Identifying codes in hereditary classes of graphs and VC-dimension

Nicolas Bousquet[1,5], Aurélie Lagoutte [*2], Zhentao Li [†3], Aline Parreau [‡4] and Stéphan Thomassé [*2]

[1]LIRMM, Université Montpellier 2, France
[2]LIP, UMR 5668 ENS Lyon - CNRS - UCBL - INRIA, Université de Lyon, France
[3]École Normale Supérieure, Paris, France
[4]Université de Liege, Belgique
[5]Department of Mathematics and Statistics, McGill University and GERAD, Montréal, Canada

Abstract

An identifying code of a graph is a subset of its vertices such that every vertex of the graph is uniquely identified by the set of its neighbours within the code. We show a dichotomy for the size of the smallest identifying code in classes of graphs closed under induced subgraphs. Our dichotomy is derived from the VC-dimension of the considered class $\mathcal{C}$, that is the maximum VC-dimension over the hypergraphs formed by the **closed** neighbourhoods of elements of $\mathcal{C}$. We show that hereditary classes with infinite VC-dimension have infinitely many graphs with an identifying code of size logarithmic in the number of vertices while classes with finite VC-dimension have a polynomial lower bound.

We then turn to approximation algorithms. We show that Min Id Code (the problem of finding a smallest identifying code in a given graph from some class $\mathcal{C}$) is log-APX-hard for any hereditary class of infinite VC-dimension. For hereditary classes of finite VC-dimension, the only known previous results show that we can approximate Min Id Code within a constant factor in some particular classes, e.g. line graphs, planar graphs and unit interval graphs. We prove that Min Id Code can be approximate within a factor 6 for interval graphs. In contrast, we show that Min Id Code on C_4-free bipartite graphs (a class of finite VC-dimension) cannot be approximated to within a factor of $c \log(|V|)$ for some $c > 0$.

[*]Partially supported by ANR Project Stint under Contract ANR-13-BS02-0007.
[†]Partially supported by a FQRNT B3 postdoctoral fellowship program.
[‡]Supported by a FNRS post-doctoral grant at the University of Liège.

1 Introduction

Let $G = (V, E)$ be a graph. An *identifying code* of G is a subset C of vertices of G such that, for each vertex $v \in V$, the set of vertices in C at distance at most 1 from v, is non-empty and uniquely identifies v. In other words, for each vertex $v \in V(G)$, we have $N[v] \cap C \neq \emptyset$ (C is a *dominating set*) and for each pair $u, v \in V(G)$, we have $N[u] \cap C \neq N[v] \cap C$ (C is a *separating set*), where $N[v]$ denotes the closed neighbourhood of v in G (v and all its neighbours). We say that a set X of vertices *distinguishes* $u \in V(G)$ from $v \in V(G)$ if $N[u] \cap X \neq N[v] \cap X$. This concept was introduced in 1998 by Karpovsky, Chakrabarty and Levitin [19] and has applications in various areas such as fault-diagnosis [19], routing in networks [21] or analysis of RNA structures [17]. For a complete survey on these results, the reader is referred to the online bibliography of Lobstein [22].

Two vertices u and v are *twins* if $N[u] = N[v]$. The whole vertex set $V(G)$ is an identifying code if and only if G is twin-free. Since supersets of identifying codes are identifying, an identifying code exists for G if and only if it is twin-free. A natural problem in the study of identifying codes is to find one of a minimum size. Given a twin-free graph G, the smallest size of an identifying code of G is called the *identifying code number* of G and is denoted by $\gamma^{\mathrm{ID}}(G)$. The problem of determining γ^{ID} is called the Min Id Code problem, and its decision version is NP-complete [8].

Let $X \subseteq V$. We denote by $G[X]$ the graph induced by the subset of vertices X. In this paper, we focus on hereditary classes of graphs, that is classes closed under taking induced subgraphs. We consider the two following problems: finding good lower bounds and approximation algorithms for the identifying code number.

1.1 Previous work

In the class of all graphs, the best lower bound is $\gamma^{\mathrm{ID}}(G) \geq \log(|V(G)| + 1)$, since all the vertices of the graphs have distinct non-empty neighbourhood within the code. Moncel [25] characterized all graphs reaching this lower bound. As for approximation algorithms, the general problem Min Id Code is known to be log-APX-hard [20, 21, 30]. In particular, there is no $(1 - \varepsilon)\log(|V|)$-approximation algorithm for Min Id Code. The problem Min Id Code remains log-APX-hard even in split graphs, bipartite graphs or co-bipartite graphs (complement of bipartite graphs) [13].

On the positive side, there always exists a $\mathcal{O}(\log |V(G)|)$ approximation for Min Id Code [30]. Moreover, even if in the general case Min Id Code is hard to evaluate, there exist several constant approximation algorithms for restricted classes of graphs, such as planar graphs [26] or line graphs [14].

For the remainder of this article, n denotes the number of vertices of G. Table 1 gives an overview of the currently known results for some restricted hereditary classes of graphs. The order of magnitude of all lower bounds are best possible (there are infinite families of graphs reaching the lower bounds). Min Id Code for line graphs and planar graphs have a polynomial time constant factor approximation algorithm with the best known constant written in parenthesis. From this table, we observe two behaviours: a class either

1. has a logarithmic lower bound on the size of identifying codes, and Min Id Code is log-APX-hard in this class (for example split, bipartite, co-bipartite graphs), or

2. there is a polynomial lower-bound on $\gamma^{\mathrm{ID}}(G)$ and a constant factor approximation algorithm to compute $\gamma^{\mathrm{ID}}(G)$.

Graph class	Lower bound	Complexity	Approximability	References
All graphs	$\Theta(\log(n))$	NP-c	log-APX-hard	[19, 20]
Chordal	$\Theta(\log(n))$	NP-c	log-APX-hard	[13]
Split graphs	$\Theta(\log(n))$	NP-c	log-APX-hard	[13]
Bipartite	$\Theta(\log(n))$	NP-c	log-APX-hard	[13]
Co-bipartite	$\Theta(\log(n))$	NP-c	log-APX-hard	[13]
Claw-free	$\Theta(\log(n))$	NP-c	log-APX-hard	[13]
Interval	$\Theta(n^{1/2})$	NP-c	open	[15]
Unit interval	$\Theta(n)$	open	PTAS	[12, 15]
Permutation	$\Theta(n^{1/2})$	NP-c	open	[15]
Line graphs	$\Theta(n^{1/2})$	NP-c	APX(4)	[14]
Planar	$\Theta(n)$	NP-c	APX(7)	[3, 26]

Table 1: Known lower bounds on $\gamma^{\mathrm{ID}}(G)$ and approximability of $\gamma^{\mathrm{ID}}(G)$.

1.2 Our results

The aim of this paper is to shed some light on the validity of such a dichotomy for all classes of graphs using the VC-dimension of the class of graphs.

VC-dimension Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. A subset $X \subseteq V$ of vertices is *shattered* if for every subset S of X, there is some hyperedge e such that $e \cap X = S$. The *VC-dimension* of $\mathcal{H}$ is the size of the largest shattered set of $\mathcal{H}$. We define the *VC-dimension of a graph* as the VC-dimension of the closed neighbourhood hypergraph of G (vertices are the vertices of G and hyperedges are the closed neighbourhoods of vertices of G), a classical way to define the VC-dimension of a graph (see [1, 6]).

By *a shattered set of a graph* G, we mean a shattered set of the hypergraph of the closed neighbourhoods of G. The VC-dimension of a class of graphs $\mathcal{C}$, denoted by $dim(\mathcal{C})$, is the maximum of the VC-dimension of the graphs over $\mathcal{C}$. If it is unbounded, we say that $\mathcal{C}$ has *infinite VC-dimension*.

Dichotomy for lower bounds First we will prove in Section 2 that there is indeed such a dichotomy on the minimum size of identifying codes: it is always either logarithmic or polynomial, where the exponent of the polynomial depends on the VC-dimension of the

Graph class	VC dim	IC-lower bound	IC-approx
Girth ≥ 5	2	$\Theta(n^{\frac{1}{2}})$ (opt,new)	open
Interval	2	$\Theta(n^{\frac{1}{2}})$ (opt)	6 (Thm. 21)
Chordal bipartite	3	$\Omega(n^{\frac{1}{3}})$ (new)	open
Unit disk	3	$\Omega(n^{\frac{1}{3}})$ (new)	open
C_4-free bipartite	2	$\Theta(n^{\frac{1}{2}})$ (opt,new)	no $c\log(n)$-approx (Thm. 15)
Undirected path	3	$\Omega(n^{\frac{1}{3}})$(new)	open

Table 2: Overview of the results obtained in this paper.

class of graphs. In particular, our theorem provides new lower bounds for graphs of girth at least 5, chordal bipartite graphs, unit disk graphs and undirected path graphs. Moreover, these bounds are tight for interval graphs and graphs of girth at least 5.

Approximation hardness We then try to extend this dichotomy result for constant factor approximations. First, we show in Section 3 that Min Id Code is log-APX-hard for any hereditary class with a logarithmic lower bound. The proof essentially consists in proving that a hereditary class with infinite VC-dimension contains one of these three classes, for which Min Id Code has been shown to be log-APX-hard [13]: the bipartite graphs, the co-bipartite graphs, or the split graphs. Unfortunately, the dichotomy does not extend to approximation since we show in Section 4 that C_4-free bipartite graphs have a polynomial lower bound on the size of identifying codes but Min Id Code is not approximable to within a factor $c \log n$ for some $c > 0$ (under some complexity assumption) in this class. Thus, a constant factor approximation is not always possible in the second case.

Approximation algorithm Finally, in Section 5, we conclude the paper with some positive result when the lower bound is polynomial by proving that there exists a 6-approximation algorithm for interval graphs, a problem left open in [13].

The results obtained in this paper are detailed in Table 2.

2 Dichotomy for lower bound

The following lemma, due to Sauer [27] and Shellah [29] will be useful to prove that the size of an identifying code cannot be too small if the VC-dimension is bounded. The *trace* of a set X on Y is $X \cap Y$. By extension, *the trace of a vertex x on Y* is the intersection of $N[x]$ with Y.

Lemma 1 (Sauer's lemma [27, 29]). *Let $\mathcal{H} = (V, \mathcal{E})$ be an hypergraph of VC-dimension d. For every set $X \subseteq V$, the number of (distinct) traces of $\mathcal{E}$ on X is at most*

$$\sum_{i=0}^{d} \binom{|X|}{i} \leq |X|^d + 1.$$

Let us now prove the main result of this section.

Theorem 2. *For every hereditary class of graphs $\mathcal{C}$, either*

1. *for every $k \in \mathbb{N}$, there exists a graph $G_k \in \mathcal{C}$ with more than $2^k - 1$ vertices and an identifying code of size $2k$, or*
2. *there exists $\varepsilon > 0$ such that no twin-free graph $G \in \mathcal{C}$ with n vertices has an identifying code of size smaller than n^ε.*

Proof. Let $\mathcal{C}$ be an hereditary class of graphs. The class $\mathcal{C}$ either has finite or infinite VC-dimension. First, suppose that $\mathcal{C}$ has infinite VC-dimension. We will show that $\mathcal{C}$ satisfies the first conclusion. By definition of infinite VC-dimension, there is a graph $H_k \in \mathcal{C}$ with VC-dimension k for each k. So there exists a set of vertices X of size k of H_k which is shattered. Let Y be a set of $2^k - 1$ vertices whose closed neighbourhoods have all possible traces on X except the empty set, meaning that for every $X' \subseteq X$, add a vertex y in Y such that $N[y] \cap X = X'$. Choose Y so that $|X \cap Y|$ is maximized. Let $G_k = H_k[X \cup Y]$. The graph G_k has at least $2^k - 1$ vertices since $|Y| = 2^k - 1$. By choice of Y, X dominates $X \cup Y$ and X distinguishes every pair of vertices of Y. By maximality of $|X \cap Y|$, X also distinguishes every vertex in X from every vertex in Y (otherwise a vertex of Y would have the same neighbourhood in X as a vertex $x \in X$ and thus can be replaced by x, contradicting the maximality of $|X \cap Y|$). For each $x \in X$, the vertex $y_x \in Y$ whose closed neighbourhood intersects X in exactly $\{x\}$ distinguishes x from all vertices in $X - x$. So $X \cup \{y_x | x \in X\}$ is an identifying code of size at most $2k$, as required.

Now suppose that the VC-dimension of $\mathcal{C}$ is bounded by d. For any identifying code C of a twin-free graph $G \in \mathcal{C}$, the traces of vertices of G on C are different. Hence, by Lemma 1, $n \leq \sum_{i=0}^{d} \binom{|C|}{i} \leq |C|^d + 1$. Therefore, $|C| \geq (n-1)^{\frac{1}{d}}$, proving that $\mathcal{C}$ satisfies the second claim. □

The proof gives in fact the lower bound $\gamma^{\mathrm{ID}}(G) \in \Omega(n^{\frac{1}{dim(\mathcal{C})}})$ for the second item. So if we can bound the VC-dimension of the class, then we immediately obtain lower bounds on the size of identifying codes. Lemma 3 provides such bounds for several classes of graphs.

Let us give some definitions. The *girth* of a graph is the length of a shortest cycle. A *chordal bipartite graph* is a bipartite graph without induced cycle of length at least 6. A *unit disk graph* is a graph of intersection of unit disks in the plane. An *interval graph* is a graph of intersection of segments on a line. An *undirected path graph* is a graph of vertex-intersection of paths in an undirected tree (*i.e.* two vertices are adjacent if their corresponding paths have at least one vertex in common).

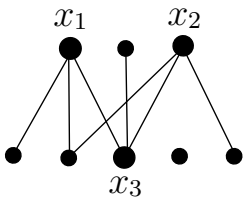


Figure 1: The set x_1, x_2, x_3 is shattered in this chordal bipartite graph.

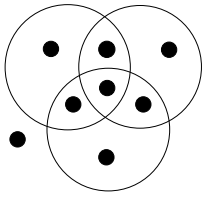


Figure 2: A set of three vertices shattered by disks in the plane.

Lemma 3. *The following upper bounds hold and are tight:*

- *The VC-dimension of graphs of girth at least* 5 *is at most* 2*.*
- *The VC-dimension of chordal bipartite graphs is at most* 3*.*
- *The VC-dimension of unit disk graphs is at most* 3*.*
- *The VC-dimension of interval graphs is at most* 2*.*
- *The VC-dimension of undirected path graphs is at most* 3*.*

Proof.

- Let G be a graph of girth at least 5. Assume by contradiction that a set $\{x_1, x_2, x_3\}$ of three vertices is shattered. Since the girth is at least 5, $x_1x_2x_3$ is not a clique. We may assume without loss of generality that x_1 and x_2 are not adjacent. Since $\{x_1, x_2, x_3\}$ is shattered, there is a vertex y_1 adjacent to both x_1 and x_2 and not x_3 (one closed neighbourhood must have trace $\{x_1, x_2\}$ on $\{x_1, x_2, x_3\}$) and a vertex y_2 adjacent to $\{x_1, x_2, x_3\}$ (one closed neighbourhood must have trace $\{x_1, x_2, x_3\}$). Note that both y_1 and y_2 are distinct from x_1 and x_2 since x_1 and x_2 are not adjacent. Moreover y_1 and y_2 are distinct since they do not have the same neighbourhood in $\{x_1, x_2, x_3\}$. So $x_1y_1x_2y_2x_1$ is a cycle of length 4, a contradiction with the girth assumption.
  This bound is tight, for instance with the path on six vertices.
- Let $G = (A \cup B, E)$ be a chordal bipartite graph. Assume by contradiction that $\{x_1, x_2, x_3, x_4\}$ is a shattered set of four vertices. Since there is a vertex whose closed neighbourhood contains the whole set of vertices, it means that at least three vertices, say x_1, x_2, x_3 are on the same side of the bipartite graph. Since a subset of a shattered set is shattered, $\{x_1, x_2, x_3\}$ is shattered. Thus there is a vertex incident to x_1, x_2 and not x_3, a vertex incident to x_1, x_3 and not x_2, and a vertex incident to x_2, x_3 and not x_1. It provides an induced cycle of length 6, a contradiction.
  Moreover the bound is tight, see Figure 1.

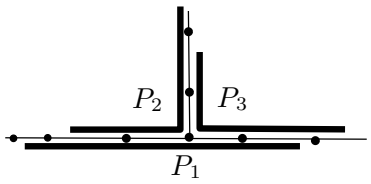


Figure 3: Paths P_1, P_2, P_3 are shattered by the eight points which are paths of length 0.

- Let G be a unit disk graph. Let us rephrase the adjacency and shattering conditions in this class: let x_1 and x_2 be any two vertices of a unit disk graph and denote by c_1 and c_2 their respective centers in a representation of the unit disk graph in the plane. The vertices x_1 and x_2 are adjacent if and only if c_1 and c_2 are at distance at most 2. Thus if a set of unit disks is shattered then for every subset of centers, there exists a point at distance at most 2 from these centers and more than 2 from the others. In other words, there exist points in all possible intersections of balls of radius 2.
  A classical result ensures that the VC-dimension of a hypergraph whose hyperedges can be represented as a set of disks in the plane (and vertices as points of the plane) has VC-dimension at most 3 (see [24] for instance). Thus unit disk graphs have VC-dimension at most 3, and the bound can be reached (see Figure 2).

- Let G be an interval graph. Assume by contradiction that there is a shattered set $\{I_1, I_2, I_3\}$ of G. Assume that I_1 starts before I_2 and that I_2 starts before I_3. Since there is an interval J intersecting both I_1 and I_3 but not I_2, J must start after I_2 and thus I_1 contains I_2. Then there is no interval intersecting I_2 but not I_1, a contradiction. Thus interval graphs have VC-dimension at most 2, and the bound is again reached with the path on six vertices.

- Let $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$ be a shattered set of four paths of a tree T. Assume first that P_2, P_3, P_4 all intersect P_1 and consider the restriction of T to P_1, which is in fact an interval graph. To ensure all possible intersections with P_1, the set $\{P_2, P_3, P_4\}$ is a shattered set of size three in an interval graph, a contradiction.

  Thus at least one path, say P_2, does not intersect P_1 and lies in a connected component C of the forest $F = T \setminus P_1$. If P_3 does not intersect C, then there is no path intersecting both P_2 and P_3 but not P_1. Thus P_3 intersects C. If moreover P_3 intersects P_1, then no path can intersect both P_1 and P_2 but not P_3. Thus P_3 is also included in C. Let P be a path intersecting P_1, P_2 and P_3. Assume first that P intersects the three paths in the order P_1, P_2 and P_3 (the case P_1, P_3, P_2) is the same. Then no path can intersect P_1 and P_3 without intersecting P_2. Assume now that P intersects the three paths in the order P_2, P_1, P_3. Similarly, no path can intersect P_2 and P_3 without intersecting P_1. Hence the path P cannot exist, a contradiction. Finally the bound of 3 can be reached, as shown in Figure 3.

□

Lemma 3 and Theorem 2 imply new lower bounds for many classes: $\Omega(n^{\frac{1}{2}})$ for graphs with girth at least 5, $\Omega(n^{\frac{1}{3}})$ for chordal bipartite graphs, $\Omega(n^{\frac{1}{3}})$ for unit-disk graphs, $\Omega(n^{\frac{1}{2}})$ for interval graphs, $\Omega(n^{\frac{1}{3}})$ for permutation graphs, and $\Omega(n^{\frac{1}{3}})$ for undirected path graphs.

The exponent given by Theorem 2 is sharp for several classes of graphs. Indeed, Foucaud et *al.* [15] proved that there are infintely many interval graphs with identifying codes of size $\Theta(n^{1/2})$. The bound is also tight for C_4-free bipartite graphs (which have girth at least 5): the following construction is a C_4-free bipartite graphs with an identifying code of size $\Theta(n^{\frac{1}{2}})$. Let $G = (X \cup Y, E)$ be a bipartite graph where Y has size n, X has size $\frac{n(n-1)}{2}$, and edges satisfy the following rule: for every pair u, v of vertices of Y, there is exactly one vertex of X adjacent to both u and v. The graph G does not contain any triangle (since it is bipartite) nor C_4 (since neighbourhoods intersect on at most one vertex). One can easily check that the set Y is an identifying code of the graph. Indeed vertices of X are adjacent to precisely two neighbours on Y and vertices of Y have precisely one neighbour on Y in their closed neighbourhood. Finally, it is also sharp for the class of all graphs of VC-dimension at most d. Indeed, consider the bipartite graph made with a stable set A of size d and a stable set B of size $\sum_{i=2}^{d} \binom{d}{i}$ representing all the subsets of A of size at least 2. Each vertex of B is adjacent to the vertices of A corresponding to its subset. This graph has VC-dimension d and the set A is an identifying code of size of order $n^{1/d}$.

Nevertheless, the bounds given by Theorem 2 are not necessarily tight. For instance, permutations graphs can have VC-dimension 3 but Foucaud et *al.* [15] recently proved that the exact lower bound is $\Omega(n^{\frac{1}{2}})$.

Note that a similar bound can be obtained for identifying codes in hypergraphs where we want to distinguish all the hyperedges with a minimal subset of vertices. Such a problem generalizes identifying codes and is known under the name of *test covering* or *discriminating codes* (see Section 4).

3 Inapproximability in infinite VC-dimension

Given a minimization problem P and a function $f : \mathbb{N} \to \mathbb{N}$, a *factor f approximation algorithm* (also called an *f-approximation*) is an algorithm that outputs a solution of value at most $f(n) \cdot OPT_I$ for every instance I of P of size n, where OPT_I is the value of an optimal solution of I. log-APX is a class of problems consisting of all problems that admit a logarithmic factor polynomial time approximation algorithm. We use the AP-reductions introduced in [9] which have now become standard. Its definition restricted minimization problems is defined as follows:

Definition 4 ([4])**.** Let P and Q be two minimization problems. An *AP-reduction* from P to Q is a triple (f, g, α) where

1. α is a constant,
2. f maps pairs consisting of an instance of P and a constant $r > 1$ to instances of Q, and

3. g maps triples consisting of a constant $r > 1$, an instance I_P of P and a solution to $f(I_P, r)$ to a solution of I_P

in such a way that

1. $f(I_P, r)$ has a solution if I_P does,
2. $f(\cdot, r)$ and $g(\cdot, \cdot, r)$ are computable in polynomial time for all fixed r, and
3. if $f(I_P, r) \leq r \cdot OPT(f(I_P, r))$ then $g(f(I_P, r), r, SOL_Q) \leq (1 + \alpha(r-1)) \cdot OPT(I_P)$.

A problem Q is log-*APX-hard* if any problem P in log-APX can be reduced to Q by an AP-reduction.

Theorem 5 ([9])**.** *Any optimization problem P that is* log-*APX-hard with respect to AP-reduction is NP-hard to approximate within a factor $c \cdot \log(n)$ where n is the size of the input, for some constant $c > 0$.*

We show that Min Id Code is log-APX-hard for classes with infinite VC-dimension. To prove this result, we will prove that a class with infinite VC-dimension contains either all the bipartite graphs, or all the co-bipartite graphs or all the split graphs. Since the problem Min Id Code is log-APX-hard in these three classes (see [13]), it implies that it is log-APX-hard for all classes with infinite VC-dimension.

Theorem 6. *Let $\mathcal{C}$ be an hereditary class. If $\mathcal{C}$ has infinite VC-dimension, then $\mathcal{C}$ must contain either all the bipartite graphs, or all the co-bipartite graphs or all the split graphs.*

Note that this result implies the first part of Theorem 2. We say that a bipartite graph $H = (A \cup B, E)$ is a *bipartisation* of G if removing all edges in A and in B in G yields H for some partition A, B of $V(G)$.

Lemma 7. *For any hereditary class $\mathcal{C}$ of graphs with infinite VC-dimension and any bipartite graph H, $\mathcal{C}$ contains a graph G whose bipartisation is H.*

Proof. Let $H = (A \cup B, E)$ be a bipartite graph with $|B| \leq |A| = k$. Since $\mathcal{C}$ has infinite VC-dimension, it contains a graph G with a shattered set S of size (at least) $\ell = k + \lceil \log(2k) \rceil$. Let A' be the first k vertices in S, and let us number $Y_1, \ldots, Y_{2k}$ some $2k$ distinct subsets of $S \setminus A'$ (they exists since $|S \setminus A'| = \lceil \log(2k) \rceil$).

By definition of a shattered set, for each $i \in \{1, \ldots 2k\}$ and for each $X \subseteq A'$ there is a vertex x_i of G such that $N[x_i] \cap S = X \cup Y_i$. Thus there are $2k$ vertices of G whose closed neighbourhoods intersect A' in exactly X. Hence there are at least k such vertices in $V(G) \setminus A'$. Label the vertices of A' by vertices in A, *i.e.* choose an arbitrary bijection between A and A'. Now for each $b \in B$, choosing $X = N(b)$ gives k vertices in $V(G) \setminus A'$ whose closed neighbourhoods intersect A' in exactly $N(b)$. So we can choose one "representative" for each b so that all the selected vertices are distinct (since $|B| \leq k$).

Note that we need k vertices in $V \setminus A'$ since up to $|B|$ vertices of B may have the same neighbourhood in A.

Since $\mathcal{C}$ is closed under taking induced subgraphs, the subgraph of G induced by A' and the set B' of all chosen vertices is in $\mathcal{C}$. The bipartisation of this graph is H, as required. □

Next we show that we can further restrict H' and now require both sides of H' to be stable sets or cliques. For a bipartite graph $H = (A \cup B, E)$, write $H^{1,0}$ for the graph obtained from H by adding a clique on A, $H^{0,1}$ the graph obtained from H by adding a clique on B and $H^{1,1}$ the graph obtained from H by adding a clique on both A and B. We also write sometimes $H^{0,0}$ for H. We show that, for each bipartite graph H, $\mathcal{C}$ contains one of these four graphs. To do so, we need the classical theorem of Erdős and Hajnal [10] as well as its bipartite version by Erdős, Hajnal and Pach [11].

Theorem 8 (Erdős, Hajnal [10])**.** *For every graph H, there exists a constant $c(H)$ such that all graphs on n vertices contain either H as an induced subgraph, a stable set of size at least $2^{c(H)\sqrt{2\log n}}$ or a clique of size at least $2^{c(H)\sqrt{2\log n}}$.*

Theorem 9 (Erdős, Hajnal, Pach [11])**.** *Let H be a bipartite graph with vertex classes U_1 and U_2, ($k = |U_1| \leq |U_2| = \ell$) and let $n > \ell^{k+1}$. Then in any bipartite graph G with vertex classes V_1 and V_2 ($|V_1| = |V_2| = n$) which contains no two subsets $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ that induce an isomorphic copy of H, there exist $V_1' \subseteq V_1$ and $V_2' \subseteq V_2$ of size $\left\lfloor \left(\frac{n}{\ell}\right)^{\frac{1}{k}} \right\rfloor$ such that either all edges between V_1' and V_2' belong to G or none of them does.*

We continue with the following technical lemmas:

Lemma 10. *For n large enough, there exists a bipartite graph $G_0 = (A \cup B, E_0)$ with $2n$ vertices ($|A| = |B| = n$) such that there is no complete nor empty bipartite graphs $G_0[A' \cup B']$ with $A' \subseteq A$ and $B' \subseteq B$ and $|A'| = |B'| = \lfloor 2\log n \rfloor$.*

Proof. Let us show its existence with a probabilistic argument. Let A and B be two stable sets each of size n and for every $a \in A, b \in B$, put the edge ab with probability $\frac{1}{2}$. Given two subsets $A' \subseteq A$, $B' \subseteq B$ with $|A'| = |B'| = \lfloor 2\log n \rfloor$, the probability that $A' \cup B'$ induces a complete bipartite graph is $\left(\frac{1}{2}\right)^{\lfloor 2\log n \rfloor^2}$. The same probability holds for $A' \cup B'$ inducing an empty bipartite graph. Thus the probability that there exists a complete or empty bipartite graph with each part of size $2\log n$ is at most

$$\binom{n}{\lfloor 2\log n \rfloor}^2 \frac{2}{2^{\lfloor 2\log n \rfloor^2}} \leq \left(\frac{n \cdot e}{\lfloor 2\log n \rfloor}\right)^{2\cdot\lfloor 2\log n \rfloor} \cdot \frac{2}{2^{\lfloor 2\log n \rfloor^2}} = 2^{-4\log n \cdot \log\log n + \mathcal{O}(\log n)}$$

using the inequality

$$\binom{n}{l} \leq \frac{n^l}{l!} \leq \left(\frac{n \cdot e}{l}\right)^l$$

This probability is strictly less than 1 for n large enough, so there exists a graph $G_0 = (A \cup B, E_0)$ for which the event does not occur. □

Lemma 11. *Let $\mathcal{C}$ be an hereditary class with infinite VC-dimension. For any bipartite graph $H = (H_\ell \cup H_r, E)$, one of the four graphs $H^{0,0}$, $H^{1,0}$, $H^{0,1}$ or $H^{1,1}$ is in $\mathcal{C}$.*

Proof. Suppose by contradiction that Lemma 11 is false for H with $|H_\ell| \leq |H_r| = k$. Let $c(H)$ be the constant from Theorem 8 and pick n large enough so that $2^{\frac{c(H)}{k}\sqrt{2\log n}} > k^{k+1}$, and $(2^{\frac{c(H)}{k}\sqrt{2\log n}})/k^{\frac{1}{k}} > 2\log n$ and n satisfies the condition of Lemma 10. Let G_0 be a bipartite graph as in Lemma 10, *i.e.* G_0 has n vertices on both sides and does not contain a complete or an empty bipartite graph with $2\log n$ vertices on each side. By Lemma 7, $\mathcal{C}$ contains a graph G whose bipartisation is G_0. Let A, B certify this bipartisation.

Since G contains no copy of $H^{0,0}$, neither does $G[A]$. So by Theorem 8, $G[A]$ contains a clique or stable set A' of size at least $n' = 2^{c(H)\sqrt{2\log n}}$. Similarly, $G[B]$ also contains a clique or stable set B' of this size. Assume without loss of generality that in both cases, A' and B' induce stable sets. By assumption, the bipartisation of $G[A' \cup B']$ contains no copy of $H^{0,0}$ (since the class $\mathcal{C}$ is closed under induced subgraphs). So by Theorem 9 and since $n' > k^{k+1}$, the bipartisation of $G[A' \cup B']$ contains a complete bipartite graph or an empty bipartite graph where each bipartition has size

$$\left(\frac{n'}{k}\right)^{\frac{1}{k}} = \frac{2^{\frac{c(H)}{k}\sqrt{2\log n}}}{k^{\frac{1}{k}}} > 2\log n$$

which is a contradiction to G_0 having no such subgraph. □

Proof of Theorem 6. Let H_n be the disjoint union of every bipartite graphs of size at most n. For every n, Lemma 11 ensures that $H_n^{a_n,b_n}$ is in $\mathcal{C}$ (for some $a_n, b_n \in \{0,1\}$) and hence there exist $a, b \in \{0,1\}$ for which $H_n^{a,b}$ is in $\mathcal{C}$ for infinitely many values of n.

If $a = b = 0$, all bipartite graphs are in $\mathcal{C}$; if $a \neq b$, all split graphs are in $\mathcal{C}$ and if $a = b = 1$, all co-bipartite graphs are in $\mathcal{C}$: indeed let $H^{a,b}$ be a bipartite graph on n vertices (resp. split graph, co-bipartite graph, depending on the value of a and b). Then there exists $n' \geq n$ such that $H_{n'}^{a,b}$ is in $\mathcal{C}$. But $H^{a,b}$ is an induced subgraph of $H_{n'}$ so $H^{a,b}$ is an induced subgraph of $H_{n'}^{a,b}$. The theorem follows. □

Foucaud [13] proved that Min Id Code is log-APX-hard for bipartite graphs, split graphs and co-bipartite graphs. So the following is a direct corollary of Theorem 6.

Corollary 12. Min Id Code *is* log*-APX-hard when the input graph is restricted to an hereditary class of graphs with infinite VC-dimension.*

4 Inapproximability for C_4-free bipartite graphs

In this section, we examine the complexity of approximating Min Id Code in classes of finite VC-dimension. Previous results suggest that all these classes may have a constant

factor approximation algorithm : this is the case for line graphs [14], planar graphs [26] or unit interval graphs (since any solution has size at least $\frac{n}{2}$) for instance.

However, we show that this intuition is false: the class $\mathcal{C}$ of C_4-free bipartite graphs (whose VC-dimension is bounded by 2) does not admit such an approximation algorithm. In fact, Min Id Code in $\mathcal{C}$ is hard to approximate to within a $c \log n$ factor (for some $c > 0$) in polynomial time, unless $NP \subseteq ZTIME(n^{O(\log \log n)})$.

Observation 13. *The class of C_4-free bipartite graphs has VC-dimension at most 2.*

Proof. Let G be a C_4-free bipartite graph. Then it has no triangle and no C_4, so we can apply the result of Lemma 3 for graphs of girth at least 5. □

We provide a polynomial time gap preserving reduction (in fact, an AP-reduction) from the following minimization problem:

Problem 14. Set cover with intersection 1 (Set Cover$_1$)
Instance: A set X and a family S of subsets of X where any two sets in S intersect in at most one element.
Solution: A subset S' of sets in S whose union contain X.
Measure: The size of S'.

Anil Kumar, Arya and Hariharan [2] have shown that this problem cannot be approximated to within a $c \log n$ factor (for some $c > 0$) in polynomial time, unless $NP \subseteq ZTIME(n^{O(\log \log n)})$.

Theorem 15. Min Id Code *with input restricted to C_4-free bipartite graphs cannot be approximated to within a $c \log n$ factor (for some $c > 0$) in polynomial time, unless $NP \subseteq ZTIME(n^{O(\log \log n)})$ where n is the size of the input.*

To give a flavour of our reduction from Set Cover$_1$ to Min Id Code, we first give an easier reduction to the Discriminating code problem [7]. The Discriminating code is often a way to design reductions which gives an overview of most complicated ones for Min Id Code : indeed a discriminating code consists in identifying vertices of a set X using vertices of a set Y.

Problem 16. Discriminating code
Instance: A bipartite graph $G = (X \cup Y, E)$.
Solution: A subset Y' of Y which dominates X and such that for every pair of vertices x_1, x_2 of X, $N[x_1] \cap Y' \neq N[x_2] \cap Y'$. Such a set is called a discriminating code.
Measure: The size of Y'.

Lemma 17. Discriminating code *with input restricted to C_4-free bipartite graphs cannot be approximated to within a $c \log n$ factor (for some $c > 0$) in polynomial time, unless $NP \subseteq ZTIME(n^{O(\log \log n)})$.*

Proof. Let $I_{SC} = (X, S)$ be an instance of Set Cover$_1$. The proof is decomposed into five steps: construct an instance I_{DC} of discriminating code that has polynomial size in $|I_{SC}|$; check that this instance is indeed a C_4-free bipartite graph; for every solution of I_{SC}, construct a solution of I_{DC}; and vice-versa; finally check that if the solution of I_{DC} is not too big with respect to the optimal one, then so is the solution of I_{SC}.

Construct the instance of Discriminating Code Let $G = (X \cup S, E)$ be the membership bipartite graph of the instance I_{SC}, that is to say that for every $x \in X, s \in S$, there is an edge $xs \in E$ if and only if $x \in s$. In the following, n denotes the size of X and we assume that $n \geq 2$ and that no $s \in S$ is connected to all of X (meaning that the optimal solution to I_{SC} has size at least 2). Note that in the other case, we can compute the optimal solution in polynomial time. Moreover, we assume that for every $x \in X$, there exists $s \in S$ such that x belong to s, otherwise there is no solution. The following construction is illustrated on Figure 4. Let $G_1, \ldots, G_\ell$ be $\ell = 2n^2 - 1$ disjoint copies of G. Denote by $X_i \cup S_i$ the i-th copy of $X \cup S$. Let X'_1 and X'_2 be copies of X. For each $x'' \in X'_2$, add an edge between x'' and its copy x' in X'_1, and add edges between x'' and its copies in all G_i for $i \leq \ell$. In other words, $G[X_i \cup X'_2]$ induces a matching for every i. Let G_{DC} be this bipartite graph with parts $X_{DC} = X_1 \cup ... \cup X_\ell \cup X'_1$ and $Y_{DC} = S_1 \cup ... \cup S_\ell \cup X'_2$. Clearly, the size of I_{DC} is polynomial in n an thus in the size of I_{SC}.

Check that the instance is C_4-free First note that the initial graph G is C_4-free. Indeed every C_4 must have two vertices in S and two vertices in C, a contradiction since the neighbourhoods of two vertices of S intersect on at most one vertex. Further, the graph G_{DC} is C_4-free. Indeed, no C_4 can contain two vertices of $S_1 \cup \cdots \cup S_l$ since any vertex $s \in S_i$ only has neighbours in X_i, and $G_{DC}[X_i \cup S_i]$ is a copy of the C_4-free graph G. Moreover, two vertices $x'' \in X'_2$ and $s \in S_i$ have at most one common neighbour $x_i \in X_i$, the copy of x''. Finally, each pair of vertices of X'_2 have disjoint neighbourhoods. Thus no vertex can be part of a C_4.

Transforming a solution of I_{SC} into a solution of I_{DC} Let D be a set cover of S of size SOL_{SC}. Construct C as the union of ℓ copies of D (one for each G_i), and finally add X'_2. Then C is dominating X_{DC} since X'_2 is, moreover C is separating all the pair of vertices of X_{DC}. Indeed, two vertices x_i, y_j inherited from two different elements $x, y \in X$ are separated by $x'' \in X'_2$, the copy of x in X'_2. Two vertices $x_i \in X_i, x_j \in X_j$ with $i \neq j$, or two vertices $x_i \in X_i, x' \in X'_1$ inherited from the same element $x \in X$ are separated by a neighbour $s_i \in S_i$ of x_i, where s_i is the i-th copy of an element $s \in D$ containing x. Consequently, C is indeed a discriminating code. For later use, observe that if D is the optimal solution of Set Cover$_1$ of size OPT_{SC}, we can derive $OPT_{DC} \leq n + \ell \cdot OPT_{SC}$.

Transforming a solution of I_{DC} into a solution of I_{SC} Let C be a solution of I_{DC} of size SOL_{DC}. We construct a set cover candidate $D_1 = S_1 \cap C$. Every vertex $x_1 \in X_1$ is dominated and separated from its copy x' in X'_1, and $x'' \in X'_2$ cannot achieve this goal,

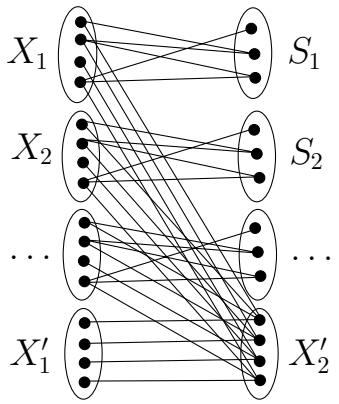


Figure 4: Construction of the proof of Lemma 17.

thus there exists $s \in S_1 \cap C$ which is linked to x_1. Thus D_1 is a set cover. The same can be done for each $i \leq \ell$, and we choose D as the minimum size such constructed set cover. For later use, observe that $X'_2 \subseteq C$ since X'_1 is dominated. Consequently if SOL_{SC} is the size of D, we have $|X'_2| + \ell \cdot SOL_{SC} \leq SOL_{DC}$ or equivalently $SOL_{SC} \leq \frac{SOL_{DC}-n}{\ell}$.

Concluding on the size of the solutions Now suppose that we can obtain a solution SOL_{DC} of Discriminating Code satisfying $SOL_{DC} \leq r \cdot OPT_{DC}$ for some value r. The above discussion gives

$$SOL_{SC} \leq \frac{SOL_{DC} - n}{\ell} \leq \frac{r \cdot OPT_{DC} - n}{\ell} \leq \frac{r(n + \ell \cdot OPT_{SC}) - n}{\ell} \leq 2r \cdot OPT_{SC}$$

In particular if $r = c' \log n$ for some well-chosen constant c', we obtain a contradiction with Set Cover$_1$ approximation hardness.

□

Let us now adapt this reduction into a reduction to identifying codes.

Proof of Theorem 15. Let I_{SC} be an instance of Set Cover$_1$.

Construct the instance of Min Id Code First construct the same graph G_{DC} as in the proof of Lemma 17. Now for identifying codes, we need to identify vertices in both sides and not only on the side of X_{DC}. For that, we add to G_{DC} a set $Z = \{z_1, \dots, z_{2n^2}\}$ in part X_{DC}. We have to be careful when we connect the vertices of Z to the vertices of the graph since we do not want to create a C_4. We aim at choosing edges between Z and $S_1 \cup ... \cup S_\ell$ such that each vertex $s \in S_1 \cup ... \cup S_\ell$ is adjacent to exactly two vertices of Z, and no two vertices $s, s' \in S_i$ share a neighbour in Z. The following claim (whose proof is postponed at the end of the section) reaches the goal:

Claim 18. *There exists a numbering of the vertices in $S_1 \cup ... \cup S_\ell$ such that:*

- *Each vertex $s \in S_1 \cup ... \cup S_\ell$ is numbered $s_{i,j}$ with $i < j \in \{1, \dots, 2n^2\}$, where the pair $\{i, j\}$ is distinct for every vertex.*
- *Two vertices $s_{j,k}$ and $s_{j',k'}$ cannot both belong to the same set S_i if one of j, k is equal to one of j', k'.*

Using the numbering of the claim, we just have to add the edges $z_k s_{k,l}$ and $z_l s_{k,l}$ for every $k, l \in \{1, \dots, 2n^2\}$. Note that every vertex of S_i is connected to precisely two vertices of Z, and that every vertex $z \in Z$ has at most one neighbour in each S_i. Let G_{IC} be this new graph. It has polynomial size in n.

Check that the instance is C_4-free Since we only add edges from $Z \subset X_{DC}$ to $S_1 \cup \dots \cup S_\ell \subset Y_{DC}$, the graph is indeed bipartite. Since G_{DC} was C_4-free, any hypothetical C_4 must intersect Z, say in $z_k \in Z$. z_k share with any $z_l \in Z$ at most one common neighbour $s_{k,l}$, so the C_4 must intersect X_i for some i, or X'_1. On the one hand, vertices in X'_1 have degree one. On the other hand, x_i only has neighbours in S_i, and z_k has one only neighbour in each S_i, so they cannot be in the same C_4.

Transforming a solution of I_{SC} into a solution of I_{IC} A set C containing $X_1 \cup X'_2 \cup Z$ is a good candidate to be an identifying code because it has the following properties:

- For every $z_k \in Z$, z_k is identified by z_k being the only vertex of $Z \cap N[z_k] \cap C$.
- For every copy $x'' \in X'_2$ of an element $x \in X$, x'' is dominated by $\{x", x_1\}$ in C a where $x_1 \in X_1$ is the copy of x. Thus it is separated from all the other vertices except maybe x_1.
- For every $s_{k,l} \in S_1 \cup \dots \cup S_\ell$, $s_{k,l}$ is identified by $\{z_k, z_l\}$.
- For every $x' \in X'_1$, x' is dominated by $x'' \in X'_2$.
- For every $x_i \in X_i$, x_i is dominated by $x'' \in X'_2$

Thus C is a dominating set, and the only sets of vertices that may be not separated are of the form $\{x'\} \cup \{x_2, \dots, x_\ell\}$ and $\{x", x_1\}$ for any element $x \in X$.

Let D be a set cover of the initial instance and $D_1, ..., D_\ell$ be the respective copies in the graphs G_i. Then $C = D_1 \cup ... \cup D_\ell \cup X_1 \cup X'_2 \cup Z$ is an identifying code of G_{IC}. Indeed, every vertex x_i is separated from x', $x"$ and x_j $(i \neq j)$ by the element of S_i that covers it in the set cover D_i. Hence any solution for Set Cover$_1$ of size SOL_{SC} gives a solution for Min Id Code of size $SOL_{IC} = \ell \cdot SOL_{SC} + 2n + 2n^2$. In particular $OPT_{IC} \leq \ell \cdot OPT_{SC} + 2n + 2n^2$.

Transforming a solution of I_{IC} into a solution of I_{SC} Let C be an identifying code of G_{IC}. We define $D_i = C \cap S_i$ as a set cover candidate. Unfortunately, D_i may not be a set cover, in which case we iteratively modify C until all D_i meet the condition. If D_i is not a set cover of X_i, then there is a vertex x_i not covered. This vertex must be separated with its copy x' in X'_1, hence x_i (case 1) or x' (case 2) must belong to C (if both occur, case 1 has priority on case 2). Then choose any neighbour $s \in S_i$ of x_i, add this vertex to C and remove x_i (in case 1) or x' (in case 2). We thus get a new set C' and claim that $C' \cup Z \cup X_1 \cup X'_2$ is an identifying code. Thanks to the above discussion, we just have to show that the sets $\{x'\} \cup \{x_2, \ldots, x_\ell\}$ and $\{x", x_1\}$ are separated.

Observe first that x_i is now separated from x' and from x_j by s for $j \neq i$. Moreover, since C was separating x_{j_1} from x_{j_2} for $j_1, j_2 \neq i$, then C' still does (because nothing changed in their neighbourhood). We also have C' that separates x' from x_j for $j \neq i$: the vertex separating those two vertices was not x_i, so in case 1 it still belongs to C'. In case 2, we have removed x' but then x_i was not in C, and C was separating x_i and x_j so there exists a vertex in $(N[x_j] \setminus \{x'\}) \cap C$, and this vertex separates x_j from x' in C'. Finally, C' still separates x_1 from $x"$ since these two vertices cannot be separated by x_i nor x'.

Hence we can assume that all the sets $C' \cap S_i$ are set covers where C' has size at most $|C| + 2n + 2n^2$. Since there are at most $|C|$ vertices of C' which are in $S_1 \cup \ldots \cup S_\ell$, It means that an identifying code with $|C| = SOL_{IC}$ vertices of G_{IC} gives a solution of set cover with $SOL_{SC} \leq \frac{SOL_{IC}}{\ell}$ vertices.

Concluding on the size of the solutions Assume now that $SOL_{IC} \leq r \cdot OPT_{IC}$ for some value r, then:

$$SOL_{SC} \leq \frac{SOL_{IC}}{\ell} \leq \frac{r \cdot OPT_{IC}}{\ell} \leq \frac{r((2n^2-1)OPT_{SC} + 2n + 2n^2)}{2n^2 - 1} \leq 2r \cdot OPT_{SC}$$

As before for discriminating codes, it achieves the proof of Theorem 15. □

An *edge colouring* of a graph with k colours is a function $c : E \to \{1, \ldots, k\}$ such that no two edges sharing an endpoint are given the same colour, that is $c(uv) \neq c(uv')$ for every pair of edges uv, uv'.

Proof of Claim 18. We first need to convince ourselves that $|S| \leq n^2$ in the instance (X, S) of Set Cover$_1$. Indeed every pair of elements of X appears in at most one $s \in S$, thus $|S| \leq \frac{n(n+1)}{2} \leq n^2$ (one for each pair plus n additional singletons).

Now the idea of the proof is the following: we will represent our problem using a clique on $2n^2$ vertices. The vertices of the clique represent vertices of Z and edges of the clique represent vertices of $S_1 \cup \cdots \cup S_\ell$.

The edges of K_{2n^2} can be partitioned into $2n^2-1$ perfect matchings, or equivalently there exists an edge colouring c of K_{2n^2} with $2n^2 - 1$ colours such that each colour class contain n^2 edges. Then label the vertices by $\{z_1, \ldots, z_{2n^2}\}$ and create a set $S' = S'_1 \cup \cdots \cup S'_{2n^2-1}$

of elements $s_{j,k}$ with $j, k \in \mathbb{N}$ according to the following rule:

$$S'_i = \{s_{j,k} | c(z_j z_k) = i\} \text{ for every } i \in \{1, \dots 2n^2 - 1\}$$

Observe that $|S'_i| = n^2$. Since a colour class is a matching, the indices of every pair of edges in a same colour class are pairwise distinct. In other words, there cannot be two vertices $s_{j,k}$ and $s_{j',k'}$ in the same set S'_i if one of j, k is equal to one of j', k'. Now choose arbitrarily $|S| \leq n^2$ vertices in S'_i to form S_i. □

5 Constant approximation algorithm for interval graphs

We now focus on the class of interval graphs and provide a constant factor approximation algorithm for Min Id Code via a linear programming approach. More precisely we show that Min Id Code has a 6-approximation algorithm. The existence of a constant approximation algorithm was left open in [13].

Let us recall that an *interval graph* is a graph which can be represented as an intersection of segments in the real line. We put an arbitrary order on the real line. The *begin date of an interval* x is the first point p of the real line (in the order) such that $p \in x$. The *end date of* x is the last point which is in x. By abuse of notations, we will denote by v both the vertex of the graph and the interval in the representation on the real line. Note that there exist many representations as intersections of segments for a same interval graphs, we choose arbitrarily one of them which can be found in linear time [5].

Let G be an interval graph together with an interval representation. We denote its vertex set by $\{1, \dots, n\}$ and $U \bigtriangledown U'$ stands for the symmetric difference of U and U' for $U, U' \subseteq V$. Let us express Min Id Code in terms of an integer program P, where x_i is the decision variable corresponding to vertex i:

Integer program	P
Objective function:	$\min \sum_{i \in V} x_i$
Separation constraint:	$\sum_{i \in N[j] \bigtriangledown N[k]} x_i \geq 1 \quad \forall j \neq k \in V$
Domination constraint:	$\sum_{i \in N[j]} x_i \geq 1 \quad \forall j \in V$
Integrality:	$x_i \in \{0, 1\} \quad \forall i \in V$

Let us denote by P^* the linear programming relaxation of P, where the integrality constraint is replaced by a non-negativity constraint $x_i \geq 0, \forall i \in V$. Recall that even if an integer linear programming cannot be solved in polynomial time, its fractional relaxation can on the contrary be solved, using for instance the ellipsoid method. Our goal is to

construct a feasible solution for P of value at most $6 \cdot OPT(P^*)$. To achieve this goal, we decompose P into two subproblems:

$$P_{inter}$$

$$\min \sum_{i \in V} x_i$$

$$\forall \boldsymbol{jk} \in \boldsymbol{E} \quad \sum_{i \in N[j] \bigtriangledown N[k]} x_i \geq 1$$

(Separation constraints for intersecting pairs)

$$x_i \in \{0,1\} \quad \forall i \in V$$

$$P_{disj}$$

$$\min \sum_{i \in V} x_i$$

$$\forall \boldsymbol{jk} \notin \boldsymbol{E} \quad \sum_{i \in N[j] \bigtriangledown N[k]} x_i \geq 1$$

(Separation constraints for non-intersecting pairs)

$$\sum_{i \in N[j]} x_i \geq 1 \quad \forall j \in V \text{ (Domination constraints)}$$

$$x_i \in \{0,1\} \quad \forall i \in V$$

The reason why intersecting intervals play a special role is that the symmetric difference of $N[j]$ and $N[k]$ can be expressed in some sense by an union of 2 intervals which greatly helps. We denote by P^*_{inter} (resp. P^*_{disj}) the linear programming relaxation of P_{inter} (resp. P_{disj}).

Lemma 19. *Given an optimal solution $(x^*_1, \ldots, x^*_n)$ of P^*_{inter} of cost $OPT(P^*_{inter})$, there exists a polynomial time algorithm that computes a solution of P_{inter} of value at most $4 \cdot OPT(P^*_{inter})$.*

Proof. We follow the ideas of the proof of [16] where the problem is translated in terms of Rectangle Stabbing Problem. Note that our problem can also be viewed as the transversal of 2-intervals (union of 2 intervals) and, in this respect, topological bounds can be found in [18], even if this does not provide an approximation algorithm.

Le $x^* = (x^*_1, \ldots, x^*_n)$ be an optimal solution of P^*_{inter}. For every $jk \in E$, Figure 5 shows how to partition $N[j] \bigtriangledown N[k]$ into two parts L_{jk} (stands for Left) and R_{jk} (stands for Right). The set L_{jk} is composed of the intervals that end between the begin dates of j and k, and R_{jk} is composed of the intervals that begin between the end dates of j and k. L_{jk} and R_{jk} are obviously disjoint subsets.

Let us now define two subsets of vertices L and R as follows:

$$L = \left\{ jk \in E \;\middle|\; \sum_{i \in L_{jk}} x^*_i \geq \frac{1}{2} \right\} \qquad \text{and} \qquad R = \left\{ jk \in E \;\middle|\; \sum_{i \in R_{jk}} x^*_i \geq \frac{1}{2} \right\}$$

Since all the constraints of P^*_{inter} are satisfied by $(x^*_1, \ldots, x^*_n)$ and since for every edge jk, we have $L_{jk} \cup R_{jk} = N[j] \bigtriangledown N[k]$, all the edges are in L or in R (they can be in both of them). Based on this, we define now the following two integer linear programmings:

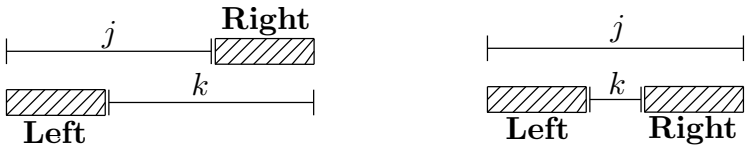


Figure 5: Given two intersecting intervals j and k, one can construct two areas **Left** and **Right** partitioning $N[j] \bigtriangledown N[k]$ between L_{jk} the set of intervals that end in **Left**, and R_{jk} the set of intervals that begin in **Right**. This figure shows how to find **Left** and **Right** depending on the configuration of j and k: either one is included in the other, or not.

$$P_L \qquad\qquad P_R$$

$$\min \sum_{i \in V} x_i \qquad\qquad \min \sum_{i \in V} x_i$$

$$\forall \boldsymbol{jk \in L} \sum_{i \in L_{jk}} x_i \geq 1 \qquad\qquad \forall \boldsymbol{jk \in R} \sum_{i \in R_{jk}} x_i \geq 1$$

$$x_i \in \{0,1\} \quad \forall i \in V \qquad\qquad x_i \in \{0,1\} \quad \forall i \in V$$

According to the previous notations, we denote by P_L^* (resp. P_R^*) the linear programming relaxation of P_L (resp. P_R). Consider now the 0/1 matrix M obtained from P_L^*, where each row represents an edge $jk \in L$, each column represents a vertex $i \in V$, and $M_{jk,i} = 1$ if $i \in L_{jk}$, 0 otherwise. By sorting the vertices of V (and thus, the columns of the matrix) by interval end date, the 1's on each row become consecutive. Indeed the 1's on the line of the constraint jk correspond to intervals that end between the begin dates of j and k which are obviously consecutive if we sort the intervals by end date. A matrix which has consecutive 1's on each row is said to have the *interval property*. Such a matrix is totally unimodular (which means that all the squared determinants of the matrix have values $-1, 0$ or 1) and this implies that there is an optimal solution of P_L^* where all the variables are integer. In particular, $OPT(P_L^*) = OPT(P_L)$. (See [23, 28] for more details about totally unimodular matrices and their use in linear programming.)

The same holds for P_R^* by sorting vertices by interval begin date. Solving P_L^* and P_R^* can be done in polynomial time, this gives us integer solutions $(x_1^L, \ldots, x_n^L)$ for P_L^* and $(x_1^R, \ldots, x_n^R)$ for P_R^* and setting $x_i = x_i^L + x_i^R$ builds a feasible solution for P_{inter} of objective value $SOL(P_{inter}) = OPT(P_L^*) + OPT(P_R^*)$. Observe now that $(2x_1^*, \ldots, 2x_n^*)$ is a feasible solution for both P_L^* and P_R^* so $OPT(P_L^*) \leq 2{\cdot}OPT(P_{inter}^*)$ and $OPT(P_R^*) \leq 2{\cdot}OPT(P_{inter}^*)$. This concludes the proof by $SOL(P_{inter}) \leq 4 \cdot OPT(P_{inter}^*)$. □

Let us now focus on the second subproblem:

Lemma 20. *Given the interval representation of G, one can compute in polynomial time a feasible solution for P_{disj} of size at most $2 \cdot OPT(P_{disj}^*)$.*

Proof. We construct a set S of intervals in the following way. Initially, set $S = \emptyset$. While V is not empty, do the following: select the interval v that ends first ; put it in S and remove $N[v]$ from V. Once V is empty, output S.

Observe that this algorithm compute a maximal (with respect to inclusion) independent set S with the property that for every vertex $v \in V$, there exists $s \in S$ such that the end date of s is in the interval v (s is the vertex selected at the same step as v was deleted). We now claim that on the one hand, S is a feasible solution for P_{disj} and on the other hand that $|S| \leq 2 \cdot OPT(P^*_{disj})$.

Let $j < k \in V, jk \notin E$. Up to symmetry, suppose that the interval j starts before k (and thus, ends before k starts). Then there exists $s \in S$ such that the end date of s is in j, implying $s \in N[j] \setminus N[k]$. As a maximal independent set, S is also a dominating set so S is a feasible solution for P_{disj} with objective value $\alpha \in \mathbb{N}$.

Let us number $\{s_1, \ldots, s_\alpha\}$ the elements of S by order of interval end date. Then observe that for every $i \in V$, there exist at most two distinct indices j such that i is in $N[s_j] \bigtriangledown N[s_{j+1}]$. Indeed, this happens if and only if i begins between the end date of s_j and the end date of s_{j+1}, or i ends between the begin date of s_j and the begin date of s_{j+1}. Then consider an optimal solution $(x^*_1, \ldots, x^*_n)$ of P^*_{disj}, we can derive:

$$2 \cdot OPT(P^*_{disj}) = \sum_{i \in V} 2 \cdot x^*_i \geq \sum_{j=1}^{\alpha} \sum_{i \in N[s_j] \bigtriangledown N[s_{j+1}]} x^*_i \geq \alpha \ .$$

□

Theorem 21. *There exists a polynomial time 6-approximation algorithm for* Min Id Code *on interval graphs.*

Proof. By Lemmas 19 and 20, we can construct in polynomial time a solution S_{inter} for P_{inter} and a solution S_{disj} for P_{disj} of cost respectively at most $4 \cdot OPT(P^*_{inter})$ and $2 \cdot OPT(P^*_{disj})$. The set $S_{inter} \cup S_{disj}$ gives a feasible solution for P of cost at most $6 \cdot OPT(P^*)$ thus as most $6 \cdot OPT(P)$. Algorithm 1 sums up the different steps of the approximation algorithm. One can check that this algorithm runs in polynomial time since computing an interval representation of an interval graph can be done in linear time [5] and solving a linear programming relaxation can also be done in polynomial time using the ellipsoid method for instance. □

Observe that the bound on the cost of the solution is in fact $6 \cdot OPT(P^*)$ which is slightly tighter than $6 \cdot OPT(P)$.

References

[1] N. Alon, G. Brightwell, H. Kierstead, A. Kostochka, and P. Winkler. Dominating sets in k-majority tournaments. *Journal of Combinatorial Theory, Series B*, 96(3):374–387, 2006.

Algorithm 1: 6-Approximation algorithm for MIN ID CODE in interval graphs

Input: An interval graph $G = (V, E)$
Output: An identifying code C of size at most $6 \cdot OPT$
begin
  Compute an interval representation of G;
begin `// Computation of the solution for` P_{inter}
  Solve P^*_{inter};
  Compute L and R;
  Solve P^*_L and P^*_R;
  Set S_{inter} to be the union of the solution to $P_biblio.bibL^*$ and P^*_R;
begin `// Computation of the solution for` P_{disj}
  $S_{disj} = \emptyset$;
  while $V \neq \emptyset$ **do**
    Select the interval v that ends first;
    Add v to S_{disj};
    Remove $N[v]$ from V;
return $C = S_{inter} \cup S_{disj}$

[2] V. S. Anil Kumar, S. Arya, and R. Hariharan. Hardness of set cover with intersection 1. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, pages 624–635, 2000.

[3] D. Auger, I. Charon, O. Hudry, and A. Lobstein. Complexity results for identifying codes in planar graphs. *International Transactions in Operational Research*, 17(6):691–710, 2010.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation.* Springer, 1999.

[5] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.

[6] N. Bousquet and S. Thomassé. Scott's induced subdivision conjecture for maximal triangle-free graphs. *Combinatorics, Probability and Computing*, 21:512–514, 2012.

[7] E. Charbit, I. Charon, G. D. Cohen, and O. Hudry. Discriminating codes in bipartite graphs. *Electronic Notes in Discrete Mathematics*, 26:29–35, 2006.

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

[9] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in approximation classes. *SIAM Journal on Computing*, 28(5):1759–1782, 1999.

[10] P. Erdős and A. Hajnal. Ramsey-type theorems. *Discrete Applied Mathematics*, 25:37–52, 1989.

[11] P. Erdös, A. Hajnal, and J. Pach. A Ramsey-type theorem for bipartite graphs. *Geombinatorics*, 10:64–68, 2000.

[12] F. Foucaud. *Combinatorial and algorithmic aspects of identifying codes in graphs*. PhD thesis, University of Bordeaux, 2012.

[13] F. Foucaud. Decision and approximation complexities for identifying codes and locating-dominating sets in restricted graph classes, 2014. To be published in *Journal of Discrete Algorithms*.

[14] F. Foucaud, S. Gravier, R. Naserasr, A. Parreau, and P. Valicov. Identifying codes in line graphs. *Journal of Graph Theory*, 73(4):425–448, 2013.

[15] F. Foucaud, G. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. *ArXiv e-prints*, 2014.

[16] D. R. Gaur, T. Ibaraki, and R. Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *Journal of Algorithms*, 43(1):138–152, 2002.

[17] T. Haynes, D. Knisley, E. Seier, and Y. Zou. A quantitative analysis of secondary RNA structure using domination based parameters on trees. *BMC Bioinformatics*, 7(108), 2006.

[18] T. Kaiser. Transversals of d-intervals. *Discrete & Computational Geometry*, 18(2):195–203, 1997.

[19] 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):599–611, 1998.

[20] M. Laifenfeld and A. Trachtenberg. Identifying codes and covering problems. *IEEE Transactions on Information Theory*, 54(9):3929–3950, 2008.

[21] M. Laifenfeld, A. Trachtenberg, and T.Y. Berger-Wolf. Identifying codes and the set cover problem. In *Proceedings of the 44th Annual Allerton Conference on Communication, Control and Computing*, Monticello, USA, 2006.

[22] A. Lobstein. Watching systems, identifying, locating-dominating and discriminating codes in graphs : a bibliography. Online bibliography: www.infres.enst.fr/~lobstein/bibLOCDOMetID.html.

[23] R. K. Martin. *Large Scale Linear and Integer Optimization: A Unified Approach: A Unified Approach*. Springer, 1999.

[24] J. Matousek, R. Seidel, and E. Welz. How to net a lot with little: Small ϵ-nets for disks and halfspace. In *Symposium on Computational Complexity*, pages 16–22, 1990.

[25] J. Moncel. On graphs on n vertices having an identifying code of cardinality $\lceil \log_2(n+1) \rceil$. *Discrete Applied Mathematics*, 154(14):2032–2039, 2006.

[26] D. Rall and P. Slater. On location-domination numbers for certain classes of graphs. *Congressus Numerantium*, 45:97–106, 1984.

[27] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[28] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[29] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[30] J. Suomela. Approximability of identifying codes and locating–dominating codes. *Information Processing Letters*, 103(1):28–33, 2007.