Communications in Mathematics and Applications
Vol. 11, No. 2, pp. 259–269, 2020
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications

http://www.rgnpublications.com

DOI: 10.26713/cma.v11i2.1356

Research Article

Multiple Intruder Locating Dominating Sets in Graphs: An Algorithmic Approach

K. Venugopal[1*] and K. A. Vidya[2]

[1] *Department of Mathematics, R.V. College of Engineering, Bengaluru 560059, Karnataka, India*
[2] *Department of Mathematics, Dayananda Sagar Academy of Technology and Management, Bengaluru 560082, Karnataka, India*
**Corresponding author:* venugopalk@rvce.edu.in

Abstract. A set $S \subseteq V$ of vertices (called codewords) of a graph $G = (V,E)$ is called a Multiple Intruder Locating Dominating set ($MILD$ set) if every non-codeword v is adjacent to at least one codeword u which is not adjacent to any other non-codeword. This enables one to locate intruders at multiple locations of a network and hence the name. The $MILD(G)$ is the minimum cardinality of a $MILD$ set in G. Here, we show that the problem of finding MILD set for general graphs is NP-Complete. Further, we provide a linear time algorithm to find the MILD number of trees through dynamic programming approach and then, we extend the algorithm for unicyclic graphs.

Keywords. Complexity; Approximation; Multiple intruder locating domination; Chordal graph; Split graph; Bipartite graph

MSC. 05C69; 05C85

Received: January 22, 2020 **Accepted:** March 11, 2020

1. Introduction

Many varieties of Locating Dominating sets have been developed in recent times keeping in mind the safeguarding of facilities in case of the presence of an intruder. After introducing and working on the locating dominating sets [10], together with S. J. Seo, P. J. Slater has worked on a variation called Open Neighborhood Locating dominating sets [7] where the possibility that the intruder might harm a detecting device and prevent it in detecting intruder at its own location is considered. The possibility that a detecting device could malfunction was also considered by P. J. Slater which led to another variation 'Fault Tolerant Locating Dominating sets' [9]. A closely related problem of 'Identifying codes' where a detection device detects an intruder in its closed neighborhood but not the exact location was introduced by Chakrabarty *et al*. [5]. For some recent works on locating domination parameters (refer [3, 4, 6, 8]).

It was only a matter of time that a variation of the locating domination problem would be developed where multiple intruders can be located in a facility, and that has been done in [11]. Given a graph $G = (V,E)$, a set $S \subseteq V$ is called a *Multiple Intruder Locating Dominating* (MILD) set if

$$\forall\ u \in [V \setminus S], \exists v \in S \ \text{ such that } \ N(v) \cap [V \setminus S] = \{u\}$$

In that case, the vertex v is called a devout dominator of u, and in turn, u is called the secure non-codeword of v. The MILD number for a graph G is proved to lie between $n/2$ and $n-1$, both inclusive. The characterization of the graphs attaining those bounds are also discussed in [11]. Naturally, one would be interested to develop an algorithm to find the MILD number of any given graph G. The next section proceeds on this question.

It is to be noted that a locating dominating set secures a network in the presence of one intruder. As stated before, Multiple intruder locating domination aims at securing a network even when there are intruders at multiple locations in a network, possibly at all of them. To achieve that, it might be tempting to think that a sensor might be required at all the locations. But a MILD set shows that that need not be the case. In fact, for an n location network, a MILD set can get as low as $n/2$. Thus, though the MILD number is higher when compared with the other locating domination parameters, it *is justified* since the problem of MILD set achieves something more than the other parameters.

2. MILD Problem in Arbitrary Graphs

We show that the MILD problem is NP-Complete for arbitrary graphs. This is done by reducing MILD problem to 3-SAT problem which is proved to be NP-Complete by Garey and Johnson in [2].

3-SAT Problem

INSTANCE: Collection $C = \{c_1, c_2, \dots, c_M\}$ of clauses on set $U = \{u_1, u_2, \dots, u_N\}$ such that $|c_i| = 3$ for $1 \le i \le M$.
QUESTION: Is there a satisfying truth assignment for C?

Multiple-Intruder-Locating-Domination (MILD) Problem

INSTANCE: Graph $G = (V,E)$ and positive integer $K \le |V|$.
QUESTION: Is $MILD(G) \le K$?

Theorem 1. *Problem MILD is NP-complete.*

Proof. It can be easily seen that $MILD \in NP$. We give a polynomial reduction from 3-SAT to MILD. For the given instance of 3-SAT with a given U and C, construct a graph G as follows:

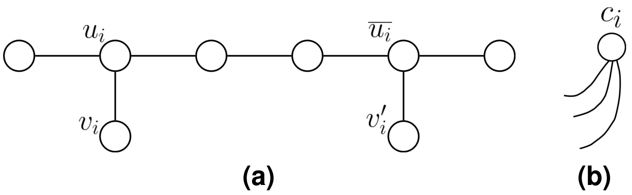

Figure 1. Variable and clause graphs

- For each u_i construct the graph G_i as shown in Figure 1a.
- For each clause c_i create the vertex c_i as shown in Figure 1b.
- Suppose a clause $c_j = \{u_{j,1}, u_{j,2}, u_{j,3}\}$ where each $u_{j,t}$ is either u_i or $\overline{u}_i$, let the vertex c_j be adjacent to $\{v_{j,1}, v_{j,2}, v_{j,3}\}$ where each $v_{j,t}$ is either v_i or v_i'.
- Suppose a u_i or $\overline{u}_i$ is repeated in different clauses, then all the u_i vertices are made adjacent to each other and the corresponding $\overline{u}_i$ vertices are also made adjacent to each other. An example where u_1 is repeated in the clauses c_1 and c_2 is shown in Figure 2.
- Finally, connect all the clause vertices with each other so as to form a clique.

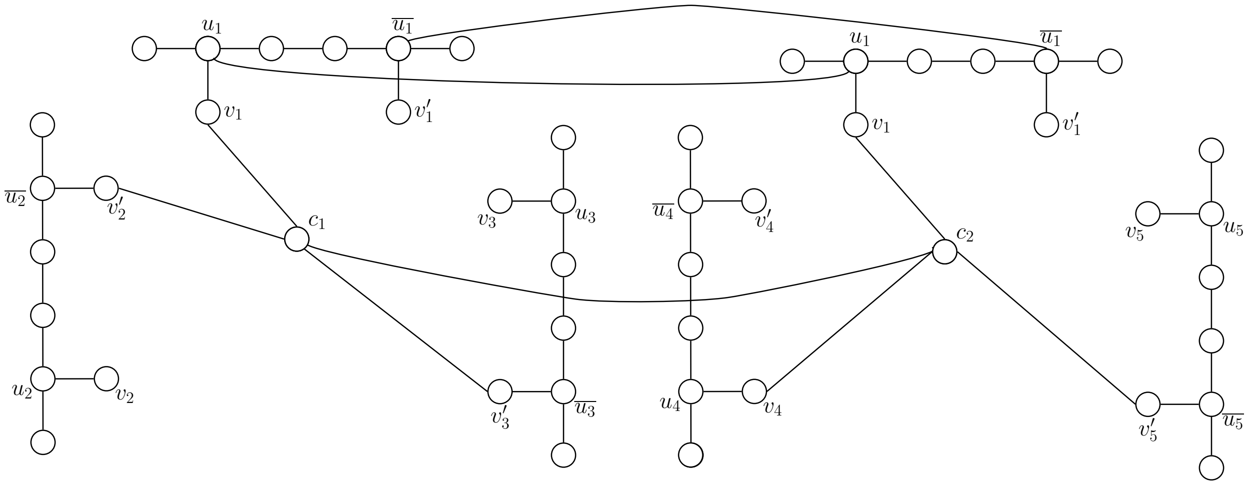

Figure 2. An example of repeated u_i

Every vertex c_i is adjacent to three vertices each of which is a part of the graph containing 8 vertices. Thus there are $(8 \times 3 + 1)M = 25M$ vertices. Suppose vertices $u_{p,1}, u_{p,2}, \ldots, u_{p,l}$ (each of them is some u_i or $\overline{u}_i$) appear more than once, say, $r_1, r_2, \ldots, r_l$ times respectively. If $r = C(r_1,2)+C(r_2,2)+\ldots+C(r_l,2)$ then G contains $(7\times3+3)M+\frac{M}{2}(M-1)+2r = 24M+\frac{M}{2}(M-1)+2r$ edges and hence G can be constructed from C in polynomial time.

Given a truth assignment for C, if u_i is true then the vertex $u_i \in S$ in G_i else, $\overline{u}_i \in S$. In G_i, four more vertices must be included in S correspondingly as shown in Figures 3a and 3b. It follows that $MILD(G) = (5 \times 3)M = 15M$ if C has a satisfying truth assignment.

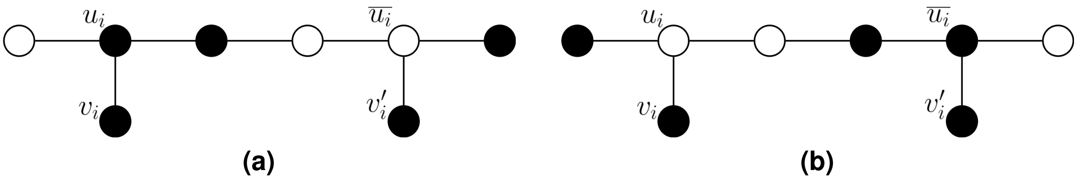

Figure 3. Formation of MILD set from a given truth assignment

Conversely, given that $MILD(G) = 15M$. Then from the given MILD set, the truth assignment is done for C as follows (without loss of generality):

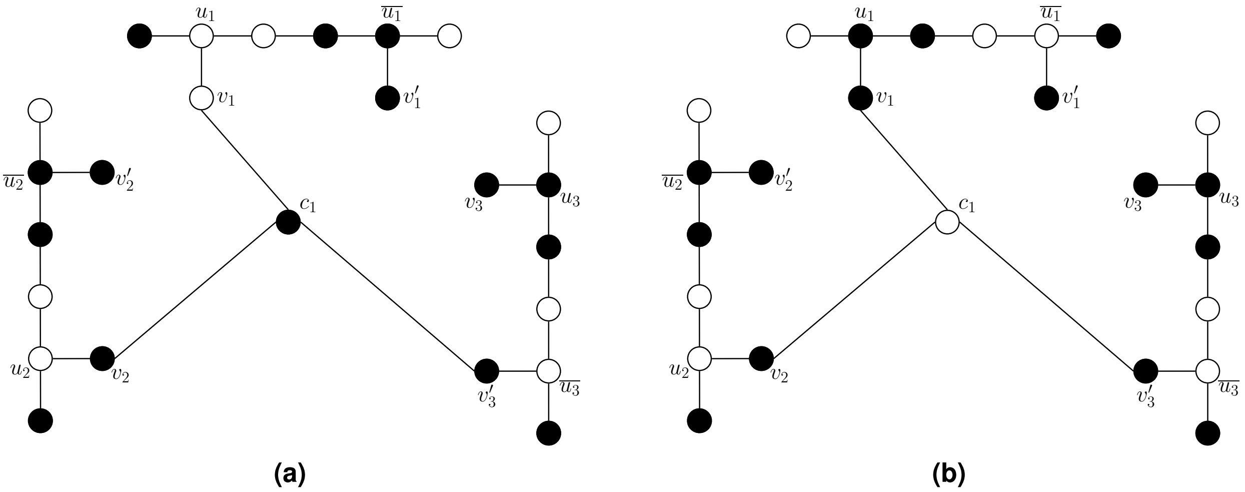

Figure 4. The exception case and the modification

- If both $u_{j,t}$ and $v_{j,t}$ are codewords, then assign *true* to the element $u_{j,t}$.
- If $u_{j,t}$ is a non-codeword and $v_{j,t}$ is a codeword, then assign *false* to the element $u_{j,t}$.

Since all the clause vertices c_i form clique and $MILD(G) = 15M$, either all of them are (secure) non-codewords or all of them are (devout-dominating) codewords. In the former case, it can be easily seen that there exists a satisfying truth assignment. In the latter case, an exception arises, for example, as shown in Figure 4a. In such a case, the question arises as to what truth value is to be assigned for u_1 (without loss of generality). To address this, we make the following changes in the graph, as illustrated in Figure 4b:

1: Interchange the black and white vertices in the path P_6 containing u_1 and $\overline{u}_1$.
2: Make v_1 codeword and c_1 non-codeword.

Now, all the c_i vertices would be (secure) non-codewords and the satisfying truth assignment can be made. Thus, C has a satisfying truth assignment whenever $MILD(G) = 15M$. This completes the proof. □

3. MILD Sets in Trees

Let $\gamma_{ml}(G)$ be the minimum size of a Multiple Intruder Locating Dominating set of the graph G. i.e.,

$$\gamma_{ml}(G) = \min\{|S|;\ S \text{ is a MILD set of } G\}.$$

Consider a graph G (a tree) with a chosen vertex u. A MILD set of G either contains or does not contain u. Then:

$$\gamma^1_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G \text{ and } u \in S\},$$
$$\gamma^0_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G \text{ and } u \notin S\}.$$

Lemma 2 ([1]). $\gamma_{ml}(G) = \min\{\gamma^1_{ml}(G,u), \gamma^0_{ml}(G,u)\}$ *for any graph G with a specific vertex u.*

Also, we define

$$\gamma^{00}_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G-u \text{ with no devout-dominator adjacent to } u\}$$

Definition 1. A vertex $u \in G$ is said to be "*dd-available*" from G if $\nexists\ w \in N(u) \cap V(G) \backslash S$.

A vertex $u \in S$ might be or might not be dd-available from G. Suppose it is not dd-available, then it might be or might not be a devout-dominator of some vertex inside G. In order to cover these cases, we introduce three new problems:

$$\gamma^{11}_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G, u \in S \text{ and } N(u) \cap [V(G)\backslash S] = \phi\}$$
$$\gamma^{100}_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G, u \in S \text{ and } u \text{ is a devout-dominator of some vertex } w \in V(G)\backslash S\}$$
$$\gamma^{101}_{ml}(G,u) = \min\{|S|;\ S \text{ is a MILD set of } G, u \in S,\ N(u) \cap [V(G)\backslash S] \neq \phi \text{ but } u \text{ is not a devout-dominator}\}$$

Theorem 3. *Suppose G and H are graphs with specific vertices u and v, respectively. Let I be the graph with the specific vertex u, which is obtained from the disjoint union of G and H by joining a new edge uv. Then the following statements hold:*

(1) $\gamma^1_{ml}(I,u) = \min\{\gamma^{11}_{ml}(G,u) + \min[\gamma^{00}_{ml}(H,v), \gamma_{ml}(H)], \gamma^{100}_{ml}(G,u) + \gamma^1_{ml}(H,v), \gamma^{101}_{ml}(G,u) + \gamma_{ml}(H)\}$

(2) $\gamma^0_{ml}(I,u) = \min\{\gamma^0_{ml}(G,u) + \min[\gamma^0_{ml}(H,v), \gamma^{101}_{ml}(H,v), \gamma^{11}_{ml}(H,v)], \gamma^{00}_{ml}(G,u) + \gamma^{11}_{ml}(H,v)\}$

(3) $\gamma^{00}_{ml}(I,u) = \gamma^{00}_{ml}(G,u) + \min[\gamma^0_{ml}(H,v), \gamma^{101}_{ml}(H,v)]$

(4) $\gamma^{100}_{ml}(I,u) = \min\{\gamma^{100}_{ml}(G,u) + \gamma^1_{ml}(H,v), \gamma^{11}_{ml}(G,u) + \gamma^{00}_{ml}(H,v)\}$

(5) $\gamma^{101}_{ml}(I,u) = \min\{\gamma^{101}_{ml}(G,u) + \gamma_{ml}(H,v), \gamma^{11}_{ml}(G,u) + \gamma^0_{ml}(H,v)\}$

(6) $\gamma^{11}_{ml}(I,u) = \gamma^{11}_{ml}(G,u) + \gamma^1_{ml}(H,v)$

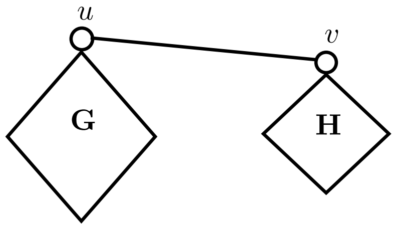

Figure 5. Formation of the graph I using the graphs G and H

Proof. (1) This follows from the fact that S is a MILD set of I with $u \in S$if and only if $S = S' \cup S''$ where, S' is a MILD set of G with $u \in S'$ and:

(i) if u is dd-available from S', then S'' is a MILD set of H or $H - v$.

(ii) if u is not dd-available from S' and devout-dominates some vertex in G, then S'' is a MILD set of H with $v \in S''$.

(iii) if u is not dd-available and is not a devout-dominator, then S'' is a MILD set of H.

(2) This follows from the fact that S is a MILD set of I with $u \notin S$ if and only if $S = S' \cup S''$, where

(i) S' is a MILD set of G with $u \notin S'$ and S'' is a MILD set of H with $u \notin S''$ or, with $u \in S''$ and u is not a devout-dominator or, with $u \in S''$ and is dd-available,

or

(ii) S' is a MILD set of $G - u$ and S'' is a MILD set of H with $v \in S''$ and v is dd-available.

(3) This follows from the fact that S is a MILD set of $I - u$ with u not being a secure neighbour if and only if $S = S' \cup ''$, where S' is a MILD set of $G - u$ with u not being a secure neighbour and, S'' is a MILD set of H with $v \notin S''$ or, with $v \in S''$ and v is not a devout-dominator.

(4) This follows from the fact that S is a MILD set of I with $u \in S$ and u devout-dominates a vertex in I (and hence not dd-available from I) if and only if $S = S' \cup S''$, where

(i) S' is a MILD set of G with $u \in S'$ and u devout-dominates a vertex in G and, S'' is a MILD set of H with $v \in S''$,

or

(ii) S' is a MILD set of G with $u \in S'$ and is dd-available and, S'' is a MILD set of $H - v$.

(5) This follows from the fact that S is a MILD set of I with $u \in S$ and $\exists\, w \in N(u) \cap [V(I) - S]$ such that u is not a devout-dominator of w if and only if $S = S' \cup S''$, where

(i) S' is a MILD set of G with $u \in S'$ and $\exists\, w \in N(u) \cap [V(G) - S']$ such that u is not a devout-dominator of w and, S'' is a MILD set of H,

or

(ii) S' is a MILD set of G with $u \in S'$ and is dd-available and, S'' is a MILD set of H with $v \notin S''$.

(6) This follows from the fact that S is a MILD set of I with $u \in S$ and is dd-available from I if and only if $S = S' \cup S''$, where S' is a MILD set of G with $u \in S'$ and is dd-available from G, and S'' is a MILD set of H with $v \in S''$.

Hence the proof. □

By using the lemmas and the theorem above, we get the following dynamic programming algorithm for the Multiple Intruder Locating Domination problem in trees.

Algorithm MILDTree

Determine the Multiple Intruder Locating Domination number of a tree.

Input. A tree $T$with a tree ordering$[v_1, v_2, \dots v_n]$.

Output. The MILD number $\gamma_{ml}(T)$ of T.
Method.

```
(Stage 1 ↓)
for $i = 1$ to $n$ do
    $\gamma^{1}_{ml}(v_i) \leftarrow 1$
    $\gamma^{0}_{ml}(v_i) \leftarrow \infty$
    $\gamma^{00}_{ml}(v_i) \leftarrow 0$
    $\gamma^{100}_{ml}(v_i) \leftarrow \infty$
    $\gamma^{101}_{ml}(v_i) \leftarrow \infty$
    $\gamma^{11}_{ml}(v_i) \leftarrow 1$
end for
(Stage 2 ↓)
for $i = 1$ to $n-1$ do
    let $v_j$ be the parent of $v_i$
    $\gamma_{ml}(v_i) \leftarrow \min\{\gamma^{1}_{ml}(v_i), \gamma^{0}_{ml}(v_i)\}$
    $\gamma^{1}_{ml}(v_j) \leftarrow \min\{\gamma^{11}_{ml}(v_j) + \min[\gamma^{00}_{ml}(v_i), \gamma_{ml}(v_i)],\ \gamma^{100}_{ml}(v_j) + \gamma^{1}_{ml}(v_i),\ \gamma^{101}_{ml}(v_j) + \gamma_{ml}(v_i)\}$
    $\gamma^{0}_{ml}(v_j) \leftarrow \min\{\gamma^{0}_{ml}(v_j) + \min[\gamma^{0}_{ml}(v_i), \gamma^{101}_{ml}(v_i), \gamma^{11}_{ml}(v_j)], \gamma^{00}_{ml}(v_j) + \gamma^{11}_{ml}(v_i)\}$
    $\gamma^{00}_{ml}(v_j) \leftarrow \gamma^{00}_{ml}(v_j) + \min[\gamma^{0}_{ml}(v_i), \gamma^{101}_{ml}(v_i)]$
    $\gamma^{100}_{ml}(v_j) \leftarrow \min\{\gamma^{100}_{ml}(v_j) + \gamma^{1}_{ml}(v_i), \gamma^{11}_{ml}(v_j) + \gamma^{00}_{ml}(v_i)\}$
    $\gamma^{101}_{ml}(v_j) \leftarrow \min\{\gamma^{101}_{ml}(v_j) + \gamma_{ml}(v_i), \gamma^{11}_{ml}(v_j) + \gamma^{0}_{ml}(v_i)\}$
    $\gamma^{11}_{ml}(v_j) \leftarrow \gamma^{11}_{ml}(v_j) + \gamma^{1}_{ml}(v_i)$
end for
(Stage 3 ↓)
  $\gamma_{ml}(T) \leftarrow \min\{\gamma^{1}_{ml}(v_n), \gamma^{0}_{ml}(v_n)\}$.
```

Complexity of Algorithm MILDTree

In Stage 1 of the algorithm, each of the statements in the body of the loop takes constant time. That collection of statements is executed n times by the for loop. Thus, the execution time of Stage 1 increases linearly with the input size, i.e., Stage 1 is executed in $O(n)$ time. Similar argument follows for stage two, and, Stage 3 is a single statement. Hence, the algorithm is executed in $O(n)$ time.

4. MILD sets in Unicyclic Graphs

To extend our Algorithm MILDTree to unicyclic graphs, we first discuss a labeling of vertices.

Labeling

Let U_n be a unicyclic graph with n vertices in its cycle. For our algorithm, we look at the unicyclic graph U_n as a set of n trees (each with vertices ≥ 1), one vertex of each coming together to form a cycle.

Suppose U_n has N vertices in total and $p(=N-n)$ vertices not belonging to the cycle. We label the p vertices of the trees one-by-one using pre-order (i.e., by the rule: label a parent vertex only after all its children are labelled) as $v_1, v_2, v_3, \dots, v_p$. Now, choose a vertex of the cycle, label it as v_{p+1}. Label its two parent vertices as v_{p+2} and v_{p+3}. Next label the parent of v_{p+2} as v_{p+4} and the parent of v_{p+3} as v_{p+5}, and so on. This ends with $v_{p+\text{odd number}}$ if the number of vertices in the cycle is odd, else it ends with $v_{p+\text{even number}}$.

Algorithm MILDUnicyclic

Determine the Multiple Intruder Locating Domination number of a unicyclic graph.

Input. A unicyclic graph U_n with N vertices, n of them in the cycle.

Output. The MILD number $\gamma_{ml}(U_n)$ of U_n.

Method

Step 1. Label the vertices of U_n by the labeling procedure as indicated.

Step 2. Perform *Stage* 1 of algorithm MILDTree by replacing the first line with

for $i = 1$ **to** N **do**

Step 3. Perform *Stage* 2 of algorithm MILDTree by replacing its first line by

for $i = 1$ **to** p **do**

(If $p = 0$, then the loop will not be entered, thus, *Stage* 2 will not be performed.)

Step 4. Now, the next vertex v_{p+1} has two parents, viz. v_{p+2} and v_{p+3}. For convenience in the further discussion, we shall name v_{p+1} as w, v_{p+2} as u, v_{p+3} as v as shown in Figure 6.

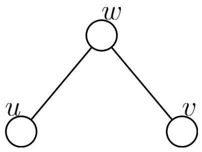

Figure 6

Step 5a. We get the following cases and, the corresponding changes to be done are indicated:

Case 5a.1. w can devout-dominate($\gamma_{ml}^{11}(w)$), u is unsecured and is ready to be secured by w ($\gamma_{ml}^{00}(u)$ for which v must be a codeword.

$\gamma_{ml}^{0}(u) \leftarrow \gamma_{ml}^{00}(u)$

$\gamma_{ml}^{00}(u) = \gamma_{ml}^{11}(u) = \gamma_{ml}^{100}(u) = \gamma_{ml}^{101}(u) = \infty$
$\gamma_{ml}^{0}(v) = \gamma_{ml}^{00}(v) = \infty$
other values: no change.
$\lambda = \gamma_{ml}^{11}(w)$
go to Step 5b

Case 5a.2. interchange u and v in Case 5a.1.
go to Step 5b

Case 5a.3. w can devout-dominate($\gamma_{ml}^{11}(w)$) but u and v are not affected by that fact.

All values for u and v remain same.
$\lambda = \gamma_{ml}^{11}(w)$
go to Step 5b

Case 5a.4. w is unsecured and is ready to be secured by u ($\gamma_{ml}^{00}(w), \gamma_{ml}^{11}(u)$) and hence v cannot be a devout dominating vertex.

$\gamma_{ml}^{100}(u) \leftarrow \gamma_{ml}^{11}(u)$
$\gamma_{ml}^{0}(u) = \gamma_{ml}^{11}(u) = \gamma_{ml}^{101}(u) = \gamma_{ml}^{00}(u) = \infty$
$\gamma_{ml}^{100}(v) = \infty$, $\gamma_{ml}^{101}(v) \leftarrow \min\{\gamma_{ml}^{101}(v), \gamma_{ml}^{11}(v)\}$
other values: no change.
$\lambda = \gamma_{ml}^{00}(w)$
go to Step 5b

Case 5a.5. interchange u and v in case (iv).
go to Step 5b

Case 5a.6. w is a secure non-codeword ($\gamma_{ml}^{0}(w)$) and hence u and v cannot be devout dominating vertices.

$\gamma_{ml}^{101}(u) \leftarrow \min\{\gamma_{ml}^{101}(u), \gamma_{ml}^{11}(u)\}$
$\gamma_{ml}^{101}(v) \leftarrow \min\{\gamma_{ml}^{101}(v), \gamma_{ml}^{11}(v)\}$
$\gamma_{ml}^{11}(u) = \gamma_{ml}^{100}(u) = \gamma_{ml}^{11}(v) = \gamma_{ml}^{100}(v) = \infty$,
other values: no change.
$\lambda = \gamma_{ml}^{0}(w)$
go to Step 5b

Case 5a.7. w is a devout dominator ($\gamma_{ml}^{100}(w)$) and hence u and v must be codewords.

$\gamma_{ml}^{0}(u) = \gamma_{ml}^{00}(u) = \gamma_{ml}^{0}(v) = \gamma_{ml}^{00}(v) = \infty$
other values: no change
$\lambda = \gamma_{ml}^{100}(w)$
go to Step 5b

Case 5a.8. w is a codeword but not a devout dominator ($\gamma_{ml}^{101}(w)$) and hence u and v remain unaffected.

```
    All values remain same for both $u$ and $v$.
    $\lambda = \gamma_{ml}^{101}(w)$
    go to Step 5b
```

Step 5b

```
do
{
    (i): Perform Stage 2 of algorithm MILDTree by replacing its first line by
            for $i = p+2$ to $N-1$ do
    (ii): $\gamma_{ml}(U_n)(k) \leftarrow \min\{\gamma^1_{ml}(v_N), \gamma^0_{ml}(v_N)\} + \lambda$
}
end do
    $k \leftarrow k+1$
if ($k \le 8$)
    go to case $k$ in Step 5a
else
    go to step 6
```

Step 6

```
    $\gamma_{ml}(U_n) \leftarrow \min\{\gamma_{ml}(U_n)(1), \gamma_{ml}(U_n)(2), \dots, \gamma_{ml}(U_n)(7), \gamma_{ml}(U_n)(8)\}$
```

Complexity of Algorithm MILDUnicyclic

As in the discussion of the complexity of MILDTree algorithm, Steps 1 and 2 of Algorithm MILDUnicyclic run in linear time. Steps 3 and 5b are together executed in $O(n)$ time whereas the Steps 4, 5 and 6 take constant time. It can be noted that none of the loops are nested. Hence, this algorithm is executed in $O(n)$ time too.

5. Conclusion

The MILD problem is proved to be NP-Complete for arbitrary graphs. However, when restricted to trees, a linear time algorithm can be formed, and that is done by the method of dynamic programming. The algorithm is then modified to a case where there would be a single cycle in a given graph, and thus, a linear time algorithm to find the MILD number of unicyclic graphs is obtained.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] G.J. Chang, Algorithmic aspects of domination in graphs, in *Handbook of Combinatorial Optimization*, 1811 – 1877,Springer (1998), DOI: 10.1007/978-1-4613-0303-9_28.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., USA (1979), `https://dl.acm.org/doi/book/10.5555/578533`.

[3] A. González, C. Hernando and M. Mora, Metric-locating-dominating sets of graphs for constructing related subsets of vertices, *Applied Mathematics and Computation* **332** (2018), 449 – 456, DOI: 10.1016/j.amc.2018.03.053.

[4] J. Guo, Z. Li and M. Lu, Locating-Total Domination in Grid Graphs, *Bulletin of Malaysian Mathematical Science Society* **43** (2020), 1195 – 1204, DOI: 10.1007/s40840-019-00733-9.

[5] M.G. Karpovsky, K. Chakrabarty and L. B. Levitin, On a new class of codes for identifying vertices in graphs, *IEEE Transactions on Information Theory* **44**(2) (1998), 599 – 61, DOI: 10.1109/18.661507.

[6] G. Rajasekar and K. Nagarajan, Algorithm for finding location domination number of Corona product of graphs, *Malaya Journal of Matematik* **S**(1), 130 – 133, DOI: 10.26637/MJM0S01/0031.

[7] S. J. Seo and P. J. Slater, Open neighborhood locating-dominating sets, *Australasian Journal of Combinatorics* **46** (2010), 109 – 120, `https://ajc.maths.uq.edu.au/pdf/46/ajc_v46_p109.pdf`.

[8] A. Simic, M. Bogdanovic and J. Milosevic, The binary locating-dominating number of some convex polytopes, *Ars Mathematica Contemporanea* **13** (2017), 367 – 377, DOI: 10.26493/1855-3974.973.479.

[9] P. J. Slater, Fault-tolerant locating-dominating sets, *Discrete Mathematics* **249**(1-3) (2002), 179 – 189, DOI: 10.1016/S0012-365X(01)00244-8.

[10] P. J. Slater, Locating dominating sets and locating-dominating sets, in *Graph Theory, Combinatorics and Applications: Proceedings of the Seventh Quadrennial International Conference on the Theory and Applications of Graphs*, **2**, 1073 – 1079 (1995), `https://scholar.google.com/scholar?q=Locating%20dominating%20sets%20and%20locatingdominating%20sets`.

[11] K. Venugopal and K. A. Vidya, Multiple intruder locating dominating sets, *International Journal of Scientific Research in Mathematical and Statistical Sciences* **6**(2) (2019), 394 – 398, `https://www.isroset.org/journal/IJSRMSS/full_paper_view.php?paper_id=1315`.