An algorithm for minimum identifying codes in some Cartesian products of graphs

Rafael de A. Cruz
Universidade Federal de Goiás
Instituto de Informática (INF) - UFG
rassisc@gmail.com

Márcia R. Cappelle
Universidade Federal de Goiás
Instituto de Informática (INF) - UFG
marcia@inf.ufg.br

ABSTRACT

An identifying code in a graph is a dominating set that also has the property that the closed neighborhood of each vertex in the graph has a distinct intersection with the set. Hedetniemi (On identifying codes in the Cartesian product of a path and a complete graph, J. J Comb Optim 31 (2016) 1405-1416) show how to construct minimum identifying codes for Cartesian products of complete graph of order $n = 3$ and $n \geq 5$ with a path graph with order $m \geq 3$. We present a dynamic programming algorithm to determine the size of an identifying code of minimum order in these graphs. For the case $n = 4$, which were not considered by the author, the algorithm has running time $\mathcal{O}(m)$.

KEYWORDS. Identifying Codes. Cartesian Product. Algorithms.

Main area: Theory and Algorithms in Graphs (TAG)

1. Introduction

We consider finite, simple, and undirected graphs, and use standard notation and terminology.

Given a graph $G = (V, E)$, let us denote by $N_G[u]$ the closed neighbourhood of $u \in V(G)$, that is, the set of vertices adjacent to u including u. For a positive integer d, let $N_G^{\leq d}[u]$ be the set of vertices of G at distance at most d from u. Note that $N_G[u]$ in G coincides with $N_G^{\leq 1}[u]$.

A set C of vertices of a graph G is a d*-identifying code* in G for a positive integer d if the sets $N_G^{\leq d}[u] \cap C$ are non-empty and distinct for all vertices u of G. A 1-identifying code is known simply as an *identifying code*. If G is clear from the context, we just write $N[u]$ instead of $N_G[u]$. Let $\gamma^{ID}(G)$ denote the minimum order of an identifying code in C and γ^{ID}-set denote such a set (See Figure 1).

Identifying codes were first introduced in 1998 [Karpovsky et al., 1998] to model a fault-detection problem in multiprocessor systems. They have found numerous applications. For instance, the concept was applied to model location detection with sensor networks [Berger-Wolf et al., 2005; Ray et al., 2003, 2004].

It is algorithmically hard [Charon et al., 2003] to determine identifying codes of minimum order even for planar graphs of arbitrarily large girth [Auger, 2010]. Some families of restricted graphs have been studied, including paths and cycles [Gravier et al., 2006; Junnila and Laihonen, 2012b] and trees [Auger, 2010; Blidia et al., 2007; Charon et al., 2006]. With respect to graph products, it was determined the $\gamma^{ID}(G)$ when G is a Cartesian product of two cliques [Goddard and Wash, 2013], and given upper and lower bounds of γ^{ID} for Cartesian products of a graph G and K_2 [Rall and Wash, 2016]. Results for grids can be found in [Ben-Haim and Litsyn, 2005; Cohen et al., 1999; Daniel et al., 2004; Junnila and Laihonen, 2012a; Martin and Stanton, 2010]. Other recent results on products consider the lexicographic product [Feng et al., 2012], the direct product [Rall and Wash, 2014], the corona product [Feng and Wang, 2014] and the complementary prism [Cappelle et al., 2015]. There is a large bibliography on identifying codes, which can be found on Antoine Lobstein's webpage [Lobstein, 2016].

In the present paper we study the Cartesian product of a complete graph and a path. Minimum identifying codes for these graphs were constructed in [Hedetniemi, 2016] where complete graphs of order four were not considered. We present a dynamic programming algorithm that determines the size of a minimum identifying code in these products. For the complete graphs of order four the algorithm is linear on the size of the path.

The organization of the paper is as follows. In Section 2 we first introduce some definitions and preliminary results. In Section 3 we present a dynamic programming algorithm to determine the minimum order of an identifying code in $K_n \square P_m$ which is linear for $n = 4$. Finally, we conclude and present some open problems in Section 4.

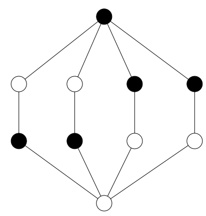
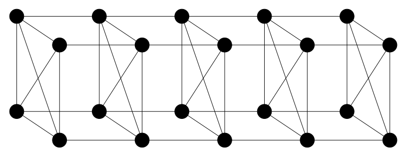


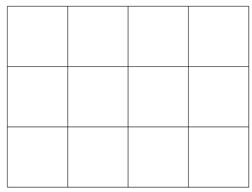
Figure 1: Graph G such that $\gamma^{ID}(G) = 5$ and γ^{ID}-set of G are the black vertices.

2. Definitions and preliminary results

The Cartesian product of two graphs G and H, denoted by $G \square H$, is the graph with vertex set $V(G \square H) = V(G) \times V(H)$ and edge set $E(G \square H)$ satisfying the following condition: $(u, u')(v, v') \in E(G \square H)$ if and only if



(a)



(b)

Figure 2: Graph $K_4 \square P_5$. (a) is its usual representation and (b) the grid representation used in this paper where each vertex is represented by crossing lines. Vertical lines represent K_n-layers and the horizontal ones P_m-layers.

- either $u = v$ and $\{u', v'\} \in E(H)$ or
- $u' = v'$ and $\{u, v\} \in E(G)$.

We consider minimum identifying codes in $K_n \square P_m$ where K_n denotes the complete graph on n vertices and P_m denotes the path on m vertices. In Figure 2 we can see two distinct representations of $K_4 \square P_5$. We assume that $n \geq 3$ and that $m \geq 3$. We define $V(K_n) = \{1, 2, \ldots, n\}$ and $V(P_m) = \{1, 2, \ldots, m\}$. For convenience, we refer to the subgraph of $K_n \square P_m$ induced by $V(K_n) \square \{y\}$ (or $V(P_m) \square \{x\}$) as the K_n-layer (or P_m-layer) through y (or x). We denote $V(K_n) \square \{y\}$ by K^y . Two K_n-layers K^i and K^j are adjacent if $\{i, j\} \in E(P_m)$, and are non-adjacent otherwise. For positive integers x, y, let $[x]$ denote the set of integers at most x and $[x, y]$ the set of integers at least x and at most y. We denote $K^{[x,y]} = \bigcup_{x \leq \ell \leq y} K^{\ell}$.

Not all graphs admit an d-identifying code. A necessary and sufficient condition to admit a d-identifying code is that for any pair of distinct vertices u and v we have $N^{\leq d}[u] \neq N^{\leq d}[v]$. Since for $n, m \geq 2$ and for every $u, v \in K_n \square P_m$, $N[u] \neq N[v]$, these graphs always admit an identifying code.

Hedetniemi [Hedetniemi, 2016] determines the minimum cardinality of an identifying code in $K_n \square P_m$ for all $m \geq 3$ when $n = 3$ and when $n \geq 5$. He does not consider the case $n = 4$.

Theorem 1. *[Hedetniemi, 2016] For $m \geq 4$, $\gamma^{ID}(K_3 \square P_m) = m + 2 + \lfloor \frac{m-4}{4} \rfloor$ and for $n \geq 5$,*

$$\gamma^{ID}(K_n \square P_m) = \begin{cases} 2k(n-1) + 3 & \text{if } m = 4k, \\ (2k+1)(n-1) + 1 & \text{if } m = 4k+1, \\ (2k+2)(n-1) & \text{if } m = 4k+2, \\ (k+1) \cdot 2(n-1), & \text{if } m = 4k+3 \text{ and } n \geq k+3, \\ (k+1) \cdot 2(n-1) + 1, & \text{if } m = 4k+3 \text{ and } n < k+3. \end{cases}$$

This kind of graph has a particular property, in view of the product construction. To verify if $C \subseteq V(K_n \square P_m)$ is an identifying code, we only need to check when vertices of every two adjacent K_n-layers are dominated and pairwise separated by C, as we prove in the next proposition. It will be useful to prove the correctness of our algorithms in Section 3.

Proposition 2. *Let $C \subseteq V(K_n \square P_m)$ such that for all $i \in [m-1]$, for every $v \in (K^i \cup K^{i+1})$, $N[v] \cap C$ are nonempty and pairwise distinct, then C is an identifying code of $K_n \square P_m$.*

Proof. Note that C is a dominating set of $V(K_n \square P_m)$. Let $u, v \in (K^i \cup K^j)$, for some $i, j \in [m]$. If $|j - i| \leq 1$, by hypothesis, they are separated by C. If $|j - i| \geq 3$, for every $u \in K^i$ and $v \in K^j$, $N[u] \cap N[v] = \emptyset$. Since C is a dominating set, these vertices are separated. Now, we may assume $|j - i| = 2$. Without loss of generality, assume $j = i + 2$. Suppose $u \in K^i$ and $v \in K^j$ such that u and v are not separated by C and $N[u] \cap N[v] \neq \emptyset$. So, by graph construction, u and v are in the same P_m-layer. Since C is dominating, $K^{i+1} \subseteq C$ and we can conclude $(K^i \cup K^j) \cap C = \emptyset$.

But this contradicts the fact that the vertices of K^{i+1} are pairwise separated by C. Hence C is an identifying code of $K_n \square P_m$. □

3. An algorithm for minimum identifying codes in $K_n \square P_m$

In this section we consider $m \geq 3$ and prove the following result:

Theorem 3. *There exists an algorithm which computes the minimum size of an identifying code in a graph $K_n \square P_m$ which is linear for a fixed n.*

For the algorithm we use the following notion: if G is a graph and $A \subseteq V(G)$, we say that a subset C of $V(G)$ is an A-almost identifying code of G if the sets $C \cap N[v]$ are all nonempty and pairwise distinct for all $v \in V(G) \setminus A$. With this definition, an $\emptyset$-almost identifying code is just an identifying code.

We use the dynamic programming method to determine the size of a minimum identifying code of $K_n \square P_m$. First we show that the optimal substructure of this problem is as follows. Suppose that for a minimum identifying code C of $K_n \square P_m$ we know the vertices of $C \cap K^{[m-2,m]} = S$. Hence, $C \cap K^{[1,m-3]}$ must have minimum cardinality possible among all solutions. If there were a set C' such that $C' \cup S$ is an identifying code of $K_n \square P_m$ and $|C'| < |C \setminus S|$, then we could substitute C to $C' \cup S$ in the optimal solution to produce another set with size lower than the optimum: a contradiction. Thus, an optimal solution of the given problem can be obtained by using optimal solutions of its subproblems.

To obtain a solution, we will first consider the problem to recursively find the size of C, an K^m-almost identifying code of $K_n \square P_m$, for a fixed $S \subseteq K^{[m-2,m]}$, that has minimum possible size for $C \cap K^{[1,m-3]}$. That is, the sets $C \cap N[v]$ are all nonempty and pairwise distinct for all $v \in V(K_n \square P_m) \setminus K^m$. This problem can be defined recursively as follows.

Let $C(r, S)$ be an $K^{[r,m]}$-almost identifying code of $K_n \square P_m$ such that $S = C(r, S) \cap K^{[m-2,m]}$. Let $S_\ell = S \cap K^m$. For $m = 3$, $C(3, S)$ is a subset of $K^{[1,3]}$ that dominates and pairwise separates every vertex in $K^{[1,2]}$. For $m \geq 4$,

$$|C(m, S)| = min\{|C(m-1, S'_i)|\} + |S_\ell|,$$

for all $S'_i \subseteq K^{[m-3,m-1]}$, $1 \leq i \leq 2^n$, with $(S'_i \cap K^{[m-2,m-1]}) = (S \cap K^{[m-2,m-1]})$ such that $C(m-1, S'_i) \cup S_\ell$ dominates and pairwise separates every vertex in $K^{[m-2,m-1]}$.

We use an auxiliary four-dimensional table

$$d[1..m-2, 1..2^n, 1..2^n, 1..2^n]$$

for storing all combination sets of three consecutive K_m-layers ($d[].code$) and the cardinality of a possible optimal solution considering the actual K_n-layer ($d[].size$). Thus, the algorithm should fill in the table d in a manner that corresponds to solving the problem of increasing length. Algorithm 1: ALMOST-ID computes the size of at most 2^{3n} K^m-almost identifying codes of $K_n \square P_m$ and returns table d. Algorithm 2: MINIMUM-ID receives as input table d and verifies among all possible solutions which are also identifying codes of $K_n \square P_m$, and chooses one with minimum cardinality.

We omit initializations in the Algorithm 1. Consider that, for all $1 \leq c \leq m-2$ and $1 \leq i, j, k \leq 2^n$, $d[c, i, j, k].size$ is initialized with the value $+\infty$ and $d[c, i, j, k].code$ with $\emptyset$. Algorithm 1 first computes (lines 2 to 8), for $1 \leq i, j, k \leq 2^n$, all sets $S_{ijk} \subseteq K^{[1,3]}$ that dominate and pairwise separate all vertices in $K^{[1,2]}$. That is, all possible $K^{[3,m]}$-almost identifying codes for $K_n \square P_m$. Each set S_{ijk} will be stored in $d[1, i, j, k].code$ and its size in $d[1, i, j, k].size$.

For $2 \leq c \leq m-2$, Algorithm 1 analyzes four K_n-layers at each step to do

$$d[c, j, k, \ell].size = \min\{d[c-1, i, j, k].size\} + |S_\ell|,$$

for all $1 \leq i \leq 2^n$, fixing vertices of $K^{[c,c+1]}$, where S_ℓ is a specific subset of $K^{[c+2]}$ and $(S_\ell \cup d[c-1,i,j,k].code)$ dominates and pairwise separates all vertices in $K^{[c,c+1]}$.

To obtain $\gamma^{ID}(K_n \,\square\, P_m)$, Algorithm 2 needs to determine $\min\{d[m-2,i,j,k].size\}$ for $1 \leq i,j,k \leq 2^n$ such that S_{ijk}, a subset of $K^{[m-2,m]}$, dominates and pairwise separates all vertices of $K^{[m-1,m]}$, since vertices of $K^{[1,m-1]}$ were already checked by Algorithm 1.

Algorithm 1: ALMOST-ID

Input: $n \geq 3$, $m \geq 3$.

Output: minimum cardinality of a identifying code of $K_n \,\square\, P_m$

1 **begin**

`/* first step */`

2   **foreach** $S_i \subseteq K^1$, $S_j \subseteq K^2$, $S_k \subseteq K^3$, $1 \leq i,j,k \leq 2^n$ **do**

3     $S \leftarrow S_i \cup S_j \cup S_k$;

4   **if** *for every* $v \in (K^{[1,2]})$, $(N[v] \cap S')$ *are nonempty and pairwise distinct* **then**

5       $d[1,i,j,k].code \leftarrow S$;

6       $d[1,i,j,k].size \leftarrow |S|$;

7   **end**

8   **end**

`/* computing table d */`

9   **for** $2 \leq c \leq m-2$ **do**

10   **foreach** *combination* ijk *with* $1 \leq i,j,k \leq 2^n$ **do**

11       $S \leftarrow d[c-1,i,j,k].code$;

12       $s \leftarrow d[c-1,i,j,k].size$;

13   **foreach** $S_\ell \subseteq K^{c+2}$, $1 \leq \ell \leq 2^n$ **do**

14   **if** *for every* $v \in (K^{[c,c+1]})$, $(N[v] \cap (S \cup S_\ell))$ *are nonempty and pairwise distinct* **then**

15   **if** $s + |S_\ell| < d[c,j,k,\ell].size$ **then**

16             $d[c,j,k,\ell].code \leftarrow (S \setminus K^{c-1}) \cup S_\ell$;

17             $d[c,i,j,k].size \leftarrow s + |S_\ell|$;

18   **end**

19   **end**

20   **end**

21   **end**

22   **end**

23 **end**

Lemma 4. *Algorithm 1, for* $m \geq 3$, $1 \leq c \leq m-2$, *and a subset* $S_{jk\ell}$ *of* $K^{[c,c+2]}$, *stores*

(i) in the entry $d[c,j,k,\ell].size$*, the cardinality of a* $K^{[c+2,m]}$*-almost identifying code* $C_{jk\ell}$ *of* $K_n \,\square\, P_m$ *such that* $C_{jk\ell} \cap K^{[c,c+2]} = S_{jk\ell}$*, and* $|C_{jk\ell} \cap K^{[1,c-1]}|$ *has the minimum value possible, if it exists.*

(ii) in the entry $d[c,j,k,\ell].code$ *the set* $S_{jk\ell}$*, if conditions above are satisfied.*

Proof. We prove by induction on c. For $c = 1$, loop of line 2 stores in $d[c].code$ all subsets of $K^{[1,3]}$ that dominate and pairwise separate all vertices of $K^{[1,2]}$ with their respective sizes in $d[c].size$. Since $K^{[1,0]}$ is empty the statement is trivially true. Assume that the above statements are true up to $c-1 > 0$. Let C_{ijk} be an $K^{[c+1,m]}$-almost identifying code of $K_n \,\square\, P_m$ such that $(C_{ijk} \cap K^{[c-1,c+1]}) = d[c-1,i,j,k].code$ for a fixed $S_{jk} \subseteq K^{[c,c+1]}$. By induction hypothesis

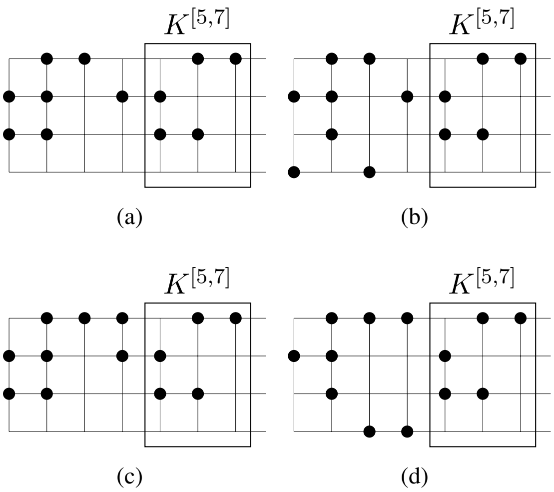


Figure 3: For $n = 4$ and $c = 5$, four possible configurations for a fixed $S_{jk\ell} = C_{jk\ell} \cap K^{[5,7]}$. Vertices in $C_{jk\ell}$ are represented by black circles. (a) is an optimal configuration, (b), (c), and (d) are not and they will be discarded. $C_{jk\ell} \cap K^{[1,4]}$ must have minimum size possible to be optimal.

$|C_{ijk} \cap K^{[1,c-2]}|$ have minimum size for all $1 \leq i \leq 2^n$ and they are stored in $d[c-1, i, j, k].size$. For a specific subset S_ℓ of K^{c+2}, if $d[c-1, i, j, k].code \cup S_\ell$ dominates and pairwise separates the vertices of $K^{[c,c+1]}$ and s is the minimum value for all $1 \leq i \leq 2^n$ (condition of line 15), then $C_{jk\ell} = C_{ijk} \cup S_\ell$ and at line 16, $d[c, j, k, \ell].code$ receives $S_{jk\ell} = S_{jk} \cup S_\ell$ and at line 17 $d[c, j, k, \ell].size$ receives $s + |S_\ell|$. See Figure 3 for an illustration. Since all possibilities were evaluated, $C_{jk\ell} \cap K^{[1,c-1]}$ is minimum for the set $S_{jk\ell}$ and all vertices in $K^{[1,c+1]}$ are dominated and pairwise separated by $C_{jk\ell}$. Hence, the above statements are true for all $1 \leq c \leq m-2$, for $m \geq 3$.

□

Algorithm 2: MINIMUM-ID

Input: Table d.
Output: Minimum cardinality of an identifying code of $K_n \square P_m$

begin
  $ic \leftarrow +\infty$;   `/* obtaining the size of a minimum ID code from` d `*/`
  foreach *combination* ijk *with* $1 \leq i, j, k \leq 2^n$ **do**
    $S \leftarrow d[m-2, i, j, k].code$;
    $s \leftarrow d[m-2, i, j, k].size$;
    if *for every* $v \in (K^{[m-1,m]})$, $(N[v] \cap S)$ *are nonempty and pairwise distinct* **then**
      if $ic > s$ **then** $ic \leftarrow s$;
    end
  end
  return ic;
end

Theorem 5. *The* ic *number returned by Algorithm 2 corresponds to* $\gamma^{ID}(K_n \square P_m)$.

Proof. Since the vertices of all adjacent K_n-layers are dominated and pairwise separated, by proposition 2, ic is an identifying code of $K_n \square P_m$. By Lemma 4, each entry $d[m-2, j, k, \ell].size$ for $1 \leq j, k, \ell \leq 2^n$ contains either $+\infty$ or the cardinality of an K^m-almost identifying code $C_{jk\ell}$ of

$K_n \square P_m$ such that it has minimum size for $C_{jk\ell} \cap K^{[1,m-3]}$ and

$$C_{jk\ell} \cap (K^{[m-2,m]}) = d[m-2, j, k, \ell].code.$$

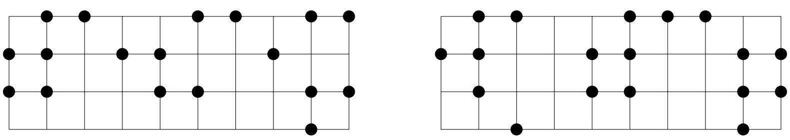
For $1 \leq j, k, \ell \leq 2^n$, all subsets of $K^{[m-2,m]}$ are evaluated to verify for every $v \in (K^{[m-1,m]})$, if $(N[v] \cap d[m-2, j, k, \ell].code)$ are nonempty and pairwise distinct. In a positive case $C_{jk\ell}$ is an identifying code of $K_n \square P_m$, otherwise the set is not considered. The cardinalities of all non empty $C_{jk\ell}$ sets are compared and hence Algorithm 2 returns a cardinality of an minimum identifying code of $K_n \square P_m$. □

Algorithm 1 proceeds in $\mathcal{O}(m)$ steps. At each step, at most 2^{4n} sets are evaluated. For each set, it is necessary to verify, in a brute-force approach, if two sets are dominated and separated. This can be done in $\mathcal{O}(n \log n)$ time. Thus the complexity of the algorithm is $\mathcal{O}(2^{4n} mn \log n)$, that is linear for a fixed n (proving Theorem 3). Considering the case $n = 4$, we have an algorithm that is $\mathcal{O}(2^{19} m)$. In a brute-force approach we would enumerate all 2^{mn} subsets of $V(K_n \square P_m)$ and check each one to see whether it is an identifying code, that can be done in $\mathcal{O}(mn \log mn)$ time. Thus, this approach requires $\mathcal{O}(2^{mn} mn \log mn)$ time, which is impractical even for some small values of m and n.

It was proved [Gravier et al., 2008] that minimum identifying codes of $K_n \square K_n$, for $n \geq 5$ and odd, are unique (up to row and column permutations). With an adaptation of Algorithms 1 and 2 we could check that for $K_4 \square P_m$ there are many optimal solutions for minimum identifying codes. The number of optimal solutions obtained for $3 \leq m \leq 38$ are given in Table 1. Many solutions are the same by row permutations. See in Figure 4 two distinct minimum identifying codes of $K_4 \square P_{10}$.

Table 1: Number of γ^{ID}-sets of $(K_4 \square P_m)$.

m	Solutions	m	Solutions	m	Solutions
3	96	15	29571072	27	80621568000
4	289	16	679829760	28	4516527734784
5	384	17	76142592	29	23219011584
6	9840	18	2985984	30	2650837155840
7	384	19	14929920	31	179631303229440
8	112512	20	2358927360	32	1393140695040
9	801024	21	89579520	33	104880275324928
10	565056	22	1875197952	34	6961818160594944
11	62976	23	82723700736	35	62691331276800
12	165888	24	2687385600	36	4487166864654336
13	17500032	25	103446429696	37	274396659898908672
14	2543616	26	3492705484800	38	2507653251072000



(a) (b)

Figure 4: Distinct minimum identifying codes of $K_4 \square P_{10}$

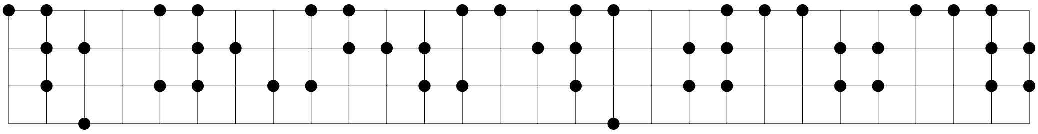


Figure 5: γ^{ID}-set of $(K_4 \,\square\, P_{28})$. $\gamma^{ID}(K_4 \,\square\, P_{28}) = 47$.

4. Concluding remarks

We have presented a dynamic programming algorithm to efficiently determine the minimum cardinality of an identifying code for the Cartesian product $K_4 \,\square\, P_m$. Although this is a restricted graph class, the approach can be used to solve problems efficiently in graphs with a similar structure.

Our dynamic-programming solution returns the value of an optimal solution, but it does not return the γ^{ID}-set of $K_n \,\square\, P_m$. We can easily extend the dynamic-programming approach to record a choice of vertices that lead to the optimal value.

We have implemented our algorithms and determined $\gamma^{ID}(K_4 \,\square\, P_m)$ quickly (less than one minute) for $3 \leq m \leq 10000$ (See in Table 2, $\gamma^{ID}(K_4 \,\square\, P_m)$ for some small graphs). From these results, we could state the conjecture below.

Conjecture 6. *For $m \geq 3$, $\gamma^{ID}(K_4 \,\square\, P_m) = 18\lfloor \frac{m}{11} \rfloor + a$, where a is a positive integer at most 17.*

Table 2: For $3 \leq m \leq 38$, $\gamma^{ID}(K_4 \,\square\, P_m)$.

m	γ^{ID}	m	γ^{ID}	m	γ^{ID}	m	γ^{ID}
3	6	12	21	21	35	30	50
4	8	13	23	22	37	31	52
5	10	14	24	23	39	32	53
6	12	15	26	24	40	33	55
7	12	16	28	25	42	34	57
8	15	17	29	26	44	35	58
9	17	18	30	27	45	36	60
10	18	19	32	28	47	37	62
11	19	20	34	29	48	38	63

In Figure 5 one can see a γ^{ID}-set of $K_4 \,\square\, P_{28}$ and in Figure 6 a block of 11 adjacent K_4-layers that frequently appears on the γ^{ID}-sets obtained by the algorithms when $m \geq 17$. Each block has 18 vertices into the γ^{ID}-set and can be connected to obtain minimum identifying codes for greater graphs. This can explain that $\gamma^{ID}(K_4 \,\square\, P_m) = 18\lfloor \frac{m}{11} \rfloor + \Theta(1)$ for the checked graphs.

Our next steps include to prove Conjecture 6 and use the approach to determine minimum identifying codes in other graph products.

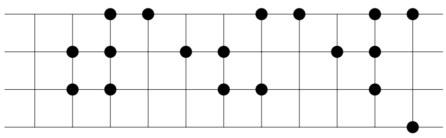


Figure 6: A block of 11 adjacent K_4-layers. Each block has 18 vertices into the γ^{ID}-set and can be connected to obtain a part of a minimum identifying code of greater graphs.

Acknowledgments

The authors would like to thank FAPEG - Fundação de Apoio à Pesquisa do Estado de Goiás for the partial financial support.

References

Auger, D. (2010). Minimal identifying codes in trees and planar graphs with large girth. *European Journal of Combinatorics*, 31(5):1372 – 1384.

Ben-Haim, Y. and Litsyn, S. (2005). Exact minimum density of codes identifying vertices in the square grid. *SIAM J. Discrete Math.*, 19(1):69–82.

Berger-Wolf, T. Y., Hart, W. E., and Saia, J. (2005). Discrete sensor placement problems in distribution networks. *Journal of Mathematical and Computer Modelling*, 42(13):1385–1396.

Blidia, M., Chellali, M., Maffray, F., Moncel, J., and Semri, A. (2007). Locating-domination and identifying codes in trees. *Australas. J. Comb.*, 39:219–232.

Cappelle, M. R., Coelho, E. M., Coelho, H., Penso, L. D., and Rautenbach, D. (2015). Identifying codes in the complementary prism of cycles. *arXiv preprint arXiv:1507.05083*.

Charon, I., Gravier, S., Hudry, O., Lobstein, A., Mollard, M., and Moncel, J. (2006). A linear algorithm for minimum 1-identifying codes in oriented trees. *Discrete Applied Mathematics*, 154(8):1246 – 1253. ISSN 0166-218X.

Charon, I., Hudry, O., and Lobstein, A. (2003). Minimizing the size of an identifying or locating-dominating code in a graph is np-hard. *Theoretical Computer Science*, 290(3):2109 – 2120. ISSN 0304-3975.

Cohen, G., Gravier, S., Honkala, I., Lobstein, A., Mollard, M., Payan, C., and Zémor, G. (1999). Improved identifying codes for the grids. *Electr. J. Comb.*, 6(1).

Daniel, M., Gravier, S., and Moncel, J. (2004). Identifying codes in some subgraphs of the square lattice. *Theoretical Computer Science*, 319(1):411 – 421. ISSN 0304-3975.

Feng, M. and Wang, K. (2014). Identifying codes of corona product graphs. *Discrete Applied Mathematics*, 169:88–96.

Feng, M., Xu, M., and Wang, K. (2012). Identifying codes of lexicographic product of graphs. *the electronic journal of combinatorics*, 19(4):P56.

Goddard, W. and Wash, K. (2013). Id codes in cartesian products of cliques. *Australas. J. Comb.*, 85:97–106.

Gravier, S., Moncel, J., and Semri, A. (2008). Identifying codes of cartesian product of two cliques of the same size. *The Electronic Journal of Combinatorics [electronic only]*, 15(1):Research Paper N4, 7 p.–Research Paper N4, 7 p.

Gravier, S., Moncel, J., and Semri, A. (2006). Identifying codes of cycles. *European Journal of Combinatorics*, 27(5):767 – 776. ISSN 0195-6698.

Hedetniemi, J. (2016). On identifying codes in the cartesian product of a path and a complete graph. *Journal of Combinatorial Optimization*, 31(4):1405–1416. ISSN 1573-2886.

Junnila, V. and Laihonen, T. (2012a). Optimal lower bound for 2-identifying codes in the hexagonal grid. *Electr. J. Comb.*, 19(2):1–16.

Junnila, V. and Laihonen, T. (2012b). Optimal identifying codes in cycles and paths. *Graphs and Combinatorics*, 28(4):469–481. ISSN 1435-5914.

Karpovsky, M. G., Chakrabarty, K., and Levitin, L. B. (1998). On a new class of codes for identifying vertices in graphs. *Information Theory, IEEE Transactions on*, 44(2):599–611.

Lobstein, A. (2016). Watching systems, identifying, locating-dominating and discriminating codes in graphs. `http://perso.telecom-paristech.fr/~lobstein/debutBIBidetlocdom.pdf`. Acessado: 2017-03-23.

Martin, R. and Stanton, B. (2010). Lower bounds for identifying codes in infinite grids. *Journal of Combinatorial Optimization*, 17.

Rall, D. F. and Wash, K. (2016). On minimum identifying codes in some Cartesian product graphs. *ArXiv e-prints*.

Rall, D. F. and Wash, K. (2014). Identifying codes of the direct product of two cliques. *European Journal of Combinatorics*, 36:159–171.

Ray, S., Starobinski, D., Trachtenberg, A., and Ungrangsi, R. (2004). Robust location detection with sensor networks. *IEEE Journal on Selected Areas in Communications (Special Issue on Fundamental Performance Limits of Wireless Sensor Networks)*, 22(6):1016–1025.

Ray, S., Ungrangsi, R., Pellegrini, D., Trachtenberg, A., and Starobinski, D. (2003). Robust location detection in emergency sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, p. 1044–1053. IEEE.