# Decision and approximation complexity for identifying codes and locating-dominating sets in restricted graph classes[☆]

Florent Foucaud[a,b,c]

[a]Department of Mathematics, University of Johannesburg, Johannesburg (South Africa).
[b]LAMSADE - CNRS UMR 7243, PSL, Université Paris-Dauphine, Paris (France).
[c]Departament de Matemàtica Aplicada III, Universitat Politècnica de Catalunya, Barcelona (Spain).

**Abstract**

An identifying code is a subset of vertices of a graph with the property that each vertex is uniquely determined (identified) by its nonempty neighbourhood within the identifying code. When only vertices out of the code are asked to be identified, we get the related concept of a locating-dominating set. These notions are closely related to a number of similar and well-studied concepts such as the one of a test cover. In this paper, we study the decision problems Identifying Code and Locating-Dominating Set (which consist in deciding whether a given graph admits an identifying code or a locating-dominating set, respectively, with a given size) and their minimization variants Minimum Identifying Code and Minimum Locating-Dominating Set. These problems are known to be NP-hard, even when the input graph belongs to a number of specific graph classes such as planar bipartite graphs. Moreover, it is known that they are approximable within a logarithmic factor, but hard to approximate within any sub-logarithmic factor. We extend the latter result to the case where the input graph is bipartite, split or co-bipartite: both problems remain hard in these cases. Among other results, we also show that for bipartite graphs of bounded maximum degree (at least 3), the two problems are hard to approximate within some constant factor, a question which was open. We summarize all known results in the area, and we compare them to the ones for the related problem Dominating Set. In particular, our work exhibits important graph classes for which Dominating Set is efficiently solvable, but Identifying Code and Locating-Dominating Set are hard (whereas in all previous works, their complexity was the same). We also introduce graph classes for which the converse holds, and for which the complexities of Identifying Code and Locating-Dominating Set differ.

*Keywords:* Test cover, Separating system, Identifying code, Locating-dominating set, NP-completeness, Approximation.

## 1. Introduction

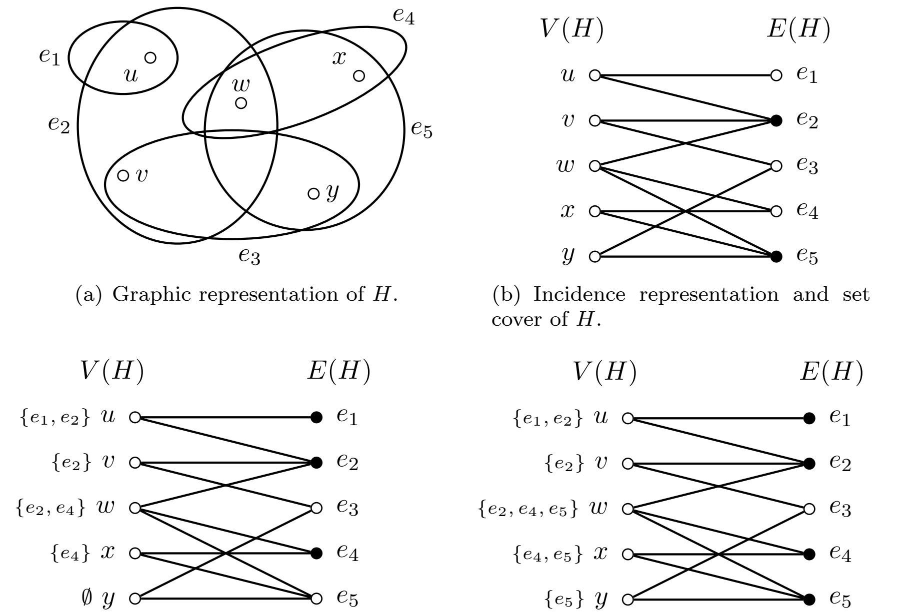
This paper studies the computational complexity of problems where one wants to find a set of vertices in a graph that uniquely identifies each vertex. In particular, we study this complexity according to the graph class of the input. We mainly focus on identifying codes, which are subsets of vertices that identify all vertices using the intersection of their closed neighbourhood with the code. The slightly less restrictive concept of locating-dominating sets will also be studied.

### 1.1. Definitions and problems

The graph-theoretic problems we will consider are special cases of more general ones, defined on hypergraphs, that we shall describe first.

[☆]Most results of this paper are from the author's PhD thesis [28], and were found while he was a PhD student at the LaBRI, University Bordeaux 1, 351 Cours de la Libération, 33405 Talence Cedex, France. An extended abstract of this paper containing most results about identifying codes appeared in the proceedings of IWOCA 2013 [29].

In this paper, to avoid confusion we will usually call hypergraphs and their vertex and edge sets $H = (I, A)$ and graphs $G = (V, E)$. Given a hypergraph $H$, a *set cover* of $H$ is a subset $\mathcal{S}$ of its edges such that each vertex $v$ belongs to at least one set $S$ of $\mathcal{S}$. We say that $S$ *dominates* $v$. A *test cover* of $H$ is a subset $\mathcal{T}$ of edges such that for each pair $u, v$ of distinct vertices of $H$, there is at least one set $T$ of $\mathcal{T}$ that contains exactly one of $u$ and $v$. We say that $T$ (and also $\mathcal{T}$) *separates* $u$ from $v$. A set of edges that is both a set cover and a test cover is called a *discriminating code* of $H$. It has to be mentioned that some hypergraphs may not admit any set cover (if some vertex is not part of any edge) or test cover (if two vertices belong to exactly the same set of edges). See Figure 1 for examples of these concepts.




(a) Graphic representation of $H$.

(b) Incidence representation and set cover of $H$.

(c) Incidence representation and test cover of $H$.

(d) Incidence representation and discriminating code of $H$.

Figure 1: A hypergraph $H$ with a set cover, a test cover and a discriminating code (black edges). In the two last figures, the sets next to the vertices represent the sets of edges of the solution containing that vertex. The test cover is not a set cover and hence it is not a discriminating code.

While set covers are a standard and widespread notion in combinatorics and theoretical computer science, test covers are less known. They were introduced under the name of *separating systems* by Rényi [53] (see also Bollobás and Scott [8] for more recent work in the same line of research). They were independetly studied in Garey and Johnson's book [33] under the name of *test collections* and further studied, see [24, 47]. Discriminating codes were more recently (and independently) introduced and studied in [12, 13]. As we will see later, the notions of test covers and discriminating codes are very similar in nature. Due to their properties enabling the unique identification of elements (vertices) of a system (hypergraph) using the set of their attributes (edges), they have had a number of interesting applications in the areas of testing individuals (such as patients or computers) for diseases or faults, see [12, 24, 47].

In this paper, we are mainly interested in special cases that can be defined over *graphs* rather than hypergraphs. The notion that we will mainly study, introduced in 1998 in [40], is the one of an identifying code. Given a graph $G$ and a vertex $v$ of $G$, we denote by $N(v)$ and by $N[v]$ the open and the closed neighbourhood of $v$, respectively. An *identifying code* of $G$ is a subset $\mathcal{C} \subseteq V(G)$ such that $\mathcal{C}$ is a *dominating set*, i.e. for each $v \in V(G)$, $N[v] \cap \mathcal{C} \neq \emptyset$ and $\mathcal{C}$ is a *separating code*, i.e. for each pair $u, v \in V(G)$, if $u \neq v$, then $N[u] \cap \mathcal{C} \neq N[v] \cap \mathcal{C}$. The minimum size of an identifying code of a given graph $G$ will be denoted $\gamma^{\text{ID}}(G)$. The notion of an identifying code is a generalization of the well-studied *locating-dominating sets*, introduced three decades ago [54, 55]. Given a graph $G$, a *locating-dominating set* of $G$ is a subset $\mathcal{C} \subseteq V(G)$

which is both a dominating set and which separates all vertices that are not in the code, i.e. for each pair $u, v \in V(G) \setminus \mathcal{C}$, if $u \neq v$, then $N(u) \cap \mathcal{C} \neq N(v) \cap \mathcal{C}$. The minimum size of a locating-dominating set of a given graph $G$ will be denoted $\gamma^{\text{LD}}(G)$. It is easily seen that any identifying code is a locating-dominating set. See Figure 2 for examples of these concepts.

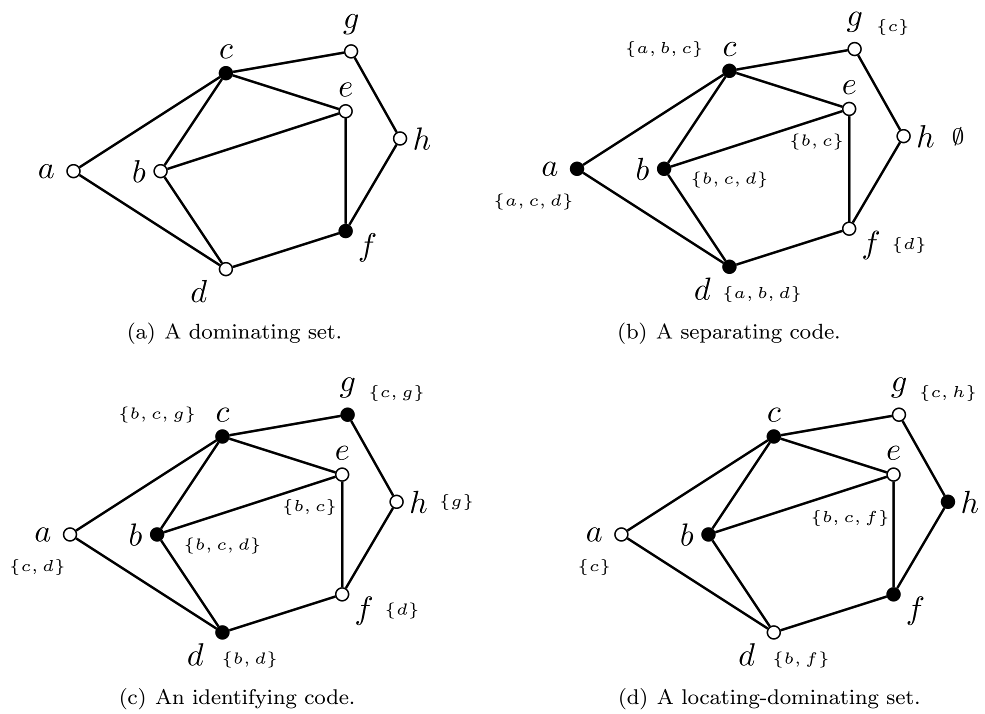



Figure 2: A graph with a dominating set, a separating set, an identifying code and a locating-dominating set (black vertices). The set next to each vertex $v$ in the three last figures indicates the intersection of $N[v]$ and the solution.

Whereas it is clear that any graph admits a (locating-)dominating set (its whole vertex set), a graph may not admit a separating code if it contains *twin* vertices, i.e. vertices having the same closed neighbourhood. In a graph containing no twins, the whole vertex set is a separating code (and therefore an identifying code); we call such graphs *twin-free*.

It can be observed that a dominating set, a separating code and an identifying code of a given graph $G$ are exactly a set cover, a test cover and a discriminating code, respectively, of the *closed neighbourhood hypergraph* of $G$, which is defined over $V(G)$ and whose edge set is the collection of closed neighbourhoods of the vertices of $G$.

Identifying codes, locating-dominating sets and further related notions have been studied extensively in the literature. We refer to [45] for an on-line bibliography on these topics, which lists more than 240 papers as of February 2013. In particular, see [2, 3, 6, 17, 28, 30, 31, 34, 43, 49, 54, 55, 56] for studies of the computational complexity of these problems. We remark that in many of these papers, due to the similarity between the two problems, the algorithmic properties of identifying codes and locating-dominating sets are studied together. For the same reason, we do so as well.

One of the interests of these notions lies in their applications to the location of threats in facilities [54, 58] and error detection in computer networks [40]. One can also mention applications to routing [44], to bioinformatics [37] (where identifying codes are called *differentiating dominating sets*) and to measuring the first-order logical complexity of graphs [41] (where locating-dominating sets are called *sieves*).

For definitions of computational complexity, we refer to the books [4, 33]. Let us formally define the decision and optimization problems (whose name is preceded by "Min") associated with identifying codes and locating-dominating sets:

Id Code
INSTANCE: A graph $G$ and an integer $k$.
QUESTION: Does $G$ have an identifying code of size at most $k$?

Min Id Code
INSTANCE: A graph $G$.
SOLUTION: An identifying code $\mathcal{C}$ of $G$.
MEASURE: $|\mathcal{C}|$.

Loc-Dom Set
INSTANCE: A graph $G$ and an integer $k$.
QUESTION: Does $G$ have a locating-dominating set of size at most $k$?

Min Loc-Dom Set
INSTANCE: A graph $G$.
SOLUTION: A locating-dominating set $\mathcal{D}$ of $G$.
MEASURE: $|\mathcal{D}|$.

Problems associated with set covers, test covers, discriminating codes, dominating sets and vertex covers are defined analogously, hence we skip their definitions.

When measuring the quality of an approximation algorithm for a minimization problem $P$ (with running time polynomial in the size of the instance), we consider its *performance ratio*, i.e. the guaranteed quantity $\alpha = \max\left\{\frac{|SOL_{I_P}|}{OPT(I_P)}\right\}$ over all instances $I_P$ of $P$ (where $SOL_{I_P}$ is a solution to $P$ given by the algorithm applied on instance $I_P$, and $OPT(I_P)$ denotes the size of an optimal solution of $P$ for $I_P$). We say that $P$ is $\alpha$-approximable and that the algorithm is an $\alpha$-approximation algorithm.

We recall that the class APX is the class of all optimization problems that are $c$-approximable for some constant $c$. Following terminology from e.g. [22], we also refer to the class log-APX as the class of all optimization problems that are $f(n)$-approximable, where $n$ is the size of the instance and $f$ is a poly-logarithmic function.

In this paper, we will study specific graph classes, of which many are standard, such as bipartite graphs, planar graphs or graphs of given maximum degree. Bipartite graphs which do not have any induced cycle of length more than 4 are called *chordal bipartite* (note that they are in general not chordal). Complements of bipartite graphs are called *co-bipartite* graphs. A class containing co-bipartite graphs is the one of asteroidal triple-free graphs (which we do not define here); in turn, *Dominating Shortest Path graphs*, introduced in [42], are those graphs admitting a dominating set whose vertices are the ones of a shortest path between two vertices of the graph; this class includes all asteroidal triple-free graphs. Interesting superclasses of co-bipartite graphs are *quasi-line graphs* (graphs for which each vertex neighbourhood can be partitioned into two cliques) and its superclass, the *claw-free graphs* (graphs having no $K_{1,3}$ as an induced subgraph). *Split graphs* are those whose vertex set can be partitioned into a clique and an independent set.

### *1.2. A few observations*

We remark a few facts that are useful in the studied context. We now show, in a series of easy observations and reductions, that test covers and discriminating codes are very close to each other.

**Observation 1.** *Let $H$ be a hypergraph admitting a set cover and a test cover. If $\mathcal{C}$ is a test cover of $H$, then there is an edge $X$ of $H$ such that $\mathcal{C} \cup \{X\}$ is a discriminating code of $H$.*

*Proof.* If $\mathcal{C}$ is not a discriminating code already, then $\mathcal{C}$ is not a set cover; that means at most one vertex of $H$, say $v$, is not dominated (if there were two, they would not be separated). Let $X$ be an edge containing $v$ (it exists since $H$ admits a set cover). Then $\mathcal{C} \cup \{X\}$ is a discriminating code. □

We remark that Observation 1 holds, of course, in the same way for separating codes and identifying codes as they are special cases of the corresponding hypergraph problems. We can use Observation 1 to define the following simple reductions. These reductions will show that problems Min Test Cover and Min Discriminating Code are computationally almost equivalent.

**Reduction 2** (Min Test Cover $\rightarrow$ Min Discriminating Code)**.** *Given a hypergraph $H = (I, A)$, we construct in polynomial time the hypergraph $H' = (I', A')$, where $I' = I \cup \{x\}$ and $A' = A \cup \{I'\}$.*

**Reduction 3** (Min Discriminating Code $\rightarrow$ Min Test Cover)**.** *Given a hypergraph $H = (I, A)$, we construct in polynomial time the hypergraph $H'' = (I'', A'')$, where $I'' = I \cup \{y\}$ and $A'' = A$.*

**Proposition 4.** *In Reduction 2, $H$ has a test cover of size $k$ if and only if $H'$ has a discriminating code of size $k+1$. In Reduction 3, $H$ has a discriminating code of size $k$ if and only if $H''$ has a test cover of size $k$.*

*Proof.* For the first claim, if $H$ has a test cover $\mathcal{T}$ of size $k$, then $\mathcal{T} \cup \{I'\}$ is a discriminating code of $H'$. If $H'$ has a discriminating code $\mathcal{T}'$ of size $k+1$, then it has to contain edge $I'$ as otherwise $x$ is not dominated. Then $\mathcal{T}' \setminus \{I'\}$ is a test cover of $H$ since $I'$ does not separate any pair of vertices.

For the second claim, it is clear that any discriminating code of $H$ is a test cover of $H''$, leaving only $y$ undominated. Now, in any test cover $\mathcal{T}$ of $H''$, $y$ must be undominated; hence all other vertices are dominated, and $\mathcal{T}$ is a discriminating code of $H$. □

Proposition 4 shows that the complexities of finding optimal discriminating codes and test covers are essentially the same (for general hypergraphs). In fact, similar reductions could also be done for more special cases such as identifying codes and separating codes in graphs.

We now show a way to relate (non-)approximability results of Min Id Code and Min Loc-Dom Set. The following theorem was given in [34].

**Theorem 5** ([34])**.** *Let $G$ be a graph having a locating-dominating set $\mathcal{D}$. If $G$ is twin-free, one can construct using $\mathcal{D}$ an identifying code $\mathcal{C}$ of $G$ with $|\mathcal{C}| \leq 2|\mathcal{D}|$.*

We observe that the reverse constructions are trivial, since any identifying code is a locating-dominating set. Hence Theorem 5 shows:

$$\frac{1}{2}OPT_{ID}(G) \leq OPT_{LD}(G) \leq OPT_{ID}(G) \leq 2OPT_{LD}(G)$$

Using Theorem 5 and these inequalities, we can link the complexity of approximating Min Loc-Dom Set and Min Id Code in the following two corollaries:

**Corollary 6.** *Any $\alpha$-approximation algorithm for* Min Loc-Dom Set *can be transformed into a $2\alpha$-approximation algorithm for* Min Id Code*, and vice-versa.*

**Corollary 7.** *If it is NP-hard to $\alpha$-approximate* Min Id Code*, then it is NP-hard to $\frac{\alpha}{2}$-approximate* Min Loc-Dom Set*. If it is NP-hard to $\alpha$-approximate* Min Loc-Dom Set*, then it is NP-hard to $\frac{\alpha}{2}$-approximate* Min Id Code*.*

We remark that, in the previous corollary, $\frac{\alpha}{2}$-approximation hardness only makes sense if $\alpha \geq 2$.

### *1.3. Related work*

It is well-known that Min Set Cover is an NP-hard problem [33], and that it is even log-APX-hard [52] (whereas logarithmic factors are tractable [39]); this even holds for the special case of Min Dominating Set [19, 35, 36]. The same properties hold for Min Test Cover [24] (and by Proposition 4, using Reduction 2 this result transfers to Min Discriminating Code) and Min Id Code (see [6, 43, 56], for different proofs). Min Discriminating Code was shown to be NP-hard, even when the bipartite incidence graph of the input hypergraph is planar [13].

Regarding the behaviour of the graph-theoretic problems of our interest when the instances are restricted to belong to specific graph classes, much is known for Dominating Set and Min Dominating Set: the

NP-completeness of Dominating Set holds for many classes of graphs such as (chordal) bipartite graphs, split graphs, line graphs or planar graphs but not for strongly chordal graphs, directed path graphs (which include the more famous interval graphs), or graphs having a dominating shortest path (see e.g. [25] for an on-line database, and [20, 35, 36] for surveys and summaries). The log-APX-completeness of Min Dominating Set is known to hold even for bipartite graphs and split graphs [19], however it does not hold for planar graphs or unit disk graphs (for which Min Dominating Set admits PTAS algorithms [5, 38, 50]) or in (bipartite) graphs of bounded maximum degree (at least 3), where it is APX-complete [19].

In comparison, much less is known about problems Id Code, Min Id Code, Loc-Dom Set and Min Loc-Dom Set; extending this knowledge is the main goal of this paper. It is known that, in general, Id Code and Loc-Dom Set are NP-complete [17, 15]. This result holds even for bipartite graphs [17], and for Id Code it holds for planar graphs of maximum degree 3 [2, 3], planar bipartite unit disk graphs [49], line graphs [30], split graphs [28, 31], and, interestingly, interval graphs [28, 31]. Regarding the minimization problems, log-APX-completeness of Min Id Code and Min Loc-Dom Set is known only for general graphs [6, 43, 56], and the two problems are APX-complete for graphs of bounded maximum degree at least 8 and 5, respectively [34].

### *1.4. Our contribution and structure of the paper*

We extend the knowledge about the computational complexity of Id Code, Loc-Dom Set, Min Id Code and Min Loc-Dom Set when these problems are restricted to specific classes of graphs. We compare these results to the corresponding ones for Dominating Set and Min Dominating Set; see Tables 1 and 2 for a summary of many known complexity results for these problems when instances are restricted to belong to some standard graph classes. These tables also indicate graph classes where the complexity of Min Id Code or Min Loc-Dom Set is unknown, giving rise to interesting open problems.

| graph class | Id Code | Loc-Dom Set | Dominating Set |
|---|---|---|---|
| bipartite | NP-c [17] | NP-c [17] | NP-c [7] |
| chordal bipartite | <u>NP-c</u> [Th. 31] | <u>NP-c</u> [Th. 33] | NP-c [48] |
| planar max. degree 3 | NP-c [3] | <u>NP-c</u> [Th. 29] | NP-c [33, 57] |
| planar bipartite max. degree 3 | <u>NP-c</u> [Th. 26] | <u>NP-c</u> [Th. 29] | NP-c [57] |
| (planar) line | NP-c [30] | OPEN | NP-c [59] |
| planar bipartite unit disk | NP-c [49] | NP-c [49] | NP-c [14] |
| bounded tree-width/clique-width | P [46] | P [46] | P [21, 16] |
| line of bounded tree-width | P [30] | P [30] | P [21] |
| split | NP-c [31] | <u>NP-c</u> [Co. 21] | NP-c [7] |
| undirected path | NP-c [31] | NP-c [31] | NP-c [9] |
| interval, directed path | NP-c [31] | NP-c [31] | P [9] |
| unit interval | OPEN | OPEN | P [9] |
| strongly chordal | NP-c [31] | NP-c [31] | P [26] |
| permutation | NP-c [31] | NP-c [31] | P [27] |
| bipartite permutation | OPEN | OPEN | P [27] |
| AT-free, DSP | NP-c [31] | <u>NP-c</u> [Co. 21] | P [42] |
| co-bipartite | <u>NP-c</u> [Co. 21] | <u>NP-c</u> [Co. 21] | P [42] |
| (planar) SC1 | <u>P</u> [Th. 35] | <u>P</u> [Th. 36] | <u>NP-c</u> [Th. 37] |
| (planar) SC2 | <u>NP-c</u> [Th. 40] | <u>P</u> [Th. 39] | <u>P</u> [Th. 39] |

Table 1: Comparison of complexities of decision problems Id Code, Loc-Dom Set and Dominating Set for selected graph classes. Underlined entries are new results proved in this paper. The abbreviations “P”, “NP-c”, “DSP” and “AT-free” stand for “polynomial-time solvable”, “NP-complete”, “Dominating Shortest Path” and “asteroidal triple-free”, respectively. SC1- and SC2-graphs will be defined in Section 4. Definitions of graph classes that are not defined in this paper can be found in [11, 25].

| graph class | Min Id Code LB | Min Id Code UB | Min Loc-Dom Set LB | Min Loc-Dom Set UB | Min Dominating Set LB | Min Dominating Set UB |
|---|---|---|---|---|---|---|
| in general | log-APX-h [6, 43, 56] | $O(\ln n)$ [24] | log-APX-h [56] | $O(\ln n)$ [34, 56] | log-APX-h [52] | $O(\ln n)$ [39] |
| bipartite | log-APX-h Co. 15 | $O(\ln n)$ [24] | log-APX-h Co. 15 | $O(\ln n)$ [34, 56] | log-APX-h [19, 52] | $O(\ln n)$ [39] |
| split, chordal | log-APX-h Co. 18 | $O(\ln n)$ [24] | log-APX-h Co. 18 | $O(\ln n)$ [34, 56] | log-APX-h [19, 52] | $O(\ln n)$ [39] |
| planar $(*)$ | NP-h [3] | 7 [55] | NP-h Th. 29 | 7 [55] | NP-h [33] | PTAS [5] |
| line $(*)$ | APX-h [28] | 4 [30] | OPEN | 8 [30]+Co. 6 | APX-h [18, 59] | 2 [59] |
| $K_{1,\ell}$-free $(\ell \geq 3)$ | log-APX-h Co. 21 | $O(\ln n)$ [24] | log-APX-h Co. 21 | $O(\ln n)$ [34, 56] | APX-h [18, 59] | $\ell - 1$ [19] |
| max. degree $\Delta$ | APX-h $\Delta \geq 8$: [34] | $O(\ln \Delta)$ [24] | APX-h $\Delta \geq 5$:[34] | $O(\ln \Delta)$ [34, 56] | APX-h $\Delta \geq 3$: [51] | $O(\ln \Delta)$ [39] |
| max. degree $\Delta \geq 3$ and bipartite | APX-h Th. 27 | $O(\ln \Delta)$ [24] | APX-h Th. 29 | $O(\ln \Delta)$ [34, 56] | APX-h [19] | $O(\ln \Delta)$ [39] |
| unit disk $(*)$ | NP-h [49] | $O(\ln n)$ [24] | NP-h [49] | $O(\ln n)$ [34, 56] | NP-h [14] | PTAS [38] |
| co-bipartite, AT-free, DSP | log-APX-h Co. 21 | $O(\ln n)$ [24] | log-APX-h Co. 21 | $O(\ln n)$ [34, 56] | P [42] | |
| strongly chordal $(*)$ | NP-h [31] | $O(\ln n)$ [24] | NP-h [31] | $O(\ln n)$ [34, 56] | P [26] | |
| undirected path $(*)$ | NP-h [31] | $O(\ln n)$ [24] | NP-h [31] | $O(\ln n)$ [34, 56] | NP-h [9] | $O(\ln n)$ [39] |
| directed path $(*)$ | NP-h [31] | $O(\ln n)$ [24] | NP-h [31] | $O(\ln n)$ [34, 56] | P [9] | |
| interval $(*)$ | NP-h [31] | 6 [10] | NP-h [31] | 12 [10]+Co. 6 | P [9] | |
| permutation $(*)$ | NP-h [31] | $O(\ln n)$ [24] | NP-h [31] | $O(\ln n)$ [34, 56] | P [27] | |

Table 2: Comparison of complexity lower bounds,"LB", and upper bounds on approximation ratios, "UB", (as functions of the order $n$ of the input graph) of optimization problems Min Id Code, Min Loc-Dom Set and Min Dominating Set for selected graph classes. Underlined entries are new results proved in this paper. Graph classes for which the precise complexity class of Min Id Code or Min Loc-Dom Set is not fully determined are marked with $(*)$ (note that for all graph classes except, up to our knowledge, undirected path graphs, the complexity class of Min Dominating Set is fully determined). The abbreviations "NP-h", "APX-h", "log-APX-h", "DSP" and "AT-free" stand for "NP-hard", "APX-hard", "log-APX-hard", "Dominating Shortest Path" and "asteroidal triple-free", respectively. Definitions of graph classes that are not defined in this paper can be found in [11, 25].

We will show in Section 2 that Min Id Code and Min Loc-Dom Set are log-APX-complete even for bipartite, split and co-bipartite graphs using approximation-preserving reductions from Min Discriminating Code. Prior, three different papers [6, 43, 56] showed that Min Id Code is log-APX-complete, but only in general graphs. Moreover, the proofs of [6, 43, 56] are relatively involved, while we use the intuitive proximity between Min Discriminating Code and Min Id Code to design much simpler reductions. Note that on co-bipartite graphs, Min Dominating Set is in fact trivially solvable in polynomial time; in contrast, our result shows that Min Id Code and Min Loc-Dom Set are computationally very hard on this class. As we will see, this result shows similar contrasts, with the complexity of Min Dominating Set for Dominating Shortest Path graphs and ($\ell$-)claw-free graphs.

In Section 3, we show that Min Id Code and Min Loc-Dom Set are APX-complete for bipartite graphs of maximum degree 3, answering a question from [34] and improving one of the results therein. Along the way, we obtain that the two problems are NP-hard for the same classes with the additional restriction of planarity (this improves three results from [2, 3, 49]), as well as for chordal bipartite graphs.

Finally, in Section 4, we exhibit two classes of graphs, which we call SC1- and SC2-graphs. For SC1-graphs, DOMINATING SET is NP-complete, but ID CODE and LOC-DOM SET are solvable in polynomial time; for SC2-graphs, DOMINATING SET and LOC-DOM SET are solvable in polynomial time, but ID CODE is NP-complete. These results are interesting because, until now, all known results for given graph classes were showing that ID CODE and LOC-DOM SET were at least as hard as DOMINATING SET, and that the complexities of ID CODE and LOC-DOM SET were the same.

## 2. Bipartite, co-bipartite and split graphs

In this section, we provide three similar reductions from MIN DISCRIMINATING CODE to MIN ID CODE for bipartite, split and co-bipartite graphs showing that MIN ID CODE is log-APX-complete in these three classes of graphs. Previously, only log-APX-completeness for general graphs was known. We begin with preliminary considerations that will be used in all three reductions.

### *2.1. Useful definitions, bounds and constructions*

In this section, we will use the framework of AP-reductions, introduced in [22] and which is now accepted as one of the most suitable kind of reductions for preserving approximability factors [4, Chapter 8.6].

**Definition 8** ([4, Definition 8.3]). *Let $P$ and $Q$ be two optimization problems. An* AP-reduction *from $P$ to $Q$ is a triple $(f, g, \alpha)$ where $f, g$ are functions and $\alpha$ is a positive constant, with the following properties:*

1. *Function $f$ maps any instance $I_P$ of $P$ together with any $c > 1$ to an instance $f(I_P, c)$ of $Q$.*
2. *For any instance $I_P$ of $P$, for any $c > 1$, and for any solution $SOL_{f(I_P,c)}$ of $f(I_P, c)$, function $g$ maps $(I_P, r, SOL_{f(I_P,c)})$ to a solution $g(f(I_P, c), SOL_{f(I_P,c)})$ of $I_P$.*
3. *For any instance $I_P$ of $P$, for any $c > 1$, if $I_P$ has a solution, then $f(I_P, c)$ has a solution.*
4. *For any fixed $c > 1$, $f(\cdot, c)$ and $g(\cdot, \cdot, c)$ are computable in polynomial time.*
5. *For every instance $I_P$ of $P$, for any $c > 1$,and for any solution $SOL_{f(I_P,c)}$ of $f(I_P, c)$, if:*

$$\max\left\{\frac{|SOL_{f(I_P,c)}|}{OPT_Q(f(I_P, c))}, \frac{OPT_Q(f(I_P, c))}{|SOL_{f(I_P,c)}|}\right\} \leq c, \text{ then:}$$

$$\max\left\{\frac{|g(f(I_P, c), SOL_{f(I_P,c)})|}{OPT_P(I_P)}, \frac{OPT_P(I_P)}{|g(f(I_P, c), SOL_{f(I_P,c)})|}\right\} \leq 1 + \alpha(c - 1).$$

We will use AP-reductions together with the following theorem:

**Theorem 9** ([22]). *Any optimization problem $P$ with instance $I_P$ that is log-APX-hard with respect to AP-reductions is NP-hard to approximate within a factor $c \ln(|I_P|)$, for some constant $c > 0$.*

We will also use the following bounds on the size of a minimum discriminating code:

**Theorem 10** ([12]). *Let $H = (I, A)$ be a hypergraph admitting a discriminating code, $\mathcal{C}$. Then $|\mathcal{C}| \geq \log_2(|I| + 1)$. If $\mathcal{C}$ is inclusion-wise minimal, then $|\mathcal{C}| \leq |I|$.*

We now describe two constructions, that ensure that the vertices of some vertex set $A$ are correctly identified using the vertices of another set $L$.

**Construction 11** (bipartite logarithmic identification of $A$ over $(A, L)$). *Given two sets of vertices $A$ and $L$ with $|L| \geq \lceil \log_2(|A| + 1) \rceil$, the bipartite logarithmic identification of $A$ over $(A, L)$, denoted $\mathcal{LOG}(A, L)$, is the graph of vertex set $A \cup L$ and where each vertex of $A$ has a distinct nonempty subset of $L$ as its neighbourhood.*

The next construction is similar, but makes sure that each vertex of $A$ has at least two neighbours in $L$.

**Construction 12** (non-singleton bipartite logarithmic identification of $A$ over $(A, L)$). *Given two sets of vertices $A$ and $L$ with $|A| \leq 2^{|L|} - |L| - 1$,*[1] *the non-single bipartite logarithmic identification of $A$ over $(A, L)$, denoted $\mathcal{LOG}^*(A, L)$, is the graph of vertex set $A \cup L$ and where each vertex of $A$ has a distinct subset of $L$ of size at least 2 as its neighbourhood.*

*2.2. Bipartite graphs*

We first give a reduction to Min Id Code for bipartite graphs.

**Reduction 13** (Min Discriminating Code $\to$ Min Id Code for bipartite graphs). *Given an instance $(I, A)$ of* Min Discriminating Code*, we construct in polynomial time the following bipartite graph $G(I, A)$ on $|I| + |A| + 9\lceil \log_2(|A| + 1)\rceil + 3$ vertices, with vertex set:*

$$V(G(I, A)) = I \cup A \cup \{x, y, z\} \cup \{a_j, b_j, c_j, d_j, e_j, f_j, g_j, h_j, i_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\},$$

*and edge set:*

$$\begin{aligned} E(G(I, A)) = \{x, y\} \cup \{y, z\} \cup \{\{z, i\} \mid i \in I\} \\ \cup\ E(\mathcal{B}(I, A)) \\ \cup\ E(\mathcal{LOG}(A, \{a_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\})) \\ \cup \{\{a_j, b_j\}, \{b_j, c_j\}, \{a_j, d_j\}, \{d_j, g_j\} \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\} \\ \cup \{\{d_j, e_j\}, \{e_j, f_j\}, \{g_j, h_j\}, \{h_j, i_j\} \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\}. \end{aligned}$$

*where $\mathcal{B}(I, A)$ denotes the bipartite incidence graph of $(I, A)$ and $E(\mathcal{LOG}(A, L))$ denotes the bipartite logarithmic identification of $A$ over $(A, L)$ (see Construction 11).*

*The construction is illustrated in Figure 3.*

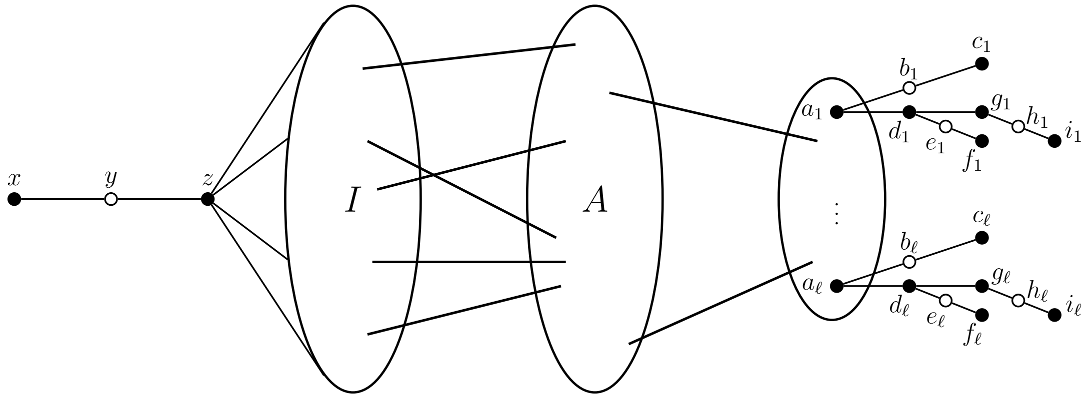



Figure 3: Reduction from Min Discriminating Code to Min Id Code (with $\ell = \lceil \log_2(|A| + 1)\rceil$). Black vertices belong to the identifying code $\mathcal{C}(\mathcal{D})$ defined in the proof of Theorem 14.

**Theorem 14.** *Let $(I, A)$ be an instance of* Min Discriminating Code*, and $G(I, A)$, the bipartite graph constructed using Reduction 13. Then, $(I, A)$ has a discriminating code of size at most $k$ if and only if $G(I, A)$ has an identifying code of size at most $k + 6\lceil \log_2(|A| + 1)\rceil + 2$, and one can construct one using the other in polynomial time.*

[1]There are exactly $2^{|L|} - |L| - 1$ distinct subsets of $L$ with size at least 2.

*Proof.* **Sufficient side (⇒)** Let $\mathcal{D} \subseteq A$ be a discriminating code of $(I, A)$, $|\mathcal{D}| = k$. We define $\mathcal{C}(\mathcal{D})$ as follows:

$$\mathcal{C}(\mathcal{D}) = \mathcal{D} \cup \{x, z\} \cup \{a_j, c_j, d_j, f_j, g_j, i_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1) \rceil\}.$$

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 6\lceil \log_2(|A| + 1)\rceil + 2$, and is clearly a dominating set. To see that it is an identifying code of $G(I, A)$, observe that vertex $z$ separates all vertices of $I$ from all vertices which are not in $I \cup \{z\}$. Vertex $z$ itself is the only vertex dominated only by $z$ (each vertex of $\mathcal{I}$ being dominating by some vertex of $\mathcal{D}$); $y$ is dominated by both $x, y$ and $x$, only by itself. Since $\mathcal{D}$ a discriminating code of $(I, A)$, all vertices of $I$ are dominated by a distinct subset of $\mathcal{D}$. Furthermore, due to the bipartite logarithmic identification of $A$ over $(A, \{a_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\})$ (and since each vertex $a_j$ belongs to the code), all vertices of $A$ are dominated by a unique subset of $\{a_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\}$. Finally, it is easy to check that all vertices of type $a_j, b_j, c_j, d_j, e_j, f_j, g_j, h_h, i_j$ are correctly separated.

**Necessary side (⇐)** Let $\mathcal{C}$ be an identifying code of $G(I, A)$, $|\mathcal{C}| = k + 6\lceil \log_2(|A| + 1)\rceil + 2$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $G(I, A)$, $|\mathcal{C}^*| \leq |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap \{V(G(I, A)) \setminus \{I \cup A\}\}| = 6\lceil \log_2(|A| + 1)\rceil + 2 \tag{1}$$

$$|\mathcal{C}^* \cap I| = \emptyset. \tag{2}$$

To get Condition (1), we replace $|\mathcal{C} \cap \{V(G(I, A)) \setminus \{I \cup A\}\}|$ by $\{x, z\} \cup \{a_j, c_j, d_j, f_j, g_j, i_j \mid 1 \leq j \leq \lceil \log_2(|A|+1)\rceil\}$ to get code $\mathcal{C}'$ (whose structure is similar to the one of the code constructed in the (⇒) part of the proof). Observe that $|\mathcal{C}'| \leq |\mathcal{C}|$. Indeed, we already had $|\mathcal{C} \cap \{V(G(I, A)) \setminus \{I \cup A\}\}| \geq 6\lceil \log_2(|A|+1)\rceil + 2$. To see this, note that vertex $z$ is the only one separating $\{x, y\}$, and $|\mathcal{C} \cap \{x, y\}| \geq 1$ since $\mathcal{C}$ must dominate $x$. Similarly, for any $j \in \{1, \ldots, \log_2(|A| + 1)\}$, vertices $a_j, d_j, g_j$ are the only ones separating $\{b_j, c_j\}$, $\{e_j, f_j\}$ and $\{h_j, i_j\}$, respectively, and $|\mathcal{C} \cap \{b_j, c_j\}| \geq 1$, $|\mathcal{C} \cap \{e_j, f_j\}| \geq 1$ and $|\mathcal{C} \cap \{h_j, i_j\}| \geq 1$, since $\mathcal{C}$ must dominate $c_j$, $f_j$ and $i_j$, respectively.

To fulfill Condition (2), we replace each vertex $i \in I \cap \mathcal{C}'$ by some vertex in $A$. If $\mathcal{C}' \setminus \{i\}$ is an identifying code, we may just remove $i$ from the code. Otherwise, note that $i$ is not needed for domination since all vertices of $I$ are dominated by $z$ and all vertices of $A$ are dominated by some vertex in $\{a_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\}$. Hence, $i$ separates $i$ itself from some other vertex $i'$ in $I$ (indeed, one can check that all other types of pairs which could be separated by $i$ are actually already separated by some vertex of $\mathcal{C}' \cap (V(G(I, A)) \setminus I)$. But then, the pair $\{i, i'\}$ is unique (suppose $i$ separates $i$ itself from two distinct vertices $i'$ and $i''$ of $I$, then $i'$ and $i''$ would not be separated by $\mathcal{C}'$, a contradiction). Since $(I, A)$ admits a discriminating code, there must be some vertex $a$ of $A$ separating $i$ from some $i'$. Hence we replace $i$ by $a$. Doing this for every $i \in \mathcal{C}' \cap I$, we get code $\mathcal{C}^*$, and $|\mathcal{C}^*| \leq |\mathcal{C}'| \leq |\mathcal{C}|$.

Using the previous observations and by similar arguments as in in the (⇒) part of the proof, one can easily check that after these two modifications performed on code $\mathcal{C}$, the obtained code $\mathcal{C}^*$ is still an identifying code.

By Condition (2), we have $|\mathcal{C}^* \cap A| \leq |\mathcal{C}| - 6\lceil \log_2(|A| + 1)\rceil + 2 = k$.

To finish the proof, we claim that $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. This is easy to observe, as all pairs $\{I, I'\}$ of $I$ are separated by $\mathcal{C}^*$. By Condition (1), they must be separated by some vertex of $A$ (note that $z$ is adjacent to all vertices of $I$). Hence $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. □

Theorem 14 proves that Id Code for bipartite graphs is NP-hard. In fact, Reduction 13 also preserves approximation ratios up to a constant factor, as shown by the following corollary.

**Corollary 15.** *Reduction 13 is an AP-reduction with parameter $\alpha = 8$ and* Min Id Code, Min Loc-Dom Set *for bipartite graphs are log-APX-complete.*

*Proof.* We will use Theorem 14 to show that any $c$-approximation algorithm $\mathscr{A}$ for Min Id Code for bipartite graphs can be transformed into a $7c$-approximation algorithm for Min Discriminating Code, and $7c \leq 1 + c(8 - 1)$; therefore, by Definition 8, we have an AP-reduction with $\alpha = 8$. Since Min

Discriminating Code is log-APX-complete by [24] together with Proposition 4 and Min Id Code is known to be in log-APX, we get the claim for Min Id Code. Corollary 7 immediately implies the claims for Min Loc-Dom Set.

Let $(I, A)$ be an instance of Min Discriminating Code with optimal value OPT, and let $G(I, A)$ be the bipartite graph constructed using Reduction 13. By Theorem 14, we have:

$$\gamma^{\text{ID}}(G(I, A)) \leq OPT + 6\lceil \log_2(|A| + 1)\rceil + 2. \tag{3}$$

Let $\mathcal{C}$ be an identifying code of $G(I, A)$ computed by $\mathscr{A}$. We have:

$$|\mathcal{C}| \leq c\gamma^{\text{ID}}(G(I, A)). \tag{4}$$

By Theorem 14, we can compute in polynomial time a discriminating code $\mathcal{D}$ of $(I, A)$. Using Inequalities 3 and 4 together with the fact that $\lceil \log_2(|A|)\rceil \leq OPT \leq |\mathcal{D}|$ (Theorem 10), we get:[2]

$$\begin{aligned}
|\mathcal{D}| &\leq |\mathcal{C}| - 6\lceil \log_2(|A| + 1)\rceil - 2 \\
&\leq c\gamma^{\text{ID}}(G(I, A)) - 6\lceil \log_2(|A| + 1)\rceil - 2 \\
&\leq c(OPT + 6\lceil \log_2(|A| + 1)\rceil + 2) - 6\lceil \log_2(|A| + 1)\rceil - 2 \\
&\leq cOPT + (c - 1)(6\lceil \log_2(|A| + 1)\rceil + 2) \\
&\leq cOPT + (c - 1)(6\lceil \log_2(|A|)\rceil + 8) \\
&\leq cOPT + (c - 1)(6OPT + 8) \\
&\leq (7c - 6)OPT + 8 \\
&\leq 7cOPT.
\end{aligned}$$

□

*2.3. Split graphs*

In this section, we use a reduction from Min Discriminating Code to Min Id Code for split graphs similar to Reduction 13.

**Reduction 16** (Min Discriminating Code → Min Id Code for split graphs)**.** *Given an instance $(I, A)$ of* Min Discriminating Code*, we construct in polynomial time the following split graph $Sp(I, A)$ on $|I| + |A| + 6\lceil \log_2(|A| + 1)\rceil + 1$ vertices, with vertex set $V(Sp(I, A)) = K \cup S$ ($K$ is a clique and $S$, an independent set). More specifically:*

$$K = I \cup \{u\} \cup \{k_j \mid 1 \leq j \leq 2\lceil \log_2(|A| + 1)\rceil\}$$
$$S = A \cup \{v\} \cup \{s_j, t_j \mid 1 \leq j \leq 2\lceil \log_2(|A| + 1)\rceil\}.$$

*$Sp(I, A)$ has edge set:*

$$\begin{aligned}
E(Sp(I, A)) = \{u, v\} & \\
\cup\ & E(\mathcal{B}(I, A)) \\
\cup\ & E(\mathcal{LOG}^*(A, \{k_j \mid 1 \leq j \leq 2\lceil \log_2(|A| + 1)\rceil\})) \\
\cup\ & \{\{k_j, s_j\}, \{k_j, t_j\} \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\} \\
\cup\ & \{a, b \mid a, b \in K, a \neq b\},
\end{aligned}$$

*where $\mathcal{B}(I, A)$ denotes the bipartite incidence graph of $(I, A)$ and $E(\mathcal{LOG}^*(A, L))$ denotes the non-singleton bipartite logarithmic identification of $A$ over $(A, L)$ (see Construction 12).*

*The construction is illustrated in Figure 4.*

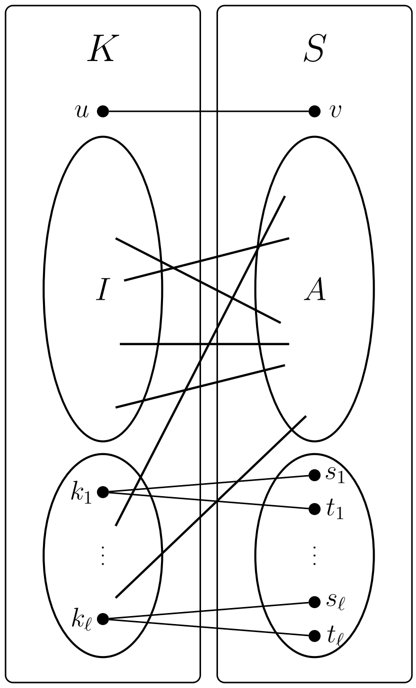



Figure 4: Reduction from MIN DISCRIMINATING CODE to MIN ID CODE (with $\ell = 2\lceil \log_2(|A|+1) \rceil$).

**Theorem 17.** *Let $(I, A)$ be an instance of* MIN DISCRIMINATING CODE*, and $Sp(I, A)$, the split graph constructed using Reduction 16. Then, $(I, A)$ has a discriminating code of size at most $k$ if and only if $Sp(I, A)$ has an identifying code of size at most $k + 4\lceil \log_2(|A|+1) \rceil + 1$, and one can construct one using the other in polynomial time.*

The proof of Theorem 17 being very similar to the one of Theorem 14, we delay it to the appendix. As Theorem 14 for bipartite graphs, Theorem 17 shows that ID CODE for split graphs is NP-hard; moreover, Reduction 16 preserves approximation ratios up to a constant factor (the proof being the same as the one of Corollary 15, we omit it).

**Corollary 18.** *Reduction 16 is an AP-reduction with parameter $\alpha = 6$ and* MIN ID CODE*,* MIN LOC-DOM SET *for split graphs are log-APX-complete.*

*2.4. Co-bipartite graphs*

We now prove that MIN ID CODE is log-APX-complete even for co-bipartite graphs, that is, graphs whose vertex set can be partitioned into two cliques. Note that this class (when assumed to be connected) is a subclass of Dominating Shortest Path graphs (a class containing the more well-known asteroidal triple-free graphs) since any pair of adjacent vertices belonging each to a distinct one among the two cliques, forms a dominating shortest path. This is particularly interesting since DOMINATING SET is solvable in polynomial time for Dominating Shortest Path graphs [42]. Any co-bipartite graph is also trivially quasi-line, and, in turn, claw-free. Hence our result contrasts again with the complexity of MIN DOMINATING SET, which is approximable within a factor of $\ell - 1$ for $\ell$-claw-free graph[3] for any fixed $\ell$, as shown in [19] using a short argument.

[2]For the last line inequality, we assume here that $OPT \geq 2$.
[3]A graph is $\ell$-claw-free if it has no $K_{1,\ell}$ as an induced subgraph — hence 3-claw-free means claw-free.

**Reduction 19** (MIN DISCRIMINATING CODE → MIN ID CODE for co-bipartite graphs)**.** *Given an instance $(I, A)$ of* MIN DISCRIMINATING CODE*, we construct in polynomial time the following co-bipartite graph $G(I, A)$ on $|I| + |A| + 6\lceil \log_2(|A| + 1)\rceil$ vertices, with vertex set $V(G(I, A)) = K^1 \cup K^2$, where $K^1$ and $K^2$ are two cliques over the following sets of vertices:*

$$K^1 = I \cup \{a_j, b_j, c_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\}$$
$$K^2 = A \cup \{d_j, e_j, f_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\}.$$

*$G(I, A)$ has edge set:*

$$\begin{aligned} E(G(I, A)) = E(\mathcal{B}(I, A)) \\ &\cup\ E(\mathcal{LOG}(A, \{a_j \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\})) \\ &\cup \{\{a_j, d_j\}, \{b_j, d_j\}, \{b_j, e_j\}, \{b_j, f_j\}, \{c_j, f_j\} \mid 1 \leq j \leq \lceil \log_2(|A| + 1)\rceil\} \\ &\cup \{x, y \mid x, y \in K^1\} \cup \{x, y \mid x, y \in K^2\}. \end{aligned}$$

*where $\mathcal{B}(I, A)$ denotes the bipartite incidence graph of $(I, A)$ and $E(\mathcal{LOG}(A, L))$ denotes the bipartite logarithmic identification of $A$ over $(A, L)$ (see Construction 11).*

*The construction is illustrated in Figure 5.*

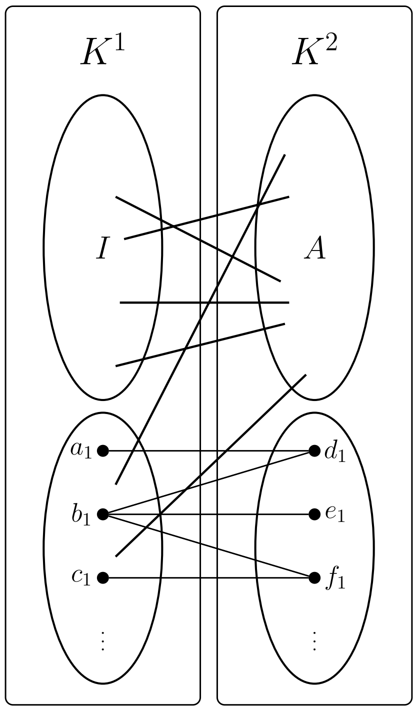



Figure 5: Reduction from MIN DISCRIMINATING CODE to MIN ID CODE (with $\ell = \lceil \log_2(|A| + 1)\rceil$).

**Theorem 20.** *Let $(I, A)$ be an instance of* MIN DISCRIMINATING CODE*, and $G(I, A)$, the bipartite graph constructed using Reduction 19. Then, $(I, A)$ has a discriminating code of size at most $k$ if and only if $G(I, A)$ has an identifying code of size at most $k + 5\lceil \log_2(|A| + 1)\rceil - 2$, and one can construct one using the other in polynomial time.*

Once again, the proof of Theorem 20 being very similar to the one of Theorem 14, we delay it to the appendix. Again, we can show that Reduction 19 also preserves approximation ratios up to a constant factor (the proof being the same as the one of Corollaries 15 and 18, we omit it).

**Corollary 21.** *Reduction 19 is an AP-reduction with parameter $\alpha = 7$ and* Min Id Code*,* Min Loc-Dom Set *for co-bipartite graphs (and hence for quasi-line graphs, asteroidal triple-free graphs and Dominating Shortest Path graphs) are* log-APX*-complete.*

## 3. Reductions for (planar) bipartite graphs of bounded maximum degree and chordal bipartite graphs

In this section, we improve the NP-completeness results for various subclasses of planar graphs from [2], [3] and [49] by showing that Id Code and Loc-Dom Set are NP-complete for planar bipartite graphs of maximum degree 3. We also improve and extend the APX-hardness results for Min Id Code and Min Loc-Dom Set for non-bipartite graphs of maximum degree at least 8 and 5, respectively, from [34] by showing that they are APX-hard even for bipartite graphs of maximum degree 3 (the authors of [34] asked whether their result could be extended to bipartite graphs, hence our result answers their question in positive). Finally, we show that Id Code and Loc-Dom Set are NP-complete for chordal bipartite graphs. Note that the class of chordal bipartite graphs is interesting for the following reason: Dominating Set is NP-complete for this class [48], but the related problem Total Dominating Set is polynomial-time solvable [23].

### *3.1. Definition of L-reductions*

The following type of reductions, called *L-reductions* (for "linear reductions") was introduced in [51]; they are now widely used to prove APX-hardness of optimization problems.

**Definition 22** ([51])**.** *Let $P$ and $Q$ be two optimization problems. An* L-reduction *from $P$ to $Q$ is a four-tuple $(f, g, \alpha, \beta)$ where $f$ and $g$ are polynomial time computable functions and $\alpha, \beta$ are positive constants with the following properties:*

1. *Function $f$ maps instances of $P$ to instances of $Q$ and for every instance $I_P$ of $P$:*

$$OPT_Q(f(I_P)) \leq \alpha \cdot OPT_P(I_P).$$

2. *For every instance $I_P$ of $P$ and every solution $SOL_{f(I_P)}$ of $f(I_P)$, $g$ maps the pair $(f(I_P), SOL_{f(I_P)})$ to a solution $SOL_{I_P}$ of $I_P$ such that:*

$$|OPT_P(I_P) - |SOL_{I_P}|| \leq \beta \cdot |OPT_Q(f(I_P)) - |SOL_{f(I_P)}||.$$

L-reductions are useful due to the following fact:

**Theorem 23** ([51])**.** *Let $P$ and $Q$ be two optimization problems. If there exists an L-reduction from $P$ to $Q$ with parameters $\alpha$ and $\beta$ and $Q$ has a $(1 + \epsilon)$-approximation algorithm for some $\epsilon > 0$, then $P$ has a $(1 + \alpha\beta\epsilon)$-approximation algorithm.*

This can be used to derive hardness results in the following way:

**Corollary 24** ([51])**.** *Let $P$ and $Q$ be two optimization problems. If there exists an L-reduction from $P$ to $Q$ with parameters $\alpha$ and $\beta$ and it is* NP*-hard to approximate $P$ within ratio $r_P = 1 + \delta$, then it is* NP*-hard to approximate $Q$ within ratio $r_Q = 1 + \frac{\delta}{\alpha\beta}$.*

### *3.2. Reductions from* Min Vertex Cover

We first present a reduction from Min Vertex Cover to Min Id Code. It consists in a very simple edge-gadget.

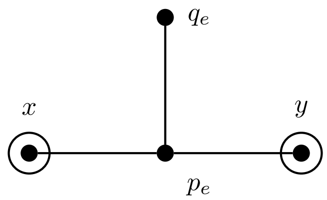



Figure 6: Reduction gadget for edge $e = \{x, y\}$ in Reduction 25 from Min Vertex Cover to Min Id Code. The original vertices of $G$, $x$ and $y$, are circled.

**Reduction 25** (Min Vertex Cover $\rightarrow$ Min Id Code)**.** *Given a graph $G$, we construct the graph $G'$ on vertex set*

$$V(G') = V(G) \cup \{p_e, q_e \mid e \in E(G)\},$$

*and edge set*

$$E(G') = \{\{x, p_e\}, \{y, p_e\}, \{p_e, q_e\} \mid e = \{x, y\} \in E(G)\}.$$

*The construction is illustrated in Figure 6.*

For the following claims, let $G$ be a connnected cubic graph and $G'$, the graph obtained from $G$ using Reduction 25.

**Claim A.** *Let $\mathcal{N}$ be a vertex cover of $G$. Using $\mathcal{N}$, one can build an identifying code of $G'$ of size at most $|\mathcal{N}| + |E(G)|$.*

*Proof.* Let $\mathcal{C} = \mathcal{N} \cup \{p_e \mid e \in E(G)\}$. We can easily check that $\mathcal{C}$ is an identifying code of $G'$: any original vertex $x$ of $G$ is dominated by the unique set of vertices $\{p_e \mid x \in e, e \in E(G)\}$ (this set having at least two elements by the first paragraph of the proof). For each edge $\{x, y\} = e \in E(G)$, vertex $p_e$ is dominated by itself and at least one of $x, y$; $q_e$ is dominated by $p_e$ only. $\square$

**Claim B.** *Let $\mathcal{C}$ be an identifying code of $G'$. One can use $\mathcal{C}$ to build a vertex cover of $G$ of size at most $|\mathcal{C}| - |E(G)|$.*

*Proof.* We observe that for each edge $e = \{x, y\}$ of $G$, one of $p_e, q_e$ belongs to $\mathcal{C}$, since $\mathcal{C}$ has to dominate $q_e$. Moreover, one of $x, y$ belongs to $\mathcal{C}$ since $p_e, q_e$ need to be separated by $\mathcal{C}$. Hence, the restriction of the code to the original vertices of $G$, $\mathcal{C} \cap V(G)$, is a vertex cover of $G$ with size at most $|\mathcal{C}| - |E(G)|$. $\square$

In what follows, let $\tau(G)$ denote the minimum size of a vertex cover of $G$. The previous claims are enough to give a new proof that Id Code is NP-complete:

**Theorem 26.** Id Code *is NP-complete, even for planar bipartite graphs of maximum degree 3.*

*Proof.* We apply Reduction 25 to Vertex Cover for planar cubic graphs, which is known to be NP-complete [1, 32].[4] Given a planar cubic graph $G$, it is easy to check that $G'$ is planar, has maximum degree 3, and is bipartite, since the edge gadget for edge $e = \{x, y\}$ is bipartite, with $x, y$ in the same part. Now, Claims A and B applied to an optimal vertex cover and an optimal identifying code show that $\gamma^{\text{ID}}(G') = \tau(G) + |E(G)|$, completing the proof. $\square$

In fact, we can show that Reduction 25 applied to Min Vertex Cover for graphs of maximum degree 3 is an L-reduction.

[4] The reduction in [32] is for planar *subcubic* graphs, but one can make the constructions cubic using the gadgets for vertices of degree less than 3 given in [1].

**Theorem 27.** *Reduction 25 applied to graphs of maximum degree 3 is an L-reduction with parameters $\alpha = 4$ and $\beta = 1$. Therefore,* Min Id Code *is* APX*-complete, even for bipartite graphs of maximum degree 3.*

*Proof.* Let $G$ be a graph of maximum degree 3 and $G'$ the graph constructed from $G$ using Reduction 25. We have to prove Properties 1 and 2 from Definition 22.

First of all, by Claim A, given an optimal vertex cover $\mathcal{N}^*$ of $G$, we can construct an identifying code $\mathcal{C}$ with $\gamma^{\text{ID}}(G') \le |\mathcal{C}| \le |\mathcal{N}^*|+|E(G)| = \tau(G)+|E(G)|$. Similarly, by Claim B, given an optimal identifying code $\mathcal{C}^*$ of $G'$, we can construct a vertex cover $\mathcal{N}$ of $G$ such that $\tau(G) \le |\mathcal{N}| \le |\mathcal{C}^*| - |E(G)| = \gamma^{\text{ID}}(G) - |E(G)|$. Hence we have:

$$\gamma^{\text{ID}}(G') = \tau(G) + |E(G)|. \tag{5}$$

**Property 1.** Since $G$ has maximum degree 3, each vertex can cover at most three edges, hence we have $\tau(G) \ge \frac{|E(G)|}{3}$, so $|E(G)| \le 3\tau(G)$. Using Equality (5), we get:

$$\gamma^{\text{ID}}(G') = \tau(G) + |E(G)| \le 4\tau(G),$$

which proves Property 1 of Definition 22.

**Property 2.** Let $\mathcal{C}$ be an identifying code of $G'$. Using Claim B applied to $\mathcal{C}$, we obtain a vertex cover $\mathcal{N}$ with $|\mathcal{N}| \le |\mathcal{C}| - |E(G)|$. By Equality (5), we have $-\tau(G) = |E(G)| - \gamma^{\text{ID}}(G')$. So we obtain:

$$\begin{aligned} |\mathcal{N}| - \tau(G) \le& |\mathcal{C}| - |E(G)| + |E(G)| - \gamma^{\text{ID}}(G') \\ |\tau(G) - |\mathcal{N}|| \le& |\gamma^{\text{ID}}(G') - |\mathcal{C}||, \end{aligned}$$

which proves Property 2 of Definition 22.

For the second part of the statement, note that Min Vertex Cover is known to be APX-complete, even for graphs of maximum degree 3 [1, 19]. By construction and as observed in the proof of Theorem 26, the graphs built from graphs with maximum degree 3 in Reduction 25 are bipartite and of maximum degree 3. □

A similar, slightly more involved, reduction from Min Vertex Cover to Min Loc-Dom Set is as follows:

**Reduction 28** (Min Vertex Cover → Min Loc-Dom Set)**.** *Given a graph $G$, we construct the graph $G'$ on vertex set*

$$V(G') = V(G) \cup \{q_e, r_e, s_e, t_e, u_e \mid e \in E(G)\},$$

*and edge set*

$$\begin{aligned} E(G') =& \{\{x, r_e\}, \{y, s_e\}, \{r_e, t_e\}, \{t_e, s_e\}, \{u_e, r_e\}, \{u_e, s_e\}, \\ & \{q_e, t_e\}, \{q_e, u_e\} \mid e = \{x, y\} \in E(G)\}. \end{aligned}$$

*The construction is illustrated in Figure 7.*

For the following claims, let $G$ be a graph and $G'$, the graph obtained from $G$ using Reduction 28.

**Claim C.** *Let $\mathcal{N}$ be a vertex cover of $G$. Using $\mathcal{N}$, one can build a locating-dominating set of $G'$ of size at most $|\mathcal{N}| + 2|E(G)|$.*

*Proof.* Once again, we assume that $G$ has minimum degree 2.

First, let $\mathcal{D} = \mathcal{N}$. Then, for each edge $e = \{x, y\} \in E(G)$, if $x \in \mathcal{N}$, put vertices $s_e, t_e$ into $\mathcal{D}$. Otherwise, put vertices $r_e, t_e$ into $\mathcal{D}$.

We can check that $\mathcal{D}$ is a locating-dominating set of $G'$. Recall that we need separation only for vertices in $V(G') \setminus \mathcal{D}$. If an original vertex $x$ of $G$ does not belong to $\mathcal{N}$, all its neighbours in $G$ belong to $\mathcal{N}$; hence $x$ is dominated by two or three vertices from $\mathcal{D}$ of the form $s_e$, hence $x$ is separated from every other vertex. Moreover, for each edge $e$ of $G$, vertices $q_e, r_e, s_e, u_e$ are separated from all vertices of $V(G') \setminus \{q_e, r_e, s_e, t_e, u_e\}$ by either $r_e, s_e$ or $t_e$; finally, it is easy to check that they are correctly separated from each other. □

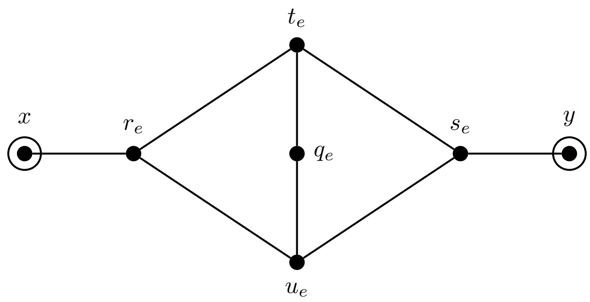



Figure 7: Reduction gadget for edge $e = \{x, y\}$ in Reduction 28 from MIN VERTEX COVER to MIN LOC-DOM SET. The original vertices of $G$, $x$ and $y$, are circled.

**Claim D.** *Let $\mathcal{D}$ be a locating-dominating set of $G'$. For each $e \in E(G)$, we have:*

$$|\mathcal{D} \cap \{q_e, r_e, s_e, t_e, u_e\}| \geq 2.$$

*Proof.* Note that $N(t_e) = N(u_e)$, hence one of them (say $t_e$) belongs to $\mathcal{D}$. Now, $u_e$ needs to be dominated, hence one of $q_e, r_e, s_e, u_e$, belong to $\mathcal{D}$. □

**Claim E.** *Let $\mathcal{D}$ be a locating-dominating set of $G'$. For each $e = \{x, y\} \in E(G)$, we have:*

$$|\mathcal{D} \cap \{x, y, q_e, r_e, s_e, t_e, u_e\}| \geq 3.$$

*Proof.* By contradiction, suppose $|\mathcal{D} \cap \{x, y, q_e, r_e, s_e, t_e, u_e\}| = 2$. By the same argument as in the proof of Claim D, we can assume $t_e \in \mathcal{D}$, and $|\mathcal{D} \cap \{q_e, r_e, s_e, u_e\}| = 1$. We derive a contradiction for each case: if $q_e$ or $u_e$ belong to $\mathcal{D}$, $r_e, s_e$ are not separated. If $r_e \in \mathcal{D}$, $q_e, s_e$ are not separated. If $s_e \in \mathcal{D}$, $q_e, r_e$ are not separated. □

**Claim F.** *Let $\mathcal{D}$ be a locating-dominating set of $G'$. From $\mathcal{D}$, we can build a locating-dominating set $\mathcal{D}'$ with $|\mathcal{D}'| \leq |\mathcal{D}|$ such that for each $e = \{x, y\} \in E(G)$, we have $|\mathcal{D}' \cap \{q_e, r_e, s_e, t_e, u_e\}| = 2$.*

*Proof.* Let $e = \{x, y\} \in E(G)$. Assume that $|\mathcal{D} \cap \{q_e, r_e, s_e, t_e, u_e\}| \geq 3$. By the same argument as in the previous proofs, we may also assume that $t_e \in \mathcal{D}$. If neither $r_e$ nor $s_e$ belongs to the solution, then $\mathcal{D} \cap \{q_e, r_e, s_e, t_e, u_e\} = \{q_e, t_e, u_e\}$. Then, observe that $\mathcal{D}' := \mathcal{D} \setminus \{u_e\}$ is still a locating-dominating set, as $u_e$ is now the only vertex out of the solution dominated only by $q_e$, and $u_e$ was not involved in the separation of $r_e, s_e$, being their common neighbour. Now, either $|\mathcal{D}' \cap \{q_e, r_e, s_e, t_e, u_e\}| = 2$ and we are done, or one of $r_e, s_e$ belongs to $\mathcal{D}'$ (say, $r_e \in \mathcal{D}'$, the other case following by symmetry). We let $\mathcal{D}'' := (\mathcal{D}' \setminus \{q_e, s_e, u_e\}) \cup \{y\}$. Now, vertices $q_e, s_e, u_e$ are separated; the only problem could come from the separation of $x$ and $u_e$, however, since we repeat the operation for each original edge $e$ of $G$, in the end either $x$ belongs to $\mathcal{D}''$, or all its neighbours (at least two) do, and $x, u_e$ are separated. □

**Claim G.** *Let $\mathcal{D}$ be a locating-dominating set of $G'$. One can use $\mathcal{D}$ to build a vertex cover of $G$ of size at most $|\mathcal{D}| - 2|E(G)|$.*

*Proof.* Use Claim F to build code $\mathcal{D}''$ such that $|\mathcal{D}''| \leq |\mathcal{D}|$ and for each $e = \{x, y\} \in E(G)$, we have $|\mathcal{D}'' \cap \{q_e, r_e, s_e, t_e, u_e\}| = 2$. By this property and Claim E, we have $|\mathcal{D}'' \cap \{x, y\}| \geq 1$. Hence $\mathcal{N} = \mathcal{D}'' \setminus \{q_e, r_e, s_e, t_e, u_e \mid e \in E(G)\}$ is a vertex cover of $G$ with $|\mathcal{N}| \leq |\mathcal{D}''| - 2|E(G)| \leq |\mathcal{D}| - 2|E(G)|$. □

We are now ready to prove the following:

**Theorem 29.** *Reduction 28 applied to graphs of maximum degree 3 is an L-reduction with parameters $\alpha = 7$ and $\beta = 1$. Therefore,* MIN LOC-DOM SET *is APX-complete, even for bipartite graphs of maximum degree 3, and* LOC-DOM SET *is NP-complete, even for planar bipartite graphs of maximum degree 3.*

*Proof.* Once again, applying Reduction 28 to Vertex Cover for planar graphs of maximum degree 3, Claims C and G show that:

$$\gamma^{\text{LD}}(G') = \tau(G) + 2|E(G)|, \tag{6}$$

proving the NP-completeness part of the statement.

For showing that we have an L-reduction, let $G$ be a graph of maximum degree 3 (and minimum degree 2) and $G'$ the graph constructed from $G$ using Reduction 28. We have to prove Properties 1 and 2 from Definition 22.

**Property 1.** Again, we have $\tau(G) \geq \frac{|E(G)|}{3}$, so $|E(G)| \leq 3\tau(G)$. Using Equality (6), we get:

$$\gamma^{\text{LD}}(G') = \tau(G) + 2|E(G)| \leq 7\tau(G),$$

which proves Property 1 of Definition 22.

**Property 2.** Let $\mathcal{D}$ be a locating-dominating set of $G'$. Using Claim G applied to $\mathcal{D}$, we obtain a vertex cover $\mathcal{N}$ with $|\mathcal{N}| \leq |\mathcal{D}| - 2|E(G)|$. By Equality (6), we have $-\tau(G) = 2|E(G)| - \gamma^{\text{LD}}(G')$. So we obtain:

$$\begin{aligned} |\mathcal{N}| - \tau(G) \leq& |\mathcal{D}| - 2|E(G)| + 2|E(G)| - \gamma^{\text{LD}}(G') \\ |\tau(G) - |\mathcal{N}|| \leq& |\gamma^{\text{LD}}(G') - |\mathcal{D}||, \end{aligned}$$

which proves Property 2 of Definition 22. □

*3.3. Reductions from* Min Dominating Set

We now give another reduction, this times from Min Dominating Set to Min Id Code.

**Reduction 30** (Min Dominating Set → Min Id Code)**.** *Given a graph $G$, we construct the graph $G'$ on vertex set*

$$V(G') = V(G) \cup \{a_x, b_x, c_x, d_x, e_x \mid x \in V(G)\},$$

*and edge set*

$$E(G') = E(G) \cup \{\{x, a_x\}, \{x, e_x\}, \{a_x, b_x\}, \{a_x, c_x\}, \{a_x, d_x\}, \{e_x, b_x\}, \{e_x, c_x\}, \{e_x, d_x\} \mid x \in V(G)\}.$$

*The construction is illustrated in Figure 8.*

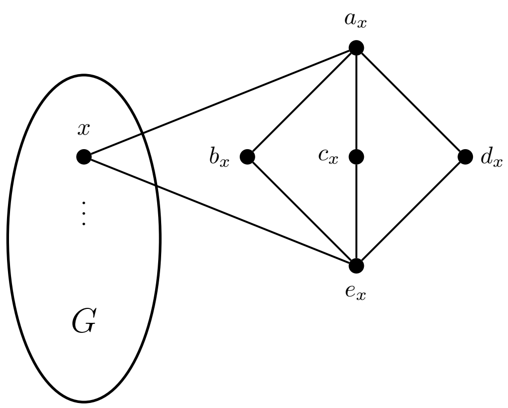



Figure 8: Reduction from Min Dominating Set to Min Id Code.

Even though Reduction 30 can be shown to be an L-reduction when applied to Min Dominating Set for graphs of bounded maximum degree, we use it only as a classical reduction between decision problems, as the reduction would not strengthen our previous non-approximability results.

**Theorem 31.** Id Code *is NP-complete, even for chordal bipartite graphs.*

*Proof.* We apply Reduction 30 to the class of chordal bipartite graphs, for which Dominating Set is known to be NP-complete [48]. Given a chordal bipartite graph $G$, it is easy to check that the parts added to $G$ to construct $G'$ do not add any induced cycle of length more than 4. We now show that $G$ has a dominating set of size at most $k$ if and only if $G'$ has an identifying code of size at most $k + 3|V(G)|$.

**Sufficient side (⇒)** Let $\mathcal{D}$ be a dominating set of size $k$. Consider the code $\mathcal{C} = \mathcal{D} \cup \{\{a_x, b_x, c_x\} \mid x \in V(G)\}$. One can easily check that each pair $x, y$ of original vertices of $G$ are separated by $a_x$ and $a_y$, and $x$ is separated from $a_y, b_y, c_y, d_y$ by at least one of $a_y, b_y$. For each original vertex $x$ of $G$, since $\mathcal{D}$ is a dominating set of $G$, $x$ and $d_x$ are separated by the vertex of $\mathcal{D}$ that dominates $x$. Vertices $a_x, b_x, c_x, d_x$ are easily seen to be separated among themselves by one of $a_x, b_x, c_x$, as well as $a_x, b_x, c_x$ are separated from $x$ by at least one of $b_x$, $c_x$.

**Necessary side (⇐)** Let $\mathcal{C}$ be an identifying code of $G'$ of size $k+3|V(G)|$. Observe that vertices $b_x$, $c_x$ and $d_x$ have the same open neighbourhoods, so $|\mathcal{C} \cap \{b_x, c_x, d_x\}| \geq 2$. For the same reason, $|\mathcal{C} \cap \{a_x, e_x\}| \geq 1$. Without loss of generality, we may assume that $\mathcal{C} \cap \{b_x, c_x, d_x\} = \{b_x, c_x\}$ and $\mathcal{C} \cap \{a_x, e_x\} = \{a_x\}$. Now, since $\mathcal{C}$ is an identifying code, $x$ and $e_x$ are separated, that is, $\mathcal{C} \cap (N[x] \cup \{d_x\} \setminus \{a_x, e_x\})) \neq \emptyset$. We build $\mathcal{D}$ as follows: first, $\mathcal{D} = \mathcal{C} \cap V(G)$. For each $x$ such that $x, d_x$ are separated by $d_x$ in $\mathcal{C}$, add $x$ to $\mathcal{D}$. It is easy to observe that $\mathcal{D}$ is a dominating set, and by the first part of the proof, that $|\mathcal{D}| \leq |\mathcal{C}| - 3|V(G)| = k$. □

A similar reduction to Min Loc-Dom Set can be given:

**Reduction 32** (Min Dominating Set → Min Loc-Dom Set)**.** *Given a graph $G$, we construct the graph $G'$ on vertex set*

$$V(G') = V(G) \cup \{a_x, b_x, c_x, d_x \mid x \in V(G)\},$$

*and edge set*

$$E(G') = E(G) \cup \{\{x, a_x\}, \{x, d_x\}, \{a_x, b_x\}, \{a_x, c_x\}, \{d_x, b_x\}, \{d_x, c_x\} \mid x \in V(G)\}.$$

*The construction is illustrated in Figure 9.*

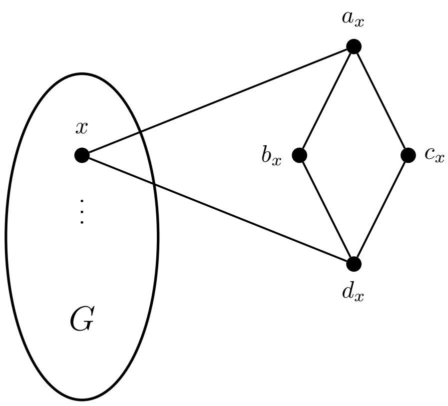



Figure 9: Reduction from Min Dominating Set to Min Loc-Dom Set.

Again, similar arguments than for Reduction 30 apply to Reduction 32, leading to the following result:

**Theorem 33.** Loc-Dom Set *is NP-complete, even for chordal bipartite graphs.*

## 4. Further classes of graphs for which the complexities of Dominating Set, Id Code and Loc-Dom Set differ

An interesting question is for which classes of graphs the complexities of the decision problems Dominating Set, Id Code and Loc-Dom Set differ. We saw in Subsection 2.4 that for co-bipartite graphs, Id Code and Loc-Dom Set are hard, but Dominating Set is trivially solvable in polynomial time.[5] Such a result was not known prior to the author's PhD thesis [28], from which many results of this paper are taken. In [28, 31], it was also proved that Id Code is NP-complete for the class of interval graphs, whereas Dominating Set is linear-time solvable in that class [9].

In this section, we define a subclass of bipartite graphs for which the converse holds: Dominating Set is NP-complete, but Id Code and Loc-Dom Set are solvable in polynomial time. We call these graphs *SC1-graphs* (the name comes from the fact that the hardness of the considered problems is due to their similarity to instances of Set Cover). We also define the similar class of *SC2-graphs*, for which, interestingly, the complexities of Id Code and Loc-Dom Set differ. Such a class was not known to exist before this work.

**Definition 34.** *A graph $G$ is said to be an* SC1-graph *if it can be built from a bipartite graph with parts $S$ and $T$ and an additional set $S'$ disjoint from $S$ and $T$ with $|S'| = 5|S|$ such that:*

- *for each vertex $x$ of $S$, there are two vertex-disjoint paths $xa_xb_xc_x$ and $xd_xe_x$ of length 3 and 2, respectively, with $a_x, b_x, c_x, d_x, e_x \in S'$ and no other edges from vertices of $S'$ than those of the two paths, and*
- *each vertex of $T$ has a distinct neighbourhood within $S$, and this neighbourhood has at least two elements.*

An example of an SC1-graph is pictured in Figure 10.

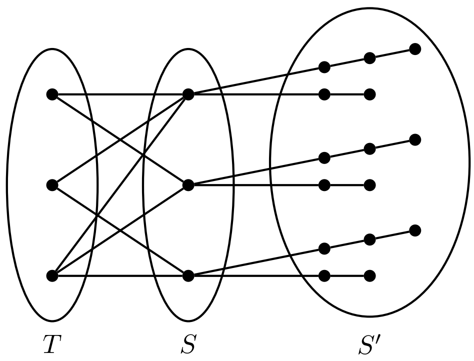



Figure 10: Example of an SC1-graph.

**Theorem 35.** *Let $G$ be an SC1-graph built from a bipartite graph with parts $S$ and $T$. Then $\gamma^{\text{ID}}(G) = 4|S|$, $S \cup \{a_s, b_s, d_s \mid s \in S\}$ is an optimal identifying code of $G$, and it can be computed in polynomial time.*

*Proof.* Note that each vertex $s$ of $S$ necessarily belongs to any identifying code, since it is the only one separating $d_s$ from $e_s$. Similarly, each vertex $a_s$ with $s \in S$ is the only one separating $b_s$ from $c_s$. In order to dominate $c_s$ and $e_s$, one of $b_s, c_s$ and $d_s, e_s$ respectively, has to belong to any identifying code. Hence $\gamma^{\text{ID}}(G) \geq 4|S|$.

Using the facts that each vertex of $T$ is adjacent to at least two vertices of $S$ and that all the neighbourhoods of vertices of $T$ within $S$ are distinct, it is easy to show that $S \cup \{a_s, b_s, d_s \mid s \in S\}$ is an identifying code.

[5] Recall that this class is included in the class of asteroidal triple-free graphs, which itself is a subclass of the class of Dominating Shortest Path graphs, in which Dominating Set is still solvable in polynomial time [42].

Finally, one can easily check in polynomial time whether a given graph is an SC1-graph; once this is done, it is straightforward to compute the code $S \cup \{a_s, b_s, d_s \mid s \in S\}$. $\square$

We have a similar statement for locating-dominating sets:

**Theorem 36.** *Let $G$ be an SC1-graph built from a bipartite graph with parts $S$ and $T$. Then $\gamma^{\text{LD}}(G) = 3|S|$, $S \cup \{b_s, d_s \mid s \in S\}$ is an optimal locating-dominating set of $G$, and it can be computed in polynomial time.*

*Proof.* As in the proof of Theorem 35, for each $s \in S$, in order to dominate $c_s$ and $e_s$, one of $b_s, c_s$ and $d_s, e_s$ respectively, has to belong to any locating-dominating set. For any possible choice of vertex among $b_s, c_s$, one additional vertex among $s, a_s, b_s, c_s$ is needed (either to separate $a_s$ from $c_s$ or to dominate $b_s$). Hence $\gamma^{\text{ID}}(G) \geq 3|S|$.

As in the proof of Theorem 35, it is easy to show that $S \cup \{b_s, d_s \mid s \in S\}$ is a locating-dominating set and that it can be computed in polynomial time. $\square$

**Theorem 37.** Dominating Set *is NP-complete for planar (bipartite) SC1-graphs of maximum degree 5.*

*Proof.* We reduce Set Cover to Dominating Set for SC1-graphs. Let $(I, A)$ be a hypergraph such that each vertex of $A$ has a distinct neighbourhood within $I$ and at least two neighbours in $I$. For example, one can take an instance of Vertex Cover for planar cubic graphs, which is a special case of Set Cover (where $I$ is the set of edges of a simple graph; each set of $A$ stands for a given vertex and contains all edges incident to it), known to be NP-complete [1, 32]. Let $\mathcal{B}(I, A)$ be the bipartite incidence graph of $(I, A)$, and build the SC1-graph $G$ from $\mathcal{B}(I, A)$ with parts $S = A$ and $T = I$. If $(I, A)$ comes from Vertex Cover for planar graphs of maximum degree 3, $G$ is planar and has maximum degree 5. We claim that $(I, A)$ has a set cover of size $k$ if and only if $G$ has a dominating set of size $k + 2|S|$.

For the first part, let $\mathcal{C} \subseteq A$ be a set cover of $(I, A)$. One can easily check that the set $\mathcal{C} \cup \{b_s, d_s \mid s \in S\}$ is a dominating set of $G$.

For the converse, let $\mathcal{D}$ be a dominating set of $G$ of size $k + 2|S|$. Since for each vertex $s \in S$, $c_s$ and $e_s$ need to be dominated, we have $|\mathcal{D} \cap S'| \geq 2|S|$. In fact, we can assume that for each vertex $s \in S$, $|\mathcal{D} \cap S'| = \{b_s, d_s\}$; then all vertices of $S \cup S'$ are dominated by some vertex of $\mathcal{D} \cap S'$. We can also assume that $\mathcal{D} \cap T = \emptyset$, since all vertices of $S$ are dominated by some vertex of $\mathcal{D} \cap S'$: if a vertex $t \in T$ belongs to $\mathcal{D}$, we can replace it by an arbitrary neighbour of $t$ in $S$ to get a dominating set $\mathcal{D}'$ with $|\mathcal{D}'| \leq |\mathcal{D}|$. Observe that $\mathcal{D}' \cap S$ has to dominate all the vertices of $T$, hence $(I, A)$ has a set cover of size $|\mathcal{D}' \cap S| \leq |\mathcal{D}| - 2|S| = k$. $\square$

We now introduce the class of SC2-graphs.

**Definition 38.** *A graph $G$ is said to be an* SC2-graph *if it can be built from a bipartite graph with parts $S$ and $T$ and an additional set $S'$ disjoint from $S$ and $T$ with $|S'| = |S|$ such that:*

- *for each vertex $x$ of $S$, there is a vertex $l_x \in S'$ of degree 1 adjacent to $x$, and*
- *each vertex of $T$ has a distinct neighbourhood within $S$, and this neighbourhood has at least two elements.*

An example of an SC2-graph is pictured in Figure 11.

In what follows, $\gamma(G)$ denotes the size of a smallest dominating set of graph $G$.

**Theorem 39.** *Let $G$ be an SC2-graph built from a bipartite graph with parts $S$ and $T$. Then $\gamma(G) = \gamma^{\text{LD}}(G) = |S|$, $S$ is an optimal (locating-)dominating set of $G$, and it can be computed in polynomial time.*

*Proof.* Since for every $x \in S$, every vertex $l_x$ needs to be dominated, either $x$ or $l_x$ belongs to any dominating set of $G$. It is easily observed that $S$ is a dominating set of $G$. Moreover, since by the definition of an SC2-graph, each vertex of $T$ has a distinct neighbourhood within $S$, and this neighbourhood has at least two elements, all vertices of $T$ and $S'$ are separated, hence $S$ is also locating-dominating. Now, as for SC1-graphs, one can easily check in polynomial time whether a given graph is an SC2-graph; once this is done, it is straightforward to compute $S$. $\square$

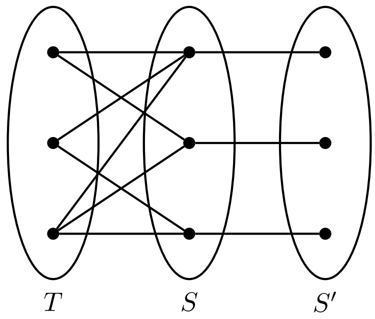



Figure 11: Example of an SC2-graph.

**Theorem 40.** Id Code *is NP-complete for planar (bipartite) SC2-graphs of maximum degree 3.*

*Proof.* We observe that Reduction 25 from Subsection 3.2 yields planar SC2 graphs of maximum degree 3 when applied to Vertex Cover for planar graphs of maximum degree 3. Hence Theorem 27 shows the claim. □

## 5. Open problems

We conclude with open problems. The complexities of (Min) Id Code and (Min) Loc-Dom Set are open for several important input graph classes, as shown in Tables 1 and 2. Regarding interval graphs and permutation graphs, the approximation complexity of Min Id Code (and Min Loc-Dom Set) is still an open question.[6] It is also of interest to determine the complexity of Id Code and Loc-Dom Set for *bipartite* permutation graphs and *unit* interval graphs. Finally, we remark that Min Dominating Set admits PTAS algorithms for planar graphs [5] and for unit disk graphs [38, 50]. Does the same hold for Min Id Code and Min Loc-Dom Set?

## References


[1] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science* 237(1-2):123–134, 2000.

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

[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] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM* 41(1):153–180, 1994.

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

[7] A. A. Bertossi. Dominating sets for split and bipartite graphs. *Information Processing Letters* 19:37–40, 1984.

[8] B. Bollobas and A. D. Scott. On separating systems. *European Journal of Combinatorics* 28:1068–1071, 2007.

[9] K. S. Booth and H. J. Johnson. Dominating sets in chordal graphs. *SIAM Journal of Computing* 11(1):191–199, 1982.

[10] N. Bousquet, A. Lagoutte, Z. Li, A. Parreau and S. Thomassé. Identifying codes in hereditary classes of graphs and VC-dimension. Manuscript, 2014. `http://arxiv.org/abs/1407.5833`

[11] A. Brandstädt, V. B. Lê and J. P. Spinrad. *Graph classes: a survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.

[12] E. Charbit, I. Charon, G. Cohen, O. Hudry and A. Lobstein. Discriminating codes in bipartite graphs: bounds, extremal cardinalities, complexity. *Advances in Mathematics of Communications* 4(2):403–420, 2008.


[6] After the submission of this paper, a 6-approximation algorithm for Min Id Code (implying a 12-approximation algorithm for Min Loc-Dom Set by Corollary 6) appeared in [10]; however it is not known whether a PTAS exists.

[13] I. Charon, G. Cohen, O. Hudry and A. Lobstein. Discriminating codes in (bipartite) planar graphs. *European Journal of Combinatorics* 29(5):1353–1364, 2008.
[14] B. N. Clark, C. J. Colbourn and D. S. Johnson. Unit disk graphs. *Discrete Mathematics* 86(1-3):165–177, 1990.
[15] C. Colbourn, P. J. Slater and L. K. Stewart. Locating-dominating sets in series-parallel networks. *Congressus Numerantium* 56:135–162, 1987.
[16] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science* 109:49–82, 1993.
[17] 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.
[18] M. Chlebík and J. Chlebíková. Approximation hardness of edge dominating set problems. *Journal of Combinatorial Optimization* 11:279–290, 2006.
[19] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation* 206:1264–1275, 2008.
[20] D. G. Corneil and L. K. Stewart. Dominating sets in perfect graphs. *Discrete Mathematics* 86(1–3):145–164, 1990.
[21] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation* 85(1):12–75, 1990.
[22] P. Crescenzi, V. Kann, R. Silvestri and L. Trevisan. Structure in approximation classes. *SIAM Journal on Computing* 28(5):1759–1782, 1999.
[23] P. Damaschke, H. Müller and D. Kratsch. Domination in convex and chordal bipartite graphs. *Information Processing Letters* 36:52:31–236, 1990.
[24] K. M. J. De Bontridder, B. V. Halldórsson, M. M. Halldórsson, C. A. J. Hurkens, J. K. Lenstra, R. Ravi and L. Stougie. Approximation algorithms for the test cover problem. *Mathematical Programming Series B* 98:477–491, 2003.
[25] H. N. De Ridder and others. Information System on Graph Classes and their Inclusions (ISGCI). `http://www.graphclasses.org`
[26] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics* 7:115–130, 1984.
[27] M. Farber and J. M. Keil. Domination in permutation graphs. *Journal of Algorithms* 6:309–321, 1985.
[28] F. Foucaud. *Combinatorial and algorithmic aspects of identifying codes in graphs.* PhD thesis, Université Bordeaux 1, France, December 2012. `http://tel.archives-ouvertes.fr/tel-00766138`.
[29] F. Foucaud. The complexity of the identifying code problem in restricted graph classes. *Proceedings of the International Workshop on Combinatorial Algorithms, IWOCA 2013, Lecture Notes in Computer Science* 8288:150–163, 2013.
[30] 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.
[31] F. Foucaud, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. Manuscript, 2014. `http://arxiv.org/abs/1405.2424`
[32] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics* 32(4):826–834, 1977.
[33] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman, 1979.
[34] S. Gravier, R. Klasing and J. Moncel. Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs. *Algorithmic Operations Research* 3(1):43–50, 2008.
[35] T. W. Haynes, S. T. Hedetniemi and P. J. Slater. *Fundamentals of domination in graphs*, Marcel Dekker, 1998.
[36] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, editors. *Domination in graphs: advanced topics*, Marcel Dekker, 1998.
[37] T. W. Haynes, D. J. Knisley, E. Seier and Y. Zou. A quantitative analysis of secondary RNA structure using domination based parameters on trees. *BMC Bioinformatics* 7:108, 2006.
[38] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz and R. E. Stearns. NC-approximation schemes for NP-and PSPACE-hard problems for geometric graphs. *Journal of Algorithms* 26:238–274, 1998.
[39] D. S. Johnson, Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* 9:256–278, 1974.
[40] 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* 44:599–611, 1998.
[41] J. H. Kim, O. Pikhurko, J. Spencer and O. Verbitsky. How complex are random graphs in First Order logic? *Random Structures and Algorithms* 26(1-2):119–145, 2005.
[42] D. Kratsch. Domination and total domination on asteroidal triple-free graphs. *Discrete Applied Mathematics* 99(1-3):111–123, 2000.
[43] M. Laifenfeld and A. Trachtenberg. Identifying codes and covering problems. *IEEE Transactions on Information Theory* 54(9):3929–3950, 2008.
[44] M. Laifenfeld, A. Trachtenberg, R. Cohen and D. Starobinski. Joint monitoring and routing in wireless sensor networks using robust identifying codes. *Proceedings of IEEE Broadnets 2007*, pages 197–206, September 2007.
[45] A. Lobstein. Watching systems, identifying, locating-dominating and discriminating codes in graphs: a bibliography. `http://www.infres.enst.fr/~lobstein/debutBIBidetlocdom.pdf`
[46] J. Moncel. *Codes identifiants dans les graphes.* PhD thesis, Université Joseph Fourier Grenoble I, France, 2005.
[47] B. M. E. Moret and H. D. Shapiro. On minimizing a set of tests. *SIAM Journal of Scientifical and Statistical Computation* 6(4):983–1003, 1985.
[48] H. Müller and A. Brandtädt. The NP-completeness of Steiner Tree and Dominating Set for chordal bipartite graphs.

*Theoretical Computer Science* 53:257–265, 1987.
[49] T. Müller and J.-S. Sereni. Identifying and locating-dominating codes in (random) geometric networks. *Combinatorics, Probability and Computing* 18(6):925–952, 2009.
[50] T. Nieberg and J. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. *Proceedings of the Third international conference on Approximation and Online Algorithms, WAOA'05*, *Lecture Notes in Computer Science* 3879:296–306, 2006.
[51] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences* 43(3):425–440, 1991.
[52] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 475–484, 1997.
[53] A. Rényi. On random generating elements of a finite Boolean algebra, *Acta Scientiarum Mathematicarum Szeged* 22:75–81, 1961.
[54] P. J. Slater. Domination and location in acyclic graphs. *Networks* 17(1):55–64, 1987.
[55] P. J. Slater and D. F. Rall. On location-domination numbers for certain classes of graphs. *Congressus Numerantium* 45:97–106, 1984.
[56] J. Suomela. Approximability of identifying codes and locating-dominating codes. *Information Processing Letters* 103(1):28–33, 2007.
[57] J. Suomela (`http://cstheory.stackexchange.com/users/74`). Answer to the question "Is the dominating set problem restricted to planar bipartite graphs of maximum degree 3 NP-complete?", `http://cstheory.stackexchange.com/a/2508/1930`.
[58] R. Ungrangsi, A. Trachtenberg and D. Starobinski. An implementation of indoor location detection systems based on identifying codes. *Proceedings of Intelligence in Communication Systems, INTELLCOMM 2004*, *Lecture Notes in Computer Science* 3283:175–189, 2004.
[59] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics* 38:364–372, 1980.

## Appendix A. Omitted proofs

*Proof of Theorem 17.* **Sufficient side ($\Rightarrow$)** Let $\mathcal{D} \subseteq A$ be a discriminating code of $(I, A)$, $|\mathcal{D}| = k$. We define $\mathcal{C}(\mathcal{D})$ as follows:

$$\mathcal{C}(\mathcal{D}) = \mathcal{D} \cup \{u\} \cup \{k_j, t_j \mid 1 \leq j \leq 2\lceil \log_2(|A|+1) \rceil\}.$$

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 2\lceil \log_2(|A|+1)\rceil + 1$ and is a dominating set of $Sp(I, A)$. To see that it is also an identifying code of $Sp(I, A)$, observe that each vertex of $K$ is separated from each vertex of $S$ by $u$. Moreover vertex $u$ is the only vertex that is dominated only by the vertices of $\mathcal{C}(\mathcal{D})$ from $K$. All pairs of vertices of $K$ are separated: each vertex $k_i$ is separated from each other vertex of $K$ by its private neighbour $t_i$, and since $\mathcal{D}$ is a discriminating code of $(I, A)$, each vertex of $\mathcal{I}$ is dominated by a distinct and nonempty set of vertices of $\mathcal{D}$. Finally, all pairs of vertices of $S$ are separated: due to the non-singleton bipartite logarithmic identification of $A$, each vertex of $A$ is dominated by a distinct subset of vertices of $\{k_j \mid 1 \leq j \leq 2\lceil \log_2(|A|+1)\rceil\}$ that has size at least 2. Finally, each vertex $s_i$ is the only vertex dominated only by $k_i$, and each vertex $t_i$ is the only vertex of $S$ dominated by itself.

**Necessary side ($\Leftarrow$)** Let $\mathcal{C}$ be an identifying code of $Sp(I, A)$ with $|\mathcal{C}| = k + 4\lceil \log_2(|A|+1)\rceil + 1$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $Sp(I, A)$, $|\mathcal{C}^*| \leq |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap (V(Sp(I, A)) \setminus (I \cup A))| = 4\lceil \log_2(|A|+1)\rceil + 1 \tag{A.1}$$

$$|\mathcal{C}^* \cap I| = \emptyset. \tag{A.2}$$

To get Condition (A.1), we replace $|\mathcal{C} \cap \{V(Sp(I, A)) \setminus \{I \cup A\}\}|$ by $\{u\} \cup \{k_j, t_j \mid 1 \leq j \leq 2\lceil \log_2(|A|+1)\rceil\}$ to get code $\mathcal{C}'$ (whose structure is similar to the one of the code constructed in the ($\Rightarrow$) part of the proof). Observe that $|\mathcal{C}'| \leq |\mathcal{C}|$. Indeed, we had $|\mathcal{C} \cap \{V(Sp(I, A)) \setminus \{I \cup A\}\}| \geq 4\lceil \log_2(|A|+1)\rceil + 1$. To see this, note that for any $j \in \{1, \ldots, 2\lceil \log_2(|A|+1)\rceil\}$, $|\mathcal{C} \cap \{k_j, s_j, t_j\}| \geq 2$. Indeed, $N(s_j) = N(t_j)$; as they must be separated by $\mathcal{C}$, one of them, say $s_j$, belongs to $\mathcal{C}$. Bu $t_j$ must be dominated, hence one of $k_j$ and $t_j$ belongs to $\mathcal{C}$. Finally, $v$ must be dominated, hence $|\mathcal{C} \cap \{u, v\}| \geq 1$.

To fulfill Condition (A.2), we note that each vertex $i \in I \cap \mathcal{C}'$ can simply be removed from the code. Assume for the sake of contradiction, that $\mathcal{C}' \setminus \{i\}$ is not an identifying code. Note that $i$ cannot be needed for domination since all vertices of $I$ are dominated (e.g. by $u$) and all vertices of $A$ are dominated by some vertex in $\{k_j \mid 1 \le j \le 2\lceil \log_2(|A|+1)\rceil\}$. Hence, $i$ is needed for separation. Since $K$ is a clique and contains already many vertices of $\mathcal{C}'$ (i.e. $u$ and all vertices of $\{k_j \mid 1 \le j \le 2\lceil \log_2(|A|+1)\rceil\}$), $i$ may only separate two vertices of $S$ (no vertex of $S$ is adjacent to all the vertices of $\mathcal{C}' \cap K$, hence all vertices of $S$ are separated from all vertices of $K$). Actually, these two vertices have to both belong to $A$ since no other vertex from $S$ can be adjacent to $I$. But all pairs in $A$ are separated by some vertex in $\{k_j \mid 1 \le j \le 2\lceil \log_2(|A|+1)\rceil\}$, a contradiction. Removing every $i \in \mathcal{C}' \cap I$ in this way, we get code $\mathcal{C}^*$, and $|\mathcal{C}^*| \le |\mathcal{C}'| \le |\mathcal{C}|$.

Using the previous observations and by similar arguments as in in the ($\Rightarrow$) part of the proof, one can easily check that after these two modifications performed on code $\mathcal{C}$, the obtained code $\mathcal{C}^*$ is still an identifying code.

By Condition (A.2), we have $|\mathcal{C}^* \cap A| \le |\mathcal{C}| - 4\lceil \log_2(|A|+1)\rceil + 1 = k$.

To finish the proof, we claim that $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. This is easy to observe, as all pairs $\{I, I'\}$ of $I$ are dominated and separated by $\mathcal{C}^*$. By Condition (A.1), they must be separated by some vertex of $A$. Hence $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. □

*Proof of Theorem 20.* We first assume that $a_1$ is the vertex adjacent to all vertices of $A$ as given by the construction of $E(\mathcal{LOG}(A, L))$.

**Sufficient side ($\Rightarrow$)** Let $\mathcal{D} \subseteq A$ be a discriminating code of $(I, A)$, $|\mathcal{D}| = k$. Without loss of generality, we assume that $a_1$ is adjacent to some vertex of $\mathcal{D}$. We define $\mathcal{C}(\mathcal{D})$ as follows:

$$\mathcal{C}(\mathcal{D}) = \mathcal{D} \cup \{a_j, b_j, c_j, d_j, f_j \mid 1 \le j \le \lceil \log_2(|A|+1)\rceil\} \setminus \{b_1, f_1\}.$$

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 5\lceil \log_2(|A|+1)\rceil - 2$ and is a dominating set of $G(I, A)$. Let us show that it is also an identifying code of $G(I, A)$. First of all, due to the bipartite logarithmic identification of $A$ over $(A, \{a_j \mid 1 \le j \le \lceil \log_2(|A|+1)\rceil\})$, each vertex of $\mathcal{A}$ is dominated by a distinct subset of vertices of $\{a_j \mid 1 \le j \le \lceil \log_2(|A|+1)\rceil\}$; note that any other vertex (except $e_1$, which however is not dominated by any vertex $a_i$) is dominated by some vertex $b_i$. Hence each vertex of $\mathcal{A}$ is separated from all other vertices. Next, each vertex of $\mathcal{I}$ is dominated by a distinct nonempty subset of $\mathcal{D}$ since $\mathcal{D}$ is a discriminating code of $(I, A)$. Within $V(G) \setminus (A \cup I)$, only vertices of the form $a_i$ may be dominated by vertices of $\mathcal{D}$; however each vertex $a_i$ is separated from any vertex of $\mathcal{I}$ by $d_i$. It remains to check that vertices of the form $a_i, b_i, c_i, d_i, e_i, f_i$ are separated from each other. For any $i, j$ (possibly $i = j$), any vertex among $\{a_i, b_i, c_i\}$ is separated from any vertex of $\{d_j, e_j, f_j\}$ by the set $\{a_k \mid 1 \le k \le \lceil \log_2(|A|+1)\rceil\}$. Similarly, for $i \ne j$, any vertex of $\{a_i, b_i, c_i\}$ is separated from any vertex of $\{a_j, b_j, c_j\}$ by either $d_i, d_j, f_i$ or $f_j$ (noticing that each vertex $c_i$ except $c_1$ is dominated by $f_i$). Again, for $i \ne j$, $d_i, e_i, f_i$ are separated from $d_j, e_j, f_j$ by at least one of $c_i, c_j$ (noticing that each vertex among $\{d_i, e_i, f_i\}$ is dominated by $b_i$, except when $i = 1$). For any $i$, it remains to check the separation of any pair within $\{a_i, b_i, c_i\}$ and within $\{a_i, b_i, c_i\}$. If $i \ne 1$, observe that $a_i$ is dominated by $d_i$, $b_i$ is dominated by both $d_i, f_i$, and $c_i$ is dominated by $f_i$. Furthermore, $a_1, b_1$ and $a_1, c_1$ are separated by some vertex of $\mathcal{D}$ that is adjacent to $a_1$ (we assumed that it exists); $b_1, c_1$ are separated by $d_1$. Finally for any $i$, $d_i$ is separated from both $e_i, f_i$ by $a_i$; $e_i$ and $f_i$ are separated by $c_i$.

**Necessary side ($\Leftarrow$)** Let $\mathcal{C}$ be an identifying code of $G(I, A)$, $|\mathcal{C}| = k + 5\lceil \log_2(|A|+1)\rceil - 2$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $G(I, A)$, $|\mathcal{C}^*| \le |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap \{V(G(I, A)) \setminus \{I \cup A\}\}| = 5\lceil \log_2(|A|+1)\rceil - 2 \quad \text{(A.3)}$$
$$|\mathcal{C}^* \cap I| = \emptyset. \quad \text{(A.4)}$$

To get Condition (A.3), we first replace $|\mathcal{C} \cap \{V(G(I, A)) \setminus \{I \cup A\}\}|$ by $\{a_j, b_j, c_j, d_j, f_j \mid 1 \le j \le \lceil \log_2(|A|+1)\rceil\} \setminus \{b_1, f_1\}$ to get code $\mathcal{C}'$ (whose structure is similar to the one of the code constructed in the ($\Rightarrow$) part of the proof). Observe that $|\mathcal{C}'| \le |\mathcal{C}|$. Indeed, we had $|\mathcal{C} \cap \{V(G(I, A)) \setminus \{I \cup A\}\}| \ge 5\lceil \log_2(|A|+1)\rceil - 2$. To see this, note that for any $j \in \{1, \ldots, \log_2(|A|+1)\}$, vertices $a_j, c_j$ are forced by $\{d_j, e_j\}$ and $\{e_j, f_j\}$, respectively, and $|\mathcal{C} \cap \{d_j, e_j\}| \ge 1$ since $\mathcal{C}$ must separate $b_j$ from $c_j$. Finally, consider the two sets $F = \{f_j \mid j \in \{1, \ldots, \log_2(|A|+1)\}\}$ and $B = \{b_j \mid j \in \{1, \ldots, \log_2(|A|+1)\}\}$. Finally, observe that at least $|F|-1$ vertices of $F$ ($|B|-1$ vertices and of $B$, respectively) do not need to belong to $\mathcal{C}$. Indeed, for any pair $c_i, c_j$ of vertices with $i \neq j$ and $1 \leq i, j \leq \lceil \log_2(|A|+1) \rceil$ ($e_i, e_j$, respectively), either $f_i$ or $f_j$ ($b_i$ or $b_j$, respectively) must belong to $\mathcal{C}$.

To fulfill Condition (A.4), we replace each vertex $i \in I \cap \mathcal{C}'$ by some vertex in $A$. If $\mathcal{C}' \setminus \{i\}$ is an identifying code, we may just remove $i$ from the code. Otherwise, note that $i$ is not needed for domination since all vertices of $K^1 \cup A$ are dominated by $a_1$. Hence, $i$ separates $i$ itself from some other vertex $i'$ in $I$ (indeed, one can check that all other types of pairs which could be separated by $i$ are actually already separated by some vertex of $\mathcal{C}' \cap (V(G(I,A)) \setminus I)$. But then, the pair $\{i, i'\}$ is unique (suppose $i$ separates $i$ itself from two distinct vertices $i'$ and $i''$ of $I$, then $i'$ and $i''$ would not be separated by $\mathcal{C}'$, a contradiction). Since $(I, A)$ admits a discriminating code, there must be some vertex $a$ of $A$ separating $i$ from some $i'$. Hence we replace $i$ by $a$. Doing this for every $i \in \mathcal{C}' \cap I$, we get code $\mathcal{C}^*$, and $|\mathcal{C}^*| \leq |\mathcal{C}'| \leq |\mathcal{C}|$.

Using the previous observations and by similar arguments as in in the ($\Rightarrow$) part of the proof, one can easily check that after these two modifications performed on code $\mathcal{C}$, the obtained code $\mathcal{C}^*$ is still an identifying code.

By Condition (A.4), we have $|\mathcal{C}^* \cap A| \leq |\mathcal{C}| - 5\lceil \log_2(|A|+1) \rceil + 2 = k$. To complete the proof, we claim that $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. This is easy to observe, as all pairs $\{I, I'\}$ of $I$ are separated by $\mathcal{C}^*$. By Condition (A.3), they must be separated by some vertex of $A$. Hence $\mathcal{C}^* \cap A$ is a discriminating code of $(I, A)$. $\square$