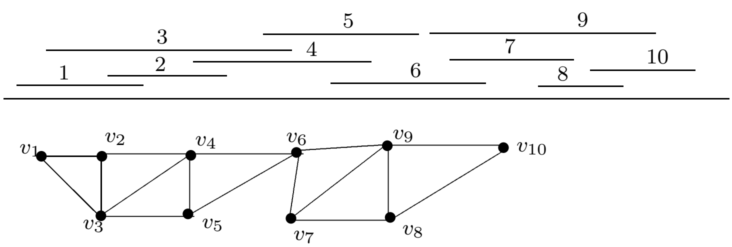

Fig. 1. An interval graph and its interval representation.

established the relationship between γ_{dev} and γ_{dve}, γ_t, γ_{EV} for arbitrary tree and graphs. They also determined the double EVDN for cycles and path graphs. In [48], Sahin *et al.* showed that the total EVDS problem is NP-hard for bipartite graphs and established that $\gamma^t_{\mathrm{EV}} \leq (n - l + 2s - 1)/2$ for a tree having n node points, where l indicates the number of leaves and s is the number of supporting node points. Venkatakrishnan *et al.* [53] derived a tight upper bound of EVDN such that $\gamma_{\mathrm{EV}}(T) \leq ((\gamma_t(T) + s - 1)/2)$ for trees and characterized the trees using this upper bound. Krishnakumari *et al.* showed that $\gamma^t(T) = 1 + \gamma_{\mathrm{EV}}$ in [30]. Recently, Kim *et al.* [31] proved for each tree T with n nodes that $(n- ,l+2)/4 \leq \gamma_{\mathrm{EV}}(T) \leq (n-1)/2$, where $n \geq 3$ and l is the number of leaves.

1.3. *Applications*

The concept of EVDS can be applied in placing monitoring/tracking devices like cameras to protect facilities in a given area. For illustration, in the graph model of placing monitoring/tracking devices problem, if we place the monitoring/tracking devices at the end-vertices of links/edges, i.e., at the members of a EVDS dominating set, then these tracking devices can track/observe all the adjacent vertices (i.e., facilities) including the tracking devices located at another end vertices of the edges belong to the EVDS. If one of the tracking devices stops working, then the immediate adjacent device can service those vertices/facilities watched by the malfunctioned device.

1.4. *Main outcome*

Here, we propose an algorithm that runs in $O(n)$ time for determining a minimum-cardinality edge-vertex dominating set on interval graph with n node points. We also calculate the time and space complexities of the proposed algorithm. Besides these, we study some properties related to edge vertex dominating set of interval graphs.

1.5. *Arrangement of our paper*

We present some notations in the next section. These notations we use in our paper. Section 3 describes some needful properties related to minimum EVDS on interval graphs. We present an optimal algorithm for computing a minimum-cardinality EVDS on interval graph in Sec. 4. Here, we also calculate compilation time and memory spaces of the proposed algorithm.

2. Notations

Here we mention some notations that we use in our paper.

D_{EV} : edge-vertex dominating set.

$|V|$: order of the set V.

$I_i : [a_i, b_i]$.
I_{m_1} : the highest indexed adjacent interval of the least indexed interval.
I_{m_2} : the highest indexed adjacent interval of I_{m_1}.
$n : |V(G)| = n$.
(v_1, v_2) : the edge between the vertices v_1 and v_2.
EVDS : edge vertex dominating set.
EVDN : edge-vertex domination number.
IG : interval graph.

3. Important Properties Related to Minimum Edge-Vertex Dominating Set on Interval Graph

Here, we study some emergent results related to the minimum edge-vertex dominating set on interval graphs.

Lemma 2. *Let I_{m_1} be the highest indexed adjacent interval of the least indexed interval I_x and I_{m_2} be the highest indexed interval among the intervals whose $a_i \in [b_x, b_{m_1}]$. If I_{m_2} does not exist or $m_2 < m_1$, then the edge (v_x, v_{m_1}) is a member of D_{EV}.*

Proof. Suppose I_{m_1} is the highest indexed adjacent interval of the least indexed interval I_x and I_{m_2} is the highest indexed interval among the intervals whose $a_i \in [b_x, b_{m_1}]$. If I_{m_2} does not exist, then $m_1 = n$. Also, in this case, either only two intersecting intervals I_x and I_{m_1} are exist in the region $[\min\{a_x, a_{m_1}\}, b_{m_1}]$ on interval representation, or the intervals whose $a_i < b_x$ and $x < i < m_1$ are exist. So by Lemma 1, $(v_i, v_{m_1}) \in E$ and $(v_x, v_i) \in E$. Therefore, the edge (v_x, v_{m_1}) is essential for finding D_{EV}.

Again if $m_2 < m_1$, then $m_1 = n$, $x < m_2 < m_1$ and $(v_x, v_{m_1}) \in E$. So, $(v_{m_2}, v_{m_1}) \in E$, i.e., I_{m_2} is adjacent with I_{m_1}(using Lemma 1). Again the intervals whose $a_i < b_x$, intersects both the intervals I_x and I_{m_1} as $x < i < m_1$. Now, for the intervals whose $a_i \in [b_x, b_{m_1}]$, $(v_i, v_{m_1}) \in E$, and for the intervals whose $a_i < b_x$, $(v_i, v_{m_1}) \in E$ and $(v_i, v_x) \in E$. So, all the vertices whose corresponding intervals lie in $[\min\{a_x, a_{m_1}\}, b_{m_1}]$ are edge-vertex dominated by the edge (v_x, v_{m_1}). Hence the result. □

Lemma 3. *Let I_{m_1} be the highest indexed adjacent interval of the least indexed interval I_x and I_{m_2} be the highest indexed interval among the intervals whose $a_i \in [b_x, b_{m_1}]$. If I_{m_2} exists and $m_1 < m_2$, then (v_{m_1}, v_{m_2}) is a member of D_{EV}.*

Proof. Let I_{m_1} be the highest indexed adjacent interval of the least indexed interval I_x and I_{m_2} be the highest indexed interval among the intervals whose $a_i \in [b_x, b_{m_1}]$. Now, if I_{m2} exists and $m_1 < m_2$, then the interval whose $a_i \in [b_x, b_{m_1}]$ must intersect the interval I_{m_1}. Again the interval whose $a_i < b_x$ and $x < i < m_1$ intersects both I_x and I_{m_1}, as I_x intersects I_{m_1}. So, the vertex whose corresponding interval having $a_i \in [b_x, b_{m_1}]$ or $a_i < b_x$ must intersect the vertex v_{m_1} corresponding to the interval I_{m_1}.

Besides these, the interval whose $a_i \in [b_{m_1}, b_{m_2}]$ must intersect the interval I_{m_2}. So, the edge (v_{m_1}, v_{m_2}) is a member of the edge-vertex dominating set D_{EV}. Hence the result is proved. □

Lemma 4. *Let I_n be the highest indexed interval among the intervals whose $a_i \in [b_{m_1}, b_{m_2}]$. If at any stage (v_{m_1}, v_{m_2}) is a currently selected member of the set D_{EV} and there exists at least one interval whose $a_i > b_{m_2}, i < n$, then (v_{m_2}, v_n) will be the next member of the set D_{EV}.*

Proof. Suppose I_n is the highest indexed interval among the intervals whose $a_i \in [b_{m_1}, b_{m_2}]$. If at any stage (v_{m_1}, v_{m_2}) is the currently selected member of the set D_{EV}, intervals whose $a_i \in [b_{m_1}, b_{m_2}]$, must intersect I_{m_2}. Now, if there exist at least one interval whose $a_i > b_{m_2}, i < n$, then it does not intersect I_{m_2} but it intersects I_n, as $m_2 < i < n$ and $(v_{m_2}, v_n) \in E$ (using Lemma 1). So, v_i is not edge-vertex dominated by the edge (v_{m_1}, v_{m_2}). Therefore, one more edge is required. For this reason, (v_{m_2}, v_n) will be the next member of edge-vertex dominating set D_{EV}. Hence the result. □

Lemma 5. *Let at any stage (except the initial stage), I_x is the least indexed unmarked interval and I_n be the highest indexed interval among those intervals whose $a_i \in [b_{\mathrm{last}}, b_x]$, where b_{last} is the right endpoint of the last scanning region. Then (v_x, v_n) is a member of edge-vertex dominating set D_{EV}.*

Proof. We assume that (v_{m1}, v_{m2}) is the currently selected member of D_{EV} at a particular stage. So, all the vertices corresponding to the intervals whose $a_i < b_{m2}$ are EV-dominated by at least one member of EVDS D_{EV}. Now, let I_x be the least indexed unmarked interval that is not edge-vertex dominated by D_{EV} and I_n is the highest indexed interval among those intervals whose $a_i \in [b_{\mathrm{last}}, b_x]$, where b_{last} is the right endpoint of the last scanning region. Then $b_{m2} = b_{\mathrm{last}} < a_x$. So, all the vertices corresponding to the intervals whose $a_i > b_{m2}$, are not edge-vertex dominated by (v_{m1}, v_{m2}). So, one more edge is needed to edge-vertex dominates all vertices and that edge is (v_x, v_n). Hence, the result is proved. □

4. Algorithm for Computing Minimum EVDS on Interval Graph

Now, we are going to present an algorithm to find a minimum EVDS on IG in $O(n)$ time. From the results stated in the previous section, we have observed that if m_2 does not exist or $m_2 < m_1$, then (v_x, v_{m_1}) is a member of edge-vertex dominating set D_{EV} and if $m_1 < m_2$, then (v_{m1}, v_{m2}) is a member of edge-vertex dominating set D_{EV}. Also, the edges (v_{m_2}, v_n) and (v_x, v_n) can be considered a member of EVDS D_{EV} depending on the situation discussed in Lemmas 4 and 5, respectively.

The complete Algorithm **EVDSIG**, designed for computing a minimum EVDS D_{EV} of interval graph G is presented below.

Algorithm EVDSIG

Input: An interval representation of an interval graph G.

Output: Minimum-cardinality EVDS D_{EV}.

Step 1: Find the set M_1 of the intervals whose $a_i < b_1$ and mark them.

Step 2: Set $m_1 =$ highest index of the intervals of M_1.

Step 3: Find the set M_2 of the unmarked intervals whose $a_i \in [b_1, b_{m_1}]$ and mark them.

Step 4: Set $m_2 =$ highest index of the intervals of M_2.

Step 5: If m_2 does not exist or $m_2 < m_1$, then
find $D_{\mathrm{EV}} = D_{\mathrm{EV}} \cup \{(v_1, v_{m_1})\}$ and exit. //Lemma 2//
Else
go to the next step.
End If

Step 6: Find $D_{\mathrm{EV}} = D_{\mathrm{EV}} \cup \{(v_{m_1}, v_{m_2})\}$.//Lemma 3//

Step 7: Find the set M_3 of the unmarked intervals whose $a_i \in [b_{m_1}, b_{m_2}]$ and mark them.

Step 8: Set $m_3 =$ highest index of the intervals of M_3.

Step 9: If $m_3 = n$, then
find $D_{\mathrm{EV}} = D_{\mathrm{EV}} \cup \{(v_{m_2}, v_{m_3})\}$ and exit. //Lemma 4//
Else
go to the next step.
End If

Step 10: Select the least indexed unmarked interval, say I_x.

Step 11: Find the set M_{11} of the unmarked intervals whose $a_i \in [b_{m_2}, b_x]$ and mark them.

Step 12: Set $m_{11} =$ highest index of the intervals of M_{11}.

Step 13: If $m_{11} = n$ then
find $D_{\mathrm{EV}} = D_{\mathrm{EV}} \cup \{(v_x, v_{m_{11}})\}$ and exit. //Lemma 5//
Else if $m_{11} < m_3$, then
$m_{11} = m_3$.
Else
go to the next step.
End If

Step 14: Find the set M_{22} of the unmarked intervals whose $a_i \in [b_x, b_{m_{11}}]$ and mark them.

Step 15: Set $m_{22} =$ highest index of the intervals of M_{22}.

Step 16: Find $D_{\mathrm{EV}} = D_{\mathrm{EV}} \cup \{(v_{m_1 1}, v_{m_{22}})\}$.

Step 17: Find the set M_{33} of the unmarked intervals whose $a_i \in [b_{m_{11}}, b_{m_{22}}]$ and mark them.

Step 18: Set $m_{33} =$ highest index of the intervals of M_{33}.
Step 19: If m_{33} exists then $m_3 = m_{33}, m_1 = m_{11}, m_2 = m_{22}$ and go to Step 9.
Else exit.
End if
End EVDSIG.

If we apply the above Algorithm **EVDSIG** for the IG G of Fig. 1, then output will be a minimum EVDS $D_{\text{EV}} = \{(v_3, v_5), (v_9, v_{10})\}$ of G.

Lemma 6. *The Algorithm* **EVDSIG** *gives a minimum EVDS of interval graph G.*

Proof. In Step 5 of Algorithm *EVDSIG*, by Lemma 2, we set up $D_{\text{EV}} = D_{\text{EV}} \cup \{(v_1, v_{m_1})\}$, where $m_1 = n$ is highest index of the intervals whose $a_i < b_1$, depending upon the condition that if m_2 does not exist or $m_2 < m_1$, where m_2 is highest index of the intervals whose $a_i \in [b_1, b_{m1}]$. So, edge-vertex dominating set D_{EV} is minimum as its cardinality is one. Again if $m_2 > m_1$ (in Step 6), then applying the result of Lemma 3 we set up $D_{\text{EV}} = D_{\text{EV}} \cup \{(v_{m_1}, v_{m_2})\}$. Here, we select the edge (v_{m_1}, v_{m_2}) as a member of D_{EV} because the interval I_{m_1} is the highest indexed adjacent interval of I_1 and I_{m_2} is the highest indexed adjacent interval of I_{m_1}, i.e., the edge (v_{m_1}, v_{m_2}) edge-vertex dominates the maximum number of vertices. Now, if $m_3 = n$ (Step 3), we set up $D_{\text{EV}} = D_{\text{EV}} \cup \{(v_{m2}, v_{m3})\}$, by Lemma 4, because the new selected member edge-vertex dominates the all vertices that are not edge-vertex dominated by the edge (v_{m_1}, v_{m_2}). So, in that case, D_{EV} is minimum. Again, in Step 13, if $m_{11} = n$, then we include (by Lemma 5) the edge $(v_x, v_{m_{11}})$ as a member of D_{EV} because all the vertices that are not edge-vertex dominated by the previous members of D_{EV} are now edge-vertex dominated by the new selected edge. So, in that case, D_{EV} is also minimum. In Step 16, if $m_{11} > m_3$, then we select (by Lemma 3) the edge (v_x, v_{m11}) as a member of D_{EV} because $I_{m_{11}}$ is the highest indexed adjacent interval of I_x, i.e., the edge $(v_x, v_{m_{11}})$ edge-vertex dominates the maximum number of vertices.

So, in the above cases, we have selected the members of D_{EV} so that the selected member edge-vertex dominates the maximum number of node points. We follow the same rule throughout the algorithm EVDSIG. So, the final D_{EV} is a minimum EVDS of the IG G. □

Theorem 1. *The Algorithm* **EVDSIG** *takes $O(n)$ time to find a EVDS of IG, where $n = |V|$.*

Proof. Step 1 needs $O(|N[1]|)$ time for finding the set M_1 as $|M| = N[1]$. Step 2 requires $O(|N[1]|)$ time to compute m_1. Again, Step 3 finds the set M_2 in $O(|M_2|)$ time, where $|M_2| \leq n$. Like Step 3, $O(|M_2|)$ time is needed to find m_2 in Step 4. Now, Step 5 takes constant time. Also, Step 6 takes constant time to select a new member of D_{EV}. Step 7 needs $O(|M_3|)$ time, where $|M_3| \leq n$ for computing set M_3. Like Step 7, Step 8 takes $O(|M_2|)$ time to find m_3. The set D_{EV} can be updated in constant time in Step 9. Step 10 needs constant time. Again, Step 11 finds the set M_{11} in $O(|M_{11}|)$ time, where $|M_{11}| \leq n$. Like Step 11, $O(|M_{11}|)$ time is needed to find m_{11} in Step 12. Now, in Step 13, D_{EV} can be updated in constant time. Again, Step 14 takes $O(|M_{22}|)$ time to find the set M_{22}, where $|M_{22}| \leq n$. Like Step 14, Step 15 needs $O(|M_{22}|)$ time to find m_{22}. In Step 16, a new member of D_{EV} can be selected in constant time. In Step 17, for finding set M_{33}, we need $O(|M_{33}|)$ time, where $|M_{33}| \leq n$ and for finding m_{33}, $O(|M_{33}|)$ time is needed in Step 18. Also, Step 19 can be finished in constant time. It is clear that sets M_1, M_2, M_3, M_{11}, M_{22} and M_{33} are mutually disjoint. So, the total time required to compute all the members of the set D_{EV} is $O(n)$ time. Therefore, the overall compilation time of the algorithm **EVDSIG** is $O(n)$. □

Theorem 2. *$O(n)$ memory space is needed to compile the Algorithm* **EVDSIG**.

Proof. In the interval representation of an IG, there is a set of intervals I_x, where $x = 1, 2, \ldots, n$. Now, one can store the set of interval s in $O(n)$ space. Also, we can store the sets M_1, M_2, M_3, M_{11}, M_{22} and M_{33} in $O(n)$ space as these sets are mutually disjoint. Besides these for finding the highest index of the intervals of sets M_1, M_2, M_3, M_{11}, M_{22} and M_{33}, $O(n)$ space is needed. Furthermore, minimum EVDS D_{EV} can be stored in $O(n)$ space as $|D_{\text{EV}}| \leq n$. Therefore, the total space complexity of the algorithm **EVDSIG** is $O(n)$. □

5. Conclusion

In graph theory, domination plays a major role in the vulnerability analysis of communication networks modeled by graphs. There are many types of domination depending on the characteristics of dominating sets. Edge-vertex domination has a very wide range of practical-life applications side also. It can be applied in placing tracking devices such as cameras to protect facilities in a given area. In this paper, we formulate an $O(n)$ time algorithm for determining a minimum EVDS on interval graphs. We also study some properties related to the taken problem. In the future, we plan to design an optimal algorithm to determine a minimum EVDS of the social network graph, permutation graph, circular-arc graph and other intersection graphs.

References

[1] A. S. Adhya, S. Mondal and S. C. Barman, An optimal algorithm to find minimum k-hop connected dominating set of permutation graphs, *Asian-Eur. J. Math.* **14**(4) (2021) 2150049 (1-12), doi:10.1142/S1793557121500492.

[2] S. K. Ayyaswamy, B. Krishnakumari, C. Natarajan and Y. B. Venkatakrishnan, Bounds on the hop domination number of a tree, *Proc. Math. Sci. Indian Acad. Sci.* **125**(4) (2015) 449–455.

[3] A. D. Atri and M. Moscarini, Distance-hereditary graphs, Steiner trees, and connected domination, *SIAM J. Comput.* **17**(3) (1988) 521–538, doi/10.1137/0217032.

[4] S. Balamurugan, Changing and unchanging isolate domination: Edge removal, *Discrete Math. Algorithms Appl.* **9**(1) (2017) 1750003 (1-9), doi:10.1142/S1793830917500033.

[5] C. Berge, *The Theory of Graphs and its Applications* (Methuen and Co London, 1962).

[6] S. C. Barman, M. Pal and S. Mondal, An optimal algorithm to find minimum k-hop dominating set of interval graphs, *Discrete Math. Algorithms Appl.* **11**(2) (2019) 1950016 (1-18), https://doi:10.1142/S1793830919500162.

[7] P. Basuchowdhuri and S. Majumder, Finding influential nodes in social networks using minimum k-hop connected dominating set, *Proc. Int. Conf. Appl. Algorithms* **8321** (2014) 137–151, https://doi:10.1007/978-3-319-04126-1-12.

[8] M. C. Carlisle and E. L. Loyd, On the k-coloring of intervals, Lecture Notes in Computer Science, Vol. 497 (Springer, Berlin, 1991), pp. 90–101.

[9] G. J. Chang, Algorithmic aspect of domination in graphs, in *Handbook of Combinatorial Optimization*, 2nd edn. (Springer-Verlag, New York, 2013), pp. 339–405.

[10] E. J. Cockayne, R. M. Dawes and S. T. Hedetniemi, Total domination in graphs, *Networks* **10**(3) (2006) 211–219, https://doi/10.1002/net.3230100304.

[11] D. Cokuslu and K. Erciyes, A hierarchical connected dominating set based clustering algorithm for mobile *ad hoc* networks, in *Proc.* 15*th Annual Meeting of the IEEE International Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (IEEE, 2007), pp. 61–66, https://doi/10.1109/MASCOTS.2007.1.

[12] H. S. Chao, F. R. Hsu and R. C. T. Lee, An optimal algorithm for finding the minimum cardinality dominating-set on permutation graphs, *Discrete Appl. Math.* **102** (2000) 159–173, doi:10.1016/S0166-218X(98)00145-0.

[13] P. Dorbec, M. A. Henning, M. Montassier and J. Southey, Independent domination in cubic graphs, *J. Graph Theory* **80**(4) (2015) 329–349, doi.org/10.1002/jgt.21855.

[14] E. D. Demaine, M. Hajjaghay and D. M. Thilikos, Fixed parameter algorithm for (k, r) center in planar graphs and map graphs, *ACM Trans. Algorithm* **1**(1) (1862) 1–16.

[15] P. Duraisamy and S. Esakkimuthu, Linear programming approach for various domination parameters, *Discrete Mathematics, Algorithms Appl.* **13**(1) (2021) 2050096, doi:10.1142/S1793830920500962.

[16] O. Favaron, M. A. Henning, C. M. Mynhart and J. Puech, Total domination in graphs with minimum degree three, *J. Graph Theory* **34**(1) (2000) 9–19, https://doi:10.1002/(SICI)1097-0118(200005)34:1<9::AID-JGT2>3.0.CO;2-O.

[17] J. Fabri, *Automatic Storage Optimization* (UMI Press Ann Arbor, MI, 1982).

[18] X. Gao, W. Wu, X. Zhang and X. Li, A constant-factor approximation for d-hop connected dominating sets in unit disk graph, *Int. J. Sensor Netw.* **12**(3) (2012) 125–136, doi.org/10.1504/IJSNET.2012.050447.

[19] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, 2nd edn. (Academic Press Elsevier, Amsterdam, 2004).

[20] P. Gupta, A. Goyal and S. Arumugam, Line-set domination in graphs, *Discrete Math. Algorithms Appl.* (2022), doi:10.1142/S1793830922501178.

[21] I. Sahul Hamid and S. Balamurugan, Vertex criticality with respect to isolate domination, *Discrete Math. Algorithms Appl.* **7**(2) (2015) 1550010, doi:10.1142/S179383091550010X.

[22] A. Hashimoto and J. Stevens, Wire routing by optimizing channel assignment within large apertures, in *Proc. 8th IEEE Design Automation Workshop*, 1971, pp. 155–169, https://dl.acm.org/doi/pdf/10.1145/800158.805069.

[23] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Fundamentals of Domination in Graphs*: *Selected Topics* (Marcel Dekker, New York, 1998).

[24] T. W. Haynes, C. M. Mynhardt and L. C. Vander Merwe, Criticality index of total domination, *Congr. Numer.* **131** (1998) 67–73.

[25] T. W. Haynes, C. M. Mynhardt and L. C. Van der Merwe, Total domination edge critical graphs, *Utilitas Math.* **54** (1998) 229–240.

[26] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Domination in Graphs-Advanced Topics* (Marcel Dekker, Inc., New York, 1998).

[27] T. W. Haynes, C. M. Mynhardt and L. C. Vander Merwe, Total domination edge critical graphs with maximum diameter, *Discussiones Math. Graph Theory* **21** (2001) 187–205.

[28] J. R. Jungck, O. Dick and A. G. Dick, Computer-assisted sequencing, interval graphs and molecular evolution, *Biosystem* **15** (1982) 259–273.

[29] C. X. Kang and E. Yi, Bounds on the sum of domination number and metric dimension of graphs, *Discrete Math. Algorithms Appl.* **10**(5) (2018) 1850066 (1-15), doi:10.1142/S1793830918500660.

[30] B. Krishnakumari, Y. B. Venkatakrishnan and M. Krzywkowski, On trees with total domination number equal to edge-vertex domination number plus one, *Proc. Indian Acad. Sci.* **126** (2016) 153–157.

[31] K. Kim, Edge-vertex domination in trees, *Discrete Math. Algorithms Appl.* **14**(8) (2022) 2250043, doi:10.1142/S1793830922500434.

[32] S. Kundu and S. Majumder, A linear-time algorithm for optimal k-hop connected dominating set of a tree, *Inf. Process. Lett.* **116** (2016) 197–202, doi:10.1016/j.ipl.2015.07.014.

[33] M. Kamal Kumar, L. Sudershan and F. Reddy, Inverse roman domination number in graphs, *Discrete Math. Algorithms Appl.* **5**(3) (2013) 1350011 (1-4), doi:10.1142/S1793830913500110.

[34] J. R. Lewis, Vertex-edge and edge-vertex domination in graphs, Ph.D. thesis, Clemson University (2007).

[35] M. A. Meybodi, k-efficient domination: Algorithmic perspective, *Discrete Math. Algorithms Applications* **14**(8) (2022) 2250051, doi:10.1142/S1793830922500513.

[36] T. N. Nguyen and D. T. Huynh, Connected d-hop dominating sets in mobile *ad hoc* networks, in *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, IEEE, 2006, pp. 1–8, doi:10.1109/WIOPT.2006.1666454.

[37] C. Natarajan and S. K. Ayyaswamy, Hop domination in graphs-II, *An. Stt. Univ. Ovidius Constanta* **23**(2) (2015) 187–199, doi:10.1515/auom-2015-0036.

[38] T. Ohtsuki, H. Mori, E. S. Khu, T. Kashiwabara and T. Fujisawa, One dimensional logic gate assignment and interval graph, *IEEE Trans. Circuits Syst.* **26** (1979) 675–684.

[39] O. Ore, Theory of graphs, *Amer. Math. Soc. Colloq. Pub., Providence* **38** (1962) 206–212, source : https://doi.org/10.1090/coll/038.

[40] S. Olariu, An optimal greedy heuristic to color interval graphs, *Inf. Process. Lett.* **37** (1991) 21–25.

[41] K. Peters, Theoretical and algorithmic results on domination and connectivity (nordhaus-gaddum, gallai type results, max-min relationships, linear time, series-parallel) (Ph.D. thesis) Clemson University, Clemson, SC, USA (1986).

[42] P. Roushini Leely Pushpam, B. Mahavir and M. Kamalam, Resolving roman domination in graphs, *Discrete Math. Algorithms Appl.* **14**(7) (2022) 2250028, doi: 10.1142/S1793830922500288.

[43] A. Rana, A. Pal and M. Pal, An efficient algorithm to solve the distance k-domination problem on permutation graphs, *J. Discrete Math. Sci. Cryptogr.* **19**(2) (2016) 241–255, doi:10.1080/09720529.2014.986906.

[44] A. Rana, A survey on the domination of fuzzy graphs, *Discrete Math. Algorithms Appl.* **13**(1) (2021) 2130001 (1-20), doi:10.1142/S1793830921300010.

[45] M. Q. Rieck, S. Pai and S. Dhar, Distributed routing algorithms for multi-hop *ad hoc* networks using d-hop connected d-dominating sets, *Comput. Netw.* **47**(6) (2005) 785–799, https://doi:10.1016/j.comnet.2004.09.005.

[46] R. S. Shaheen, Bounds for the 2-Domination number of toroidal grid graphs, *Int. J. Comput. Math.* **86**(4) (2009) 584–599, https://doi:10.1080/00207160701690284.

[47] B. Ahin and A. Ahin, Double edge-vertex domination, in *Int. Conf. Intelligent and Fuzzy Systems* (Springer, 2020), pp. 1564–1572.

[48] A. Sahin and B. Sahin, Total edge-vertex domination, *RAIRO Theor. Inf. Appl.* **54**(1) (2020) 1-6, source: doi 10.1051/ita/2020001.

[49] E. Sampathkumar and H. B. Walikar, Connected domination number of a graph, *J. Math. Phys. Sci.* **13**(6) (1979) 607–613.

[50] B. Senthilkumar, H. Naresh Kumar and Y. B. Venkatakrishnan, Bounds on total edge domination number of a tree, *Discrete Math. Algorithms Appl.* **13**(2) (2021) 2150011 (1-8), https://doi:10.1142/S1793830921500117.

[51] P. J. Slater, R-domination in graphs, *J. ACM* **23**(2) (1976) 446–450, https://doi: 10.1145/321958.321964.

[52] D. P. Sumner and P. Blitch, Domination critical graphs, *J. Combin. Theory Ser. B* **34** (1983) 65–76.

[53] Y. B. Venkatakrishnan and B. Krishnakumari, An improved upper bound of edge-vertex domination of a tree, *Inf. Process. Lett.* **134** (2018) 14–17, doi:10.1016/j.ipl.2018.01.012.

[54] T. H. P. Vuong and D. T. Huynh, Adapting d-hop dominating sets to topology changes in *ad hoc* networks, in *Proc. 9th Int. Conf. Computer Communications and Network s*, pp. 348–353, Publisher IEEE, Year: 2006, Conference Location: Phoenix, AZ, USA, doi:10.1109/ICCCN.2000.885513.

[55] S. Yau and W. Gao, Multi-hop clustering based on neighborhood benchmark in mobile *ad-hoc* networks, *Mobile Netw. Appl.* **12** (2007) 381–391, doi:10.1007/s11036-008-0039-3.