Independent locating-dominating sets in P_4-sparse graphs

Dayllon Vinícius Xavier Lemos, Márcia Rodrigues Cappelle,
Erika Morais Martins Coelho, Leslie Richard Foulds, Humberto José Longo

[1]Instituto de Informática – Universidade Federal de Goiás
Alameda Palmeiras, qd. D, campus Samambaia, 74001-970, Goiânia, Goiás, Brasil.

dayllonxavier@discente.ufg.br, {mcappelle,erika_coelho,lesfoulds,longo}@ufg.br

Abstract. *A vertex subset S of a graph of order n is termed* independent locating-dominating *(an ILD set for short) if it is independent, dominating and no two distinct vertices of $V(G) \setminus S$ have the same open neighbourhood in S, i.e. $N_G(u) \cap S \neq N_G(v) \cap S, \forall\, u, v \in V(G) \setminus S$, $u \neq v$. We study ILD sets in P_4-*sparse *graphs and propose two algorithms: one for recognizing whether or not a graph admits an ILD set and another for computing an ILD set of minimum cardinality, both with time complexity $O(n^2)$.*

1. Introduction

Consider a graph $G = (V(G), E(G))$ of order n. A subset $S \subseteq V(G)$ is called *dominating* if every vertex in $V(G) \setminus S$ is adjacent to at least one vertex in S. If, in addition, no two distinct vertices in $V(G) \setminus S$ share the same open neighbourhood, then S is said to be *locating-dominating* [Slater 1987, Foucaud et al. 2017, Jean 2023]. S is *independent* if no two of its vertices are adjacent. When S is both independent and locating-dominating, it is referred to as *independent locating-dominating* (ILD) set [Slater and Sewell 2018]. The minimum size of such a set S is $ILD(G)$. Such sets are relevant in applications like *intruder identification* in networks, where sensors are placed on vertices to both detect and uniquely identify unauthorized activity [Ray et al. 2004]. Thus, small *ILD* sets reduce the number of sensors required while maintaining full identification capability, leading to more efficient sensor placement and enhanced monitoring in distributed environments.

A graph G is called P_4-sparse if every subset of five vertices in G induces at most one P_4. This class is a superclass of cographs (graphs which do not contain an induced P_4). Jamison and Olariu proposed a decomposition tree and linear-time recognition and optimization algorithms for P_4-sparse graphs [Jamison and Olariu 1992a, Jamison and Olariu 1992b, Jamison and Olariu 1995]. Other researchers, such as [Bravo et al. 2011], introduced new categorizations for the P_4-sparse class.

Not all graphs have an *ILD* set, such as the graphs C_3 and C_4. The problem of determining whether or not a given arbitrary graph contains an *ILD* set is $\mathcal{NP}$-complete [Slater and Sewell 2018]. We study *ILD* sets in P_4-sparse graphs and propose two algorithms: one to recognize whether a P_4-sparse admits an *ILD* set and another to compute an *ILD* set of minimum cardinality in such a graph, both with time complexity $O(n^2)$.

2. Some basic definitions

Throughout this paper, it is assumed that $G = (V(G), E(G))$ is a simple, undirected, finite, nontrivial graph of order $|V(G)| = n$. The set $N(v) = \{u \mid (u, v) \in E(G)\}$ is the *neighborhood* of a vertex $v \in V(G)$ and the set $N[v] = N(v) \cup \{v\}$ is its *closed* *neighborhood*. A pair of vertices $u, v \in V(G)$ are called *true twins* if $N[u] = N[v]$. Let $G[A]$ denote the subgraph of G induced by the vertex subset $A \subseteq V(G)$. We denote as P_n, C_n, and K_n the path, cycle, and complete graph, respectively, on n vertices.

The problem of determining whether a graph G contains an ILD set can be formally stated as: $\mathcal{ILD}$: *Given a graph G, does G admit an ILD set?* Similarly, the problem of computing an ILD set of minimum cardinality for G can be defined as: $\mathcal{MINILD}$: *Given a graph G that admits an* ILD *set, find an ILD set S of G of minimum cardinality.*

A graph G is called a *spider* if $V(G)$ can be partitioned into three subsets (R, K, S), where S is an independent set, K is a clique, $|K| = |S| \geqslant 2$, and the remaining set R (possibly empty) is termed the *head* of the spider. Moreover, $(r, k) \in E(G)$ and $(r, s) \notin E(G)$, where $r \in R$, $s \in S$, and $k \in K$. There exists a bijective function $f : S \to K$ such that

$$N(s) \cap K = \{f(s)\}, \forall\, s \in S, \tag{1}$$

or

$$N(s) \cap K = K - \{f(s)\}, \forall\, s \in S. \tag{2}$$

When (1) is satisfied, G is a *thin spider* (an example is depicted in Figure 1(a)), and when (2) is satisfied, it is a *thick spider* (as shown in Figure 1(b)).

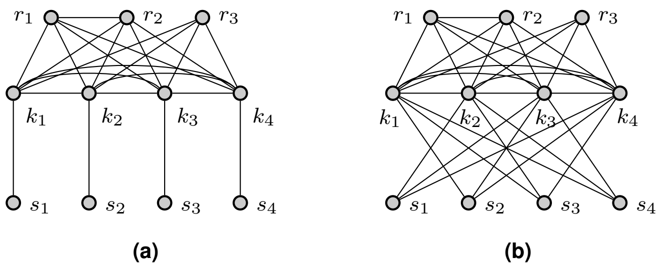

Figure 1. (a) thin spider and (b) thick spider.

Let G_1 and G_2 be two graphs. The binary operations $\oplus$ and $\otimes$ on such graphs are defined as: $G_1 \oplus G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$ and $G_1 \otimes G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{(u, v) \mid u \in V(G_1) \text{ and } v \in V(G_2)\})$. Furthermore, assume that $E(G_1) = \emptyset$ and $V(G_2) = \{v\} \cup K \cup R$, where the following conditions hold:

- $|K| = |V(G_1)| + 1 \geqslant 2$;
- K is a clique;
- $(r, k) \in E(G_2)$ and $(r, v) \notin E(G_2)$, where $r \in R$ and $k \in K$;
- There exists a vertex $v' \in K$ such that $N_{G_2}(v) = \{v'\}$ or $N_{G_2}(v) = K - \{v'\}$.

Thus, the additional binary operation $\circledcirc$ on these graphs results in a spider graph: $G_1 \circledcirc G_2 = (V(G_1) \cup V(G_2), E(G_2) \cup E')$, where

$$E' = \begin{cases} \{(u, f(u)) \mid u \in V(G_1)\}, & \text{whenever } N_{G_2}(v) = \{v'\}; \\ \{(u, z) \mid u \in V(G_1), z \in K - \{f(u)\}\}, & \text{whenever } N_{G_2}(v) = K - \{v'\}. \end{cases}$$

The above notions can be combined to establish conditions for P_4-sparse graphs to exist, as started in the following theorem.

Theorem 2.1 ([Jamison and Olariu 1992b]). *A graph G is P_4-sparse if and only if G is obtained from graphs K_1 by a finite sequence of operations $\oplus$, $\otimes$ and $\circledcirc$.*

3. Independent locating-dominating sets in P_4-sparse graphs

Now we present results that establish the existence and minimality conditions for *ILD* sets in graphs resulting from each of the operations $\oplus$, $\otimes$, and $\circledS$. These findings are essential for the development of the algorithms presented in Section 4.

Theorem 3.1. *$G_1 \oplus G_2$ admits an* ILD *set iff both G_1 and G_2 also admit* ILD *sets.*
Theorem 3.2. *If $G_1 \otimes G_2$ admits an* ILD *set S, then S is an* ILD *set of either G_1 or G_2. Furthermore, either $G_1 = K_1$ or $G_2 = K_1$.*

Proof. In $G_1 \otimes G_2$, every vertex of G_1 is adjacent to every vertex of G_2. Thus, by the independence condition, S cannot contain vertices from both G_1 and G_2 simultaneously. This implies that either $S \subseteq V(G_1)$ or $S \subseteq V(G_2)$. Since S is an *ILD* set of $G_1 \otimes G_2$, it must also be an *ILD* set of either G_1 or G_2. Now, consider, w.l.o.g., $S \subseteq V(G_1)$. Suppose, for contradiction, that $|V(G_1)|, |V(G_2)| \geqslant 2$. This implies that all vertices of G_2 in $G_1 \otimes G_2$ are located by the same neighborhood in S, violating the locating property. So either $G_1 = K_1$ or $G_2 = K_1$. □

Theorem 3.3. *If G is a P_4-sparse graph that admits an* ILD *set, then G is not the result of two consecutive $\otimes$ operations, i.e., $G \neq (G' \otimes a) \otimes b$ for some P_4-sparse graph G'.*

Proof. The vertices a and b are true twins in $(G' \otimes \{a\}) \otimes \{b\}$. As a consequence, at least one of a or b, must belong to the *ILD* set S of G. Assume, w.l.o.g., $a \in S$. So, neither b nor any other vertex from G' can be in S, since all are adjacent to a. This implies that $N(b) \cap S = N(u) \cap S$ for some $u \in V(G')$, contradicting the locating property. □

Theorem 3.4. *If G is a spider (R, K, S) that admits an* ILD *set, then $ILD(G) = |S| + ILD(G[R])$.*

Proof. We will prove this theorem by considering the following two cases.

Case (i). **G is a thin spider with R $= \emptyset$:** We have that $ILD(G) \geqslant |S|$, since each pendant vertex must either be dominated by some vertex or be a member of $ILD(G)$. Hence, $|S|$ is an *ILD* set of minimum cardinality of G. Since $ILD(\emptyset) = 0$, the theorem holds.

Case (ii). **G is a thick spider or R $\neq \emptyset$:** In this case, there is no *ILD* set that contains any vertex of K. If a vertex $k_i \in K$ belongs to the *ILD* set, then the vertices in A would be located by the same neighborhood $\{k_i\}$, where $A = R \cup \{s_i\}$ if G is a thin spider, and $A = R \cup S - \{s_i\}$ if G is a thick spider. So, due to the independence property, no other vertex adjacent to any vertex in A is a member of the *ILD* set. Therefore, to satisfy the domination property, all vertices in S must be in the *ILD* set of G. Moreover, since there is no edge between S and R, we obtain $ILD(G) = |S| + ILD(G[R])$. □

Theorem 3.5. *Let G, G_1, G_2 be P_4-sparse graphs, and let S, S_1, S_2 be their respective minimum* ILD *sets. If G is a P_4-sparse graph that admits an* ILD *set, then:*

- *If $G = G_1 \oplus G_2$, then both G_1 and G_2 must admit an* ILD *set, and $S = S_1 \cup S_2$.*
- *If $G = G_1 \otimes G_2$, then $G_1 = K_1$, and G_2 is either K_1 or a structure obtained through an operation other than $\otimes$ (ensuring that G does not contain true twins). In addition, $S = S_2$.*
- *If G is a spider graph (R, K, S'), then $G[R]$ must admit an* ILD *set, and $S = S' \cup S''$, where S'' is the minimum* ILD *set of $G[R]$.*

Proof. The proof follows directly from Theorems 2.1, 3.1, 3.2, 3.3 and 3.4. □

4. Algorithms

Based on Theorem 3.5, we also propose recursive Algorithms 1 and 2 below to solve the $\mathcal{ILD}$ and $\mathcal{MINILD}$ problems, respectively. Both algorithms involve each possible case for a P_4-sparse graph: a trivial graph K_1, a spider resulting from an application of the $\circledvee$ operation, a connected graph resulting from the $\otimes$ operation, and a disjoint graph resulting from the $\oplus$ operation. Each case can be solved in linear time, and since each recursive call reverses at least one operation from the construction of the graph, the total time complexity in each case is $\mathcal{O}(n^2)$, where n is the order of the graph.

Algorithm 1: hasILDSet(G)

Input: P_4-sparse graph G.
Output: Boolean – G admits an *ILD* set?

```
1  if G = K_1 then return True ;
2  else if G is a spider (R, K, S) then
3      return hasILDSet(G[R]);
4  else if G is connected then
5      A ← {v | v ∈ V(G), |N(v)| = n − 1};
6      if |A| = 1 then
7          return hasILDSet(G[V(G) − A]);
8      else
9          return False;
10 else
11     for each connected component G' of G do
12         if hasILDSet(G[V(G')]) = False then
13             return False;
14     return True;
```

Algorithm 2: minILDSet(G)

Input: P_4-sparse graph G that admits *ILD* sets.
Output: *ILD* set of G of minimum cardinality.

```
1  set ← ∅;
2  if G = K_1 then set ← V(G) ;
3  else if G is a spider (R, K, S) then
4      set ← S ∪ minILDSet(G[R]);
5  else if G is connected then
6      A ← {v | v ∈ V(G), |N(v)| = n − 1};
7      set ← minILDSet(G[V(G) − A]);
8  else
9      foreach connected component G' of G do
10         set ← set ∪ minILDSet(G[V(G')]);
11 return set;
```

In Algorithm 1, line 1 deals with the base case where $G = K_1$. The results of the $\circledvee$, $\otimes$, and $\oplus$ operations are verified in lines 2, 4, and 10, respectively. If the execution reaches line 9, the algorithm concludes with the fact that no *ILD* set exists for G, meaning that the second condition of Theorem 3.5 is not satisfied.

Similarly, in Algorithm 2, lines 2, 3, 5, and 8 correspond, respectively, to the base case where $G = K_1$ and the results of the $\circledvee$, $\otimes$, and $\oplus$ operations. If G consists of a single vertex, it is directly added to the *ILD* set (line 2). When G is a spider, its independent set S is included in the *ILD* set of G, and the recursion proceeds with the head R (line 4), as stated in Theorem 3.4. If G results from a $\otimes$ operation, line 7 removes the K_1 vertex that was one of the operands and recursively calls the algorithm for the remaining vertices, following Theorem 3.2. Finally, if G arises from one or more consecutive $\oplus$ operations, it is disconnected. In this case, as is stated in Theorem 3.1, the recursion runs separately on each connected component of G, and the union of the resulting *ILD* sets forms the *ILD* set of G (line 10). This is the only step in the algorithm capable of reversing multiple operations within the same recursion.

Acknowledgments: To the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Funding Code 001.

References

Bravo, R. S., Klein, S., Nogueira, L. T., and Protti, F. (2011). Characterization and recognition of P_4-sparse graphs partitionable into k independent sets and ℓ cliques. *Discrete Applied Mathematics*, 159(4):165–173.

Foucaud, F., Mertzios, G. B., Naserasr, R., Parreau, A., and Valicov, P. (2017). Identification, location–domination and metric dimension on interval and permutation graphs. i. bounds. *Theoretical Computer Science*, 668:43–58.

Jamison, B. and Olariu, S. (1992a). Recognizing P_4-sparse graphs in linear time. *SIAM Journal on Computing*, 21(2):381–406.

Jamison, B. and Olariu, S. (1992b). A tree representation for P_4-sparse graphs. *Discrete Applied Mathematics*, 35(2):115–129.

Jamison, B. and Olariu, S. (1995). Linear time optimization algorithms for P_4-sparse graphs. *Discrete Applied Mathematics*, 61(2):155–175.

Jean, D. (2023). Watching systems, identifying, locating-dominating and discriminating codes in graphs. `https://dragazo.github.io/bibdom/main.pdf`. Last accessed on: February 05, 2025.

Ray, S., Starobinski, D., Trachtenberg, A., and Ungrangsi, R. (2004). Robust location detection with sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1016–1025.

Slater, P. J. (1987). Domination and location in acyclic graphs. *Networks*, 17(1):55–64.

Slater, P. J. and Sewell, J. L. (2018). Independent locating-dominating sets and independent identifying codes in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 104:261–272.