PhD Thesis

Combinatorial and algorithmic aspects of identifying codes in graphs

Florent Foucaud

Supervisors:
Ralf Klasing, André Raspaud

Fall 2012

Aspects combinatoires et algorithmiques des codes identifiants dans les graphes

Résumé : Un code identifiant est un ensemble de sommets d'un graphe tel que, d'une part, chaque sommet hors du code a un voisin dans le code (propriété de domination) et, d'autre part, tous les sommets ont un voisinage distinct à l'intérieur du code (propriété de séparation). Dans cette thèse, nous nous intéressons à des aspects combinatoires et algorithmiques relatifs aux codes identifiants.

Pour la partie combinatoire, nous étudions tout d'abord des questions extrémales en donnant une caractérisation complète des graphes non-orientés finis ayant comme taille minimum de code identifiant leur ordre moins un. Nous caractérisons également les graphes dirigés finis, les graphes non-orientés infinis et les graphes orientés infinis ayant pour seul code identifiant leur ensemble de sommets. Ces résultats répondent à des questions ouvertes précédemment étudiées dans la littérature.

Puis, nous étudions la relation entre la taille minimum d'un code identifiant et le degré maximum d'un graphe, en particulier en donnant divers majorants pour ce paramètre en fonction de l'ordre et du degré maximum. Ces majorants sont obtenus via deux techniques. L'une est basée sur la construction d'ensembles indépendants satisfaisant certaines propriétés, et l'autre utilise la combinaison de deux outils de la méthode probabiliste : le lemme local de Lovász et une borne de Chernoff. Nous donnons également des constructions de familles de graphes en relation avec ce type de majorants, et nous conjecturons que ces constructions sont optimales à une constante additive près.

Nous présentons également de nouveaux minorants et majorants pour la cardinalité minimum d'un code identifiant dans des classes de graphes particulières. Nous étudions les graphes de maille au moins 5 et de degré minimum donné en montrant que la combinaison de ces deux paramètres influe fortement sur la taille minimum d'un code identifiant. Nous appliquons ensuite ces résultats aux graphes réguliers aléatoires. Puis, nous donnons des minorants pour la taille d'un code identifiant des graphes d'intervalles et des graphes d'intervalles unitaires. Enfin, nous donnons divers minorants et majorants pour cette quantité lorsque l'on se restreint aux graphes adjoints. Cette dernière question est abordée via la notion nouvelle de codes arête-identifiants.

Pour la partie algorithmique, il est connu que le problème de décision associés à la notion de code identifiant est NP-complet même pour des classes de graphes restreintes. Nous étendons ces résultats à d'autres classes de graphes telles que celles des graphes split, des co-bipartis, des adjoints ou d'intervalles. Pour cela nous proposons des réductions polynomiales depuis divers problèmes algorithmiques classiques. Ces résultats montrent que dans beaucoup de classes de graphes, le problème des codes identifiants est algorithmiquement plus difficile que des problèms liés (tel que le problème des ensembles dominants).

Par ailleurs, nous complétons les connaissances relatives à l'approximabilité du problème d'optimisation associé aux codes identifiants. Nous étendons le résultat connu de NP-difficulté pour l'approximation de ce problème avec un facteur sous-logarithmique (en fonction de la taille du graphe instance) aux graphes bipartis, split et co-bipartis, respectivement. Nous étendons également le résultat connu d'APX-complétude pour les graphes de degré maximum donné à une sous-classe des graphes split, aux graphes bipartis de degré maximum 4 et aux graphes adjoints. Enfin, nous montrons l'existence d'un algorithme de type PTAS pour les graphes d'intervalles unitaires.

Mots-clé : codes identifiants, ensembles dominants, théorie des graphes, NP-complétude, algorithmes d'approximation

Combinatorial and algorithmic aspects of identifying codes in graphs

Abstract: An identifying code is a set of vertices of a graph such that, on the one hand, each vertex out of the code has a neighbour in the code (domination property), and, on the other hand, all vertices have a distinct neighbourhood within the code (separation property). In this thesis, we investigate combinatorial and algorithmic aspects of identifying codes.

For the combinatorial part, we first study extremal questions by giving a complete characterization of all finite undirected graphs having their order minus one as minimum size of an identifying code. We also characterize finite directed graphs, infinite undirected graphs and infinite oriented graphs having their whole vertex set as unique identifying code. These results answer open questions that were previously studied in the literature.

We then study the relationship between the minimum size of an identifying code and the maximum degree of a graph. In particular, we give several upper bounds for this parameter as a function of the order and the maximum degree. These bounds are obtained using two techniques. The first one consists in the construction of independent sets satisfying certain properties, and the second one is the combination of two tools from the probabilistic method: the Lovász local lemma and a Chernoff bound. We also provide constructions of graph families related to this type of upper bounds, and we conjecture that they are optimal up to an additive constant.

We also present new lower and upper bounds for the minimum cardinality of an identifying code in specific graph classes. We study graphs of girth at least 5 and of given minimum degree by showing that the combination of these two parameters has a strong influence on the minimum size of an identifying code. We apply these results to random regular graphs. Then, we give lower bounds on the size of a minimum identifying code of interval and unit interval graphs. Finally, we prove several lower and upper bounds for this parameter when considering line graphs. The latter question is tackled using the new notion of an edge-identifying code.

For the algorithmic part, it is known that the decision problem associated to the notion of an identifying code is NP-complete, even for restricted graph classes. We extend the known results to other classes such as split graphs, co-bipartite graphs, line graphs or interval graphs. To this end, we propose polynomial-time reductions from several classical hard algorithmic problems. These results show that in many graph classes, the identifying code problem is computationally more difficult than related problems (such as the dominating set problem).

Furthermore, we extend the knowledge of the approximability of the optimization problem associated to identifying codes. We extend the known result of NP-hardness of approximating this problem within a sub-logarithmic factor (as a function of the instance graph) to bipartite, split and co-bipartite graphs, respectively. We also extend the known result of its APX-hardness for graphs of given maximum degree to a subclass of split graphs, bipartite graphs of maximum degree 4 and line graphs. Finally, we show the existence of a PTAS algorithm for unit interval graphs.

Keywords: identifying codes, dominating sets, graph theory, NP-completeness, approximation algorithms

Contents

Chapter 1
Introduction

This thesis deals with the subject of identifying codes in graphs. This topic has been proposed to me during my Master thesis in early 2009 by my advisors R. Klasing and A. Raspaud. This has been the starting point of a fruitful line of research thanks to the active collaboration and exchange of ideas among the small group participating in the 3-year ANR[1] research project IDEA: Identifying coDes in Evolving grAphs (2009–2012).[2] In this thesis, we present results arising from this fruitful collaboration (see the publications or manuscripts [FGK+11, FGN+12, FKKR12, FKM+12, FNP12, FP12]) as well as results that are solely the author's work and have not yet been presented elsewhere than in this thesis.

I would like to point out that this thesis is part of a growing series of PhD manuscripts on the topic of identifying codes, that started with J. Moncel's thesis in 2005 [153], and continued with R. D. Skagg's and S. Ranto's theses in 2007 [179, 166], M. Laifenfeld's thesis in 2008 [139], D. Auger's thesis in 2010 [7], B. Stanton's and V. Junnila theses in 2011 [184, 128] and, most recently, M. Bouznif's, A. Parreau's and P. Valicov's theses in 2012 [30, 164, 195] (sorted by order of defense date).

In this introductory chapter, we will present and motivate the concept of an identifying code, before giving an overview of the author's results that are presented in this work. Formal definitions and notations about graphs and computational complexity will be given in Chapter 2.

1.1 Identifying codes in graphs

An identifying code of a graph G is a subset of vertices of G that allows to distinguish each vertex of G by means of its neighbourhood within the identifying code. This notion, defined by M. G. Karpovsky, K. Chakrabarty and L. B. Levitin in 1998 [131] has been widely studied since then (more than 240 publications are listed in a bibliography maintained by A. Lobstein and available online [142]). It is related to many other kinds of identification problems in combinatorial structures (graphs, digraphs and hypergraphs) such as the notions of test covers, discriminating codes, locating-dominating sets, etc. As we will see, all these problems have various applications, e.g. in the areas of testing of diseases, fault-detection in networks or location of threats in facilities.

[1] Agence Nationale de la Recherche

[2] See `http://idea.labri.fr`.

1.1.1 Formal definition

We first present the notions of a *dominating set* and a *separating code* of a graph. In fact, we will see later that identifying codes are exactly the combination of these two notions.

Dominating sets

A dominating set is a set of vertices of a graph that covers all closed neighbourhoods:

Definition 1.1. *A* dominating set *of a graph G is a subset $\mathcal{D}$ of vertices of G such that for each vertex $v \in V(G)$, $\mathcal{D} \cap N[v] \neq \emptyset$.*

An example of a dominating set is depicted in Figure 1.1(a). A vertex x of $\mathcal{D}$ is said to *dominate* vertex v if either $x = v$, or x is adjacent to v. The minimum size of a dominating set in G, its *domination number*, is denoted $\gamma(G)$.

In a digraph, we have the same definition but replacing the closed neighbourhood $N[v]$ by the closed in-neighbourhood $N^-[v]$: a vertex x dominates itself and all its out-neighbours.[3]

We point out that dominating sets and many of their variants have been studied extensively over the years; two classic textbooks about this topic are [108, 109].

Separating codes

A separating code of a graph is a subset of vertices that allows to distinguish all vertices from each other using their neighbourhood within the code:

Definition 1.2. *A* separating code *is a subset $\mathcal{C}$ of vertices of G such that for each pair u, v of distinct vertices of G, we have $N[u] \cap \mathcal{C} \neq N[v] \cap \mathcal{C}$.*

Equivalently, the condition $N[u] \cap \mathcal{C} \neq N[v] \cap \mathcal{C}$ can be replaced by $\mathcal{C} \cap (N[u] \ominus N[v]) \neq \emptyset$ (where $\ominus$ denotes the symmetric difference operator between two sets), that is, there is a vertex $x \in \mathcal{C}$ such that $x \in N[u]$ but $x \notin N[v]$, or $x \in N[v]$ but $x \notin N[u]$; in that case we say that x *separates* the pair u, v. In the literature, separating codes have also been designated under the name of *separating sets*.

An example of a graph with a separating code is represented in Figure 1.1(b), where the vertices are labelled and the sets $N[x] \cap \mathcal{C}$ are indicated.

The minimum size of a separating code of a graph G, its *separating code number*, is denoted $\gamma^{\mathrm{S}}(G)$.

When considering digraphs, as in the case of dominating sets, the notion of a separating code has the same definition as in the undirected case, except that in the definition, we replace the closed neighbourhood by the closed in-neighbourhood. The separating code number of a digraph D is denoted $\overrightarrow{\gamma^{\mathrm{S}}}(D)$.

Identifying codes

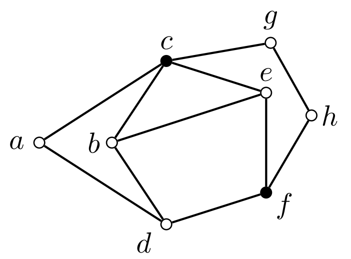
We are now ready to define the central notion of this thesis.

Definition 1.3 ([131])**.** *Given a (di)graph G, a subset $\mathcal{C}$ of $V(G)$ is an identifying code of G if $\mathcal{C}$ is both a dominating set and a separating code of G.*

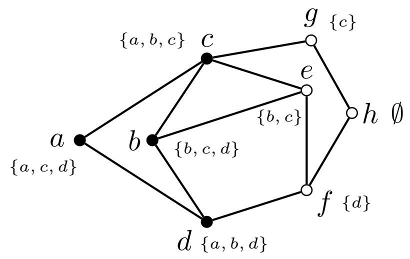
An example of a graph with an identifying code is represented in Figure 1.1(c), where the vertices are labelled and the sets $N[x] \cap \mathcal{C}$ are indicated. We point out that both the dominating set of Figure 1.1(a) and the separating code of Figure 1.1(b) are not identifying codes since in the first case, vertices d and f are not separated, and in the second case vertex h is not dominated.

The minimum size of an identifying code of a (di)graph G is called the *identifying code number* of G and is denoted $\gamma^{\mathrm{ID}}(G)$ if G is an undirected graph, and $\overrightarrow{\gamma^{\mathrm{ID}}}(G)$ if G is a digraph.

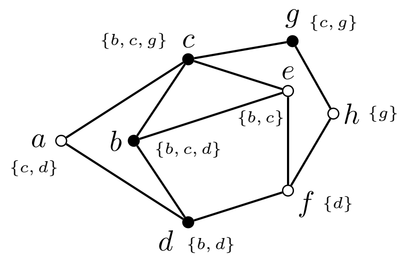
[3] We remark that one may as well choose to consider *out-neighbourhoods* and get a similar definition; but one may just reverse the direction of all arcs of the considered digraph to go from one to the other definition without changes.



(a) A dominating set.



(b) A separating code.



(c) An identifying code.

Figure 1.1: A graph with a dominating set, a separating code and an identifying code (black vertices).

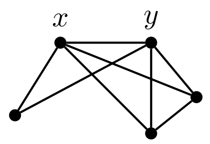
Identifying codes have been introduced in [131]; they have also been studied under the name of *differentiating-dominating sets* in [90, 110, 179].

1.1.2 First observations

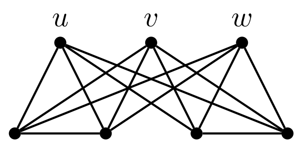
We now mention some preliminary facts about identifying codes.

Twins and identifiable graphs

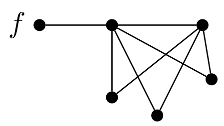
A fundamental remark when dealing with identifying codes is that not all graphs admit one: indeed, whenever two distinct vertices u, v are such that $N[u] = N[v]$, that is, $N[u] \ominus N[v] = \emptyset$, u and v cannot be separated. In that case, we say that u and v are *twins*. An example of two twins is depicted in Figure 1.2(a).



(a) A pair x, y of twins.



(b) Three mutually false twins u, v, w.



(c) A forced vertex f.

Figure 1.2: Twins, false twins and a forced vertex.

In a digraph, two vertices u, v are twins if they have the same *closed in-neighbourhood*. Observe that in that case, there must exist a pair of symmetric arcs between u and v. Hence, oriented graphs have no twins.

In fact, it is easily seen that being *twin-free*, i.e. having no pair of twins, is also a sufficient condition for a (di)graph to admit an identifying code:

Observation 1.4. *A (di)graph admits an identifying if and only if it is twin-free.*

Indeed, in a twin-free graph G, $\mathcal{C} = V(G)$ is an identifying code of G: it is certainly a dominating set, and for every vertex v, we have $N[v] \cap \mathcal{C} = N[v]$. Because of the twin-freeness of G, all these sets are distinct, and $\mathcal{C}$ is also a separating code of G.

In this thesis, a twin-free graph will be called *identifiable*. Twin-free graphs have been called *point-distinguishing* in [187] and have been studied for their own sake, see also [5, 8, 10, 42, 46].

False twins

A notion related to twins is the one of *false twins*: two distinct vertices u, v are false twins if $N(u) = N(v)$ but u, v are not adjacent. An example of three mutually false twins is given in Figure 1.2(b). The following observation is the reason of the importance of this notion in the area of identifying codes:

Observation 1.5. *Let G be an identifiable graph and let F be a set of vertices that are mutually false twins. Then for any identifying code $\mathcal{C}$ of G, at least $|F| - 1$ vertices of F belong to $\mathcal{C}$.*

Indeed, for any pair u, v of distinct vertices of F, we have $N[u] \ominus N[v] = \{u, v\}$ and hence at least one of u, v must belong to any identifying code of G.

Forced vertices

We also propose the notion of a *forced vertex* in a given identifiable graph G: vertex f is forced in G if there exist two vertices u, v such that $N[u] \ominus N[v] = \{f\}$. In other words, we have $N[v] = N[u] \cup \{f\}$ (or $N[u] = N[v] \cup \{f\}$). In the former case, we say that f is *uv-forced*; otherwise, f is *vu-forced*. In both cases, we say that f is *forced by the pair* $\{u, v\}$. An example of a forced vertex is depicted in Figure 1.2(c).

Observation 1.6. *Let G be an identifiable graph and let f be a forced vertex of G. Then f belongs to any identifying code of G.*

This fact holds since if f is uv-forced, f is the only vertex that can separate u, v.

Supersets of identifying codes are identifying codes

The following observation is easy to make, but is worth mentioning here.

Observation 1.7. *Let G be an identifiable graph and $\mathcal{C}$ an identifying code of G. Any set $\mathcal{C}'$ such that $\mathcal{C} \subseteq \mathcal{C}'$ is also an identifying code of G.*

The locality of identifying codes

The following observation gives an equivalent condition for a set to be an identifying code, and follows from the fact that for two vertices u, v at distance at least 3 from each other, $N[u] \ominus N[v] = N[u] \cup N[v]$.

Observation 1.8. *For a graph G and a set $\mathcal{C} \subseteq V(G)$, if $\mathcal{C}$ is a dominating set and $N[u] \cap \mathcal{C} \neq N[v] \cap \mathcal{C}$ holds for each pair of vertices u, v at distance at most 2 from each other, then $\mathcal{C}$ is an identifying code of G.*

Informally speaking, this shows that identifying codes have a "local" structure in the sense that a vertex from the identifying code does not influence vertices that lie arbitrarily far away from it.

Associated computational problems

Let us define the natural decision and optimization problems that are associated with identifying codes, and that will be studied in the second part of the thesis. Formal definitions about computational complexity will be given in Section 2.3.

IDENTIFYING CODE
INSTANCE: An identifiable graph G and an integer k.
QUESTION: Does G have an identifying code of size at most k?

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

1.2 Applications and motivations of identifying codes

In general, identification problems such as identifying codes or their generalizations and variants presented in Section 2.4 have a broad variety of applications in situations involving different variants of testing. For example, *test covers* (a notion that generalizes separating codes to hypergraphs and that will be defined in Chapter 2) can be used for the diagnosis of faults or diseases, biological identification of individuals according to their attributes, or pattern recognition [71, 156].

When we identify structures in a graph — as it is the case in identifying codes — the main applications are to consider the graph as a real-life structure. In this regard, graphs can for example model computer networks (each computer is a vertex and each edge is a network connection), spatial networks (each vertex is a location, and each edge is a road or gateway between two locations), social networks (each vertex is an individual, each edge is a social relationship between two of them, e.g. friendship) or molecules (vertices are atoms and edges are links between them). In general, a graph models any situation where we have a set of elements and a binary relation between them.

Identifying codes have been particularly applied to situations where one wants to detect failures in a computer network. Assume that we have an identifying code and each code vertex represents a failure detector that is able to detect a fault within its closed neighbourhood (when it does, we say it is in "alarm state"). Whenever there is one failure, by the domination property, at least one fault-detector will be in alarm state. Moreover, by the separation property, the set of detectors that are in alarm state precisely determines the vertex that is faulty. Assuming that we have a central monitoring system where we know which are the detectors in alarm state, this enables one to precisely locate the failure; see e.g. [131, 153] for more details. Observe that some variants, such as $(1, \leq \ell)$-identifying codes that will be described later, allow one to handle the case of several simultaneous failures.

A similar situation arises when the network is a complex of rooms and corridors, and detectors are e.g. fire alarms or motion sensors. In the same way as in the previous application, when the detectors are placed as an identifying code, they allow to detect and locate an intruder or a fire in the building. This idea has been explained in e.g. [168, 176] and a real experimental motion sensor system based on identifying codes has been implemented and discussed in [194].

Identifying codes have also been used in the setting of *routing problems* in networks. This problem, given two computers that are part of a network, is to send a message from one to the other (under certain constraints depending on the network). Usually, the message transits through specific computers of the network called *routers*. In some applications, these routers form a dominating set. The *naming problem* (i.e. the problem of giving a unique identifier to each member of the network) is also often addressed in computer network design. By using the fact that identifying codes are dominating sets and that they induce unique identifiers to each network node, both the routing and the naming problem can be solved using identifying codes and domination-based routing schemes [141].

Finally, identifying codes (among other domination-related parameters) have been used in the setting of comparing secondary RNA structures (viewing these molecules as graphs) [110]. Indeed, experimentations have shown that the values of domination parameters for RNA molecules help to give a good description of the molecular properties of these structures.

1.3 Overview and contributions of the thesis

Let us now give an overview of the results presented in this thesis, which is divided into two parts: combinatorial aspects of identifying codes, and algorithmic aspects of identifying codes. Before these technical parts, Chapter 2 gathers notations and useful definitions in graph theory, combinatorics and computational complexity. Therein, we also present variants of identification problems related to identifying codes, as well as results about identifying codes that are related to the results of thesis. Chapter 9 contains the general conclusion of this thesis. In Appendix A, we gather proofs that are repetitive, of minor interest or that come from the literature (but are not accessible). The bibliography is divided into two parts: general references (referenced using numbered citations) and references to the author's papers (referenced using alphabetical citations). We also provide an index of definitions and a list of notations.

1.3.1 Part I: combinatorial aspects

The first part of this thesis is devoted to the study of bounds on the identifying code number, and graphs that reach these bounds. It is divided into three chapters.

In Chapter 3, which is based on the papers [FGK+11, FNP12], we characterize graphs and digraphs that are extremal with respect to the known upper bounds on parameters γ^{ID} and $\overrightarrow{\gamma^{\text{ID}}}$. In particular, we give full characterizations for finite digraphs, infinite oriented graphs and infinite undirected graphs that have their whole vertex set as only identifying code. For both the finite directed and the infinite oriented case, we show that these graphs can be described as the closures of rooted top-down oriented trees. For the infinite undirected case, roughly speaking, these graphs can be built from disjoint copies of a specific infinite graph and complete joins between them. We also characterize finite undirected graphs that have their order minus one as minimum size of an identifying code. We show that these graphs are either stars or they can be built from a family of specific powers of paths using repeated complete join and disjoint union operations as well as the addition of a universal vertex. This work solves some open problems from the literature that had previously been studied in [36, 45, 96, 179].

Chapter 4 is based on the papers [FKKR12, FP12]. Therein, we discuss bounds on γ^{ID} in graphs of given maximum degree. We first characterize those graphs that reach the known lower bound $\gamma^{\text{ID}}(G) \geq \frac{2|V(G)|}{\Delta(G)+2}$ from the literature. We then propose the upper bound $\gamma^{\text{ID}}(G) \leq |V(G)| - \frac{|V(G)|}{\Delta(G)} + c$ (where c is an absolute constant) in the form of a conjecture (Conjecture 4.4). Discussing the tightness of our conjecture, we present several constructions of graphs that reach its bound, or are close to it. In support of the conjecture, we closely approximate its bound both for general graphs and for special graph classes such as triangle-free graphs or regular graphs. To do so, we mainly use two techniques to tackle the problem. The first technique consists in computing complements of special kinds of independent sets. We use it to prove that the bound $\gamma^{\text{ID}}(G) \leq |V(G)| - \frac{|V(G)|}{(1+o_{\Delta(G)})\Delta(G)}$ holds for triangle-free graphs. The second technique is to use the probabilistic method (more precisely, we use L. Lovász' well-celebrated Local Lemma together with Chernoff bounds). We use it to show that the bound $\gamma^{\text{ID}}(G) \leq |V(G)| - \frac{|V(G)|}{\Theta(\Delta(G)^3)}$ holds in general, and that $\gamma^{\text{ID}}(G) \leq |V(G)| - \frac{|V(G)|}{\Theta(\Delta(G))}$ holds (among other classes) for regular graphs or graphs of bounded clique number.

Finally, in Chapter 5 (which is based on the papers [FGN+12, FKM+12, FP12]), we consider identifying codes in various graph classes. We first give upper bounds on parameter γ^{ID} for graphs of girth 5 and given minimum degree. It turns out that such graphs have essentially much smaller identifying codes than when either one of the minimum degree or the girth condition are relaxed. We first use a technique based on a DFS spanning tree to show that graphs on n vertices having minimum degree at least 2 and girth at least 5 have an identifying code of size at most $\frac{7n}{8}$. We then show that one can use *2-dominating sets* to construct identifying codes in graphs of girth at least 5, and we use the probabilistic method to give upper bounds on the size of these constructions. This leads to bounds of the order $\gamma^{\text{ID}}(G) \leq (1 + o_{\delta(G)}(1))\frac{3\ln\delta(G)}{2\delta(G)}n$ for graphs G

with girth at least 5. We use these bounds to compute (with high probability) the identifying code number of a random regular graph. Then, we briefly discuss the class of interval graphs, providing a lower bound of the order $\gamma^{\mathrm{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ for the identifying code of an interval graph G, and the lower bound $\gamma^{\mathrm{ID}}(G) \geq \frac{|V(G)|+1}{2}$ when G is a *unit* interval graph. Finally, we study the class of line graphs, where we present some nontrivial lower and upper bounds by introducing the new concept of *edge-identifying codes*, a notion similar to identifying codes but which aims at identifying edges instead of vertices. This notion is equivalent to the one of (vertex-)identifying codes in line graphs. We first investigate basic properties of edge-identifying codes. We then show that $\frac{|V(G)|}{2} \leq \gamma^{\mathrm{EID}}(G) \leq 2|V(G)| - 3$, where $\gamma^{\mathrm{EID}}(G)$ is the minimum size of an edge-identifying code of G. The lower bound implies the lower bound $\gamma^{\mathrm{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ for any identifiable line graph G. The upper bound is obtained by showing that any minimal edge-identifying code induces a 2-degenerate graph. We also apply this bound to show that Conjecture 4.4 holds for dense enough line graphs.

1.3.2 Part II: algorithmic aspects

In the second part of the thesis, we prove some complexity results for Identifying Code and Min Id Code for various graph classes. This part is divided into three chapters.

In Chapter 6, we present four new AP-reductions which show that Min Id Code remains log-APX-complete even for bipartite, split, DSP and co-bipartite graphs, respectively. Three of these reductions are from Min Discrim Code. It was previously known that Min Id Code is log-APX-complete, but without any specific restriction on the graph class. The results of this chapter are new and solely the author's work, i.e. they have not been published elsewhere.

In Chapter 7, which is partly based on the papers [FGN+12, FKM+12] (the other results being new and solely the author's work), we show that Min Id Code is APX-complete for bipartite graphs of small maximum degree by reduction from Min Dom Set and Min Vertex Cover, for a subclass of split graphs, and for line graphs (for the two latter cases, we give reductions from restricted versions of Max SAT). Furthermore, our reductions imply that decision problem Identifying Code is NP-complete for even more restricted graph classes (i.e. chordal bipartite graphs, planar bipartite graphs of maximum degree 4 and planar perfect line graphs of maximum degree 4). We also prove that Identifying Code is NP-complete for interval graphs by a new reduction from 3-Dimensional Matching.

Finally, in Chapter 8 (which is based on the papers [FGN+12, FKM+12] but also contains new results that are solely the author's work), we relate Min Id Code for unit interval graphs to the new problem of covering the edges of a special graph (the *ladder graph*) by cycles from a given input set of cycles. We call this problem Min Ladder Cycle Cover. We then give a PTAS for Min Ladder Cycle Cover, implying a PTAS for Min Id Code for unit interval graphs. We also show that Edge-Identifying Code is linear-time solvable for graphs of bounded tree-width. This implies that Identifying Code is linear-time solvable for line graphs of graphs of bounded tree-width. In the previous chapters, we show that in some classes of graphs such as co-bipartite graphs or interval graphs, Dominating Set is in P but Identifying Code is NP-complete. In Chapter 8, we define the first known class of graphs for which Identifying Code is in P but Dominating Set is NP-complete, that we call the class of *SC-graphs*.

1.4 Other work done during the PhD

In this thesis, only the work that has been done in the field of "classic" identifying codes will be presented. I have also worked on other topics, resulting in the following publications or manuscripts which are the fruit of successful collaborations. Article [FLP12] gives a new lower bound for $(1, \leq 2)$-identifying codes in the infinite king grid. Article [FHL+12] deals with identifying colourings, that are proper vertex-colourings that enable to distinguish neighbouring vertices from each other. Article [FK12] deals with the new topic of identifying the vertices of a graph using paths. I have also worked on the topic of *arbitrarily partitionable graphs*, that is, graphs on n vertices for which, given any decomposition of n as a sum of integers $\lambda_1, \ldots, \lambda_k$, their vertex set can be partitioned into subsets $S_1, \ldots, S_k$ inducing connected subgraphs with $|S_i| = \lambda_i$ for each i from $\{1, \ldots, k\}$. This work has resulted in the two articles [BBFP12, BFPW12]. Finally, I have worked on the computational complexity of homomorphisms between *signed graphs* (that are specific 2-edge-coloured graphs with a particular *re-signing* operation defining an equivalence relation between signed graphs), resulting in the article [FN12].

1.5 Summary of known bounds on the identifying code number and complexity results for identifying codes

Most of the first part of this thesis deals with lower and upper bounds on parameter γ^{ID} in various graph classes. We try to give an overview of the currently known bounds in Tables 1.3, 1.4, 1.5, 1.6 and 1.7. Some of them are proved in this thesis (those marked in red), and the others are taken from the literature. Entries for which we have no information are marked “OPEN”. Each table contains graph classes that can be defined in a similar way: Table 1.3 contains bounds for classes of graphs that are defined using a set of forbidden induced subgraphs; Table 1.4 contains bounds for classes defined using a set of forbidden minors; Table 1.5 contains bounds for classes of intersection graphs; Table 1.6 contains bounds for graphs of given minimum degree; finally, Table 1.7 contains bounds for subclasses of graphs of maximum degree Δ . We believe that this presentation provides a good overview of the field; it might help us understanding some patterns in these bounds.

Known results about the computational complexity of problems Identifying Code and Min Id Code are summarized in Tables 1.8 and 1.9. In these tables, we compare these results to the known complexities of Dominating Set and Min Dom Set when restricted to the same graph classes, which are problems that have been well-studied for many years. Entries for which we have no information are marked “OPEN”. We give the reference from the literature when the result is already known, and from the thesis when the result is new. The information regarding the decision problems Identifying Code and Min Id Code is also summarized in the graph class inclusion diagram of Figure 1.10. The inclusions between graph classes of this diagram have been determined with the help of the online *Information System on Graph Classes and their Inclusions* (ISGCI) [73] (originally based on the book [31]). Here also, results that are in red are those that are proved in this thesis (or follow from our results).

graph class	Lower bound	Tightness	Upper bound	Tightness
in general	$\lceil \log_2(n+1) \rceil$ [131]	YES [154]	$n-1$ [22, 96]	YES Cor. 3.28
bipartite	$\lceil \log_2(n+1) \rceil$ [131]	YES Cor. 2.26	$n-1$ [22, 96]	YES Cor. 3.28
line	$\frac{3\sqrt{2}}{4}\sqrt{n}$ Cor. 5.25	YES Cor. 5.25	$n-1$ [22, 96]	YES Cor. 3.28
induced claw-free	$\Theta(\ln(n))$ [131]	YES Prop. 5.38	$n-1$ [22, 96]	YES Cor. 3.28
chordal	$\lceil \log_2(n+1) \rceil$ [131]	YES Cor. 2.26	$n-1$ [22, 96]	YES Cor. 3.28
interval	$\sqrt{2n+\frac{1}{4}}-\frac{1}{2}$ Cor. 5.9	YES Cor. 5.9	$n-1$ [22, 96]	YES Cor. 3.28
unit interval	$\frac{n+1}{2}$ Thm. 5.11	YES [23, 90]	$n-1$ [22, 96]	YES Cor. 3.28

Table 1.3: Known bounds on parameter γ^{ID} and their tightness in chosen graph classes defined by a set of forbidden induced subgraphs. Red entries are new results proved in this thesis.

graph class	Lower bound	Tightness	Upper bound	Tightness
in general	$\lceil \log_2(n+1) \rceil$ [131]	YES [154]	$n-1$ [22, 96]	YES Cor. 3.28
trees	$\frac{3(n+1)}{7}$ [24]	YES [24]	$n-1$ [22, 96]	YES Cor. 3.28
planar	$\frac{n+10}{7}$ [183]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28
series-parallel	$\frac{2n+3}{7}$ [183]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28
outerplanar	$\frac{2n+3}{7}$ [183]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28

Table 1.4: Known bounds on parameter γ^{ID} and their tightness in chosen graph classes defined by a set of forbidden minors. All results are from the literature.

graph class	Lower bound	Tightness	Upper bound	Tightness
in general	$\lceil \log_2(n+1) \rceil$ [131]	YES [154]	$n-1$ [22, 96]	YES Cor. 3.28
line	$\frac{3\sqrt{2}}{4}\sqrt{n}$ Cor. 5.25	YES Cor. 5.25	$n-1$ [22, 96]	YES Cor. 3.28
chordal	$\lceil \log_2(n+1) \rceil$ [131]	YES Cor. 2.26	$n-1$ [22, 96]	YES Cor. 3.28
undirected path	$\lceil \log_2(n+1) \rceil$ [131]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28
directed path	$\lceil \log_2(n+1) \rceil$ [131]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28
interval	$\sqrt{2n+\frac{1}{4}}-\frac{1}{2}$ Cor. 5.9	YES Cor. 5.9	$n-1$ [22, 96]	YES Cor. 3.28
unit interval	$\frac{n+1}{2}$ Thm. 5.11	YES [23, 90]	$n-1$ [22, 96]	YES Cor. 3.28
permutation	$\lceil \log_2(n+1) \rceil$ [131]	OPEN	$n-1$ [22, 96]	YES Cor. 3.28

Table 1.5: Known bounds on parameter γ^{ID} and their tightness in chosen intersection graph classes. Red entries are new results proved in this thesis.

graph class	Lower bound	Upper bound	Tightness
girth 5, $\delta \geq 2$	OPEN	$\frac{7n}{8}$ Thm. 5.1	OPEN
girth 5	OPEN	$\frac{3\left(\ln\delta+\ln\ln\delta+1+\frac{\ln\ln\delta}{\ln\delta}+\frac{1}{\ln\delta}\right)}{2\delta}$ Thm. 5.3	OPEN
girth 5 and avg. deg. $O_\delta(\delta(\ln\delta)^2)$	OPEN	$\frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n$ Thm. 5.3	YES (up to $\pm o\left(\frac{\ln\delta}{\delta}\right)$) Thm. 5.4

Table 1.6: Bounds on parameter γ^{ID} in graphs with minimum degree δ. Red entries are new results proved in this thesis.

graph class	Lower bound	Tightness	Upper bound	Tightness
max. deg. Δ	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{103\Delta(\Delta+1)^2}$ Cor. 4.51	OPEN
max. deg. Δ no forced vertices	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{103\Delta}$ Cor. 4.50	OPEN
Δ-regular	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{103\Delta}$ Cor. 4.50	OPEN
K_3-free, max. deg. Δ	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{\Delta+\frac{3\Delta}{\ln\Delta-1}}$ Cor. 4.40	OPEN
K_3-free, max. deg. Δ no false twins	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{\frac{3\Delta}{\ln\Delta-1}}$ Cor. 4.40	OPEN
bipartite, max. deg. Δ	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{\Delta+9}$ Cor. 4.42	OPEN
planar K_3-free max. deg. Δ	$\max\left\{\frac{2n}{\Delta+2}, \frac{n+10}{7}\right\}$ [131, 183]	OPEN	$n - \frac{n}{\Delta+9}$ Cor. 4.42	OPEN
K_{k+1}-free, max. deg. Δ	$\frac{2n}{\Delta+2}$ [131]	YES Cor. 4.2	$n - \frac{n}{103f(k)\Delta}$ Cor. 4.52	OPEN

Table 1.7: Bounds on parameter γ^{ID} in chosen subclasses of graphs with maximum degree Δ. Red entries are new results proved in this thesis.

graph class	IDENTIFYING CODE	DOMINATING SET
bipartite	NP-c [44]	NP-c [21, 35]
chordal bipartite	NP-c [Thm. 7.10]	NP-c [157]
(subcubic) planar	NP-c [9]	NP-c [88, 186]
planar (with max. degree 4, arb. girth)	NP-c [6]	NP-c [186]
planar bipartite max. degree 4	NP-c [Thm. 7.7]	NP-c [186]
subcubic planar bipartite	OPEN	NP-c [186]
line	NP-c [Cor. 7.37]	NP-c [201]
(planar bipartite) unit disk	NP-c [158]	NP-c [57]
bounded treewidth/cliquewidth	P [153]	P [3, 64]
line of bounded treewidth	P [Cor. 8.11]	P [3, 64]
split	NP-c [Cor. 6.8]	NP-c [21]
split of bounded maximum CS-degree	NP-c [Thm. 7.19]	NP-c [21]
undirected path	NP-c [Thm. 7.50]	NP-c [29]
directed path	NP-c [Thm. 7.50]	P [29]
interval	NP-c [Thm. 7.50]	P [29]
unit interval	OPEN	P [29]
permutation	OPEN	P [83]
DSP	NP-c [Cor. 6.11]	P [136]
AT-free	NP-c [Cor. 6.14]	P [136]
co-bipartite	NP-c [Cor. 6.14]	P [136]
(planar) SC-graphs	P [Cor. 8.14]	NP-c [Thm. 8.15]

Table 1.8: Comparison of complexities of decision problems IDENTIFYING CODE and DOMINATING SET for selected graph classes. Red entries are new results proved in this thesis. The abbreviation "NP-c" stands for "NP-complete".

graph class	Min Id Code lower bound	Min Id Code upper bound	Min Dom Set lower bound	Min Dom Set upper bound
in general	log-APX-hard [20, 185]	$O(\ln(n))$ [20, 94, 185]	log-APX-hard [169]	$1+\ln(n)$ [127]
bipartite	log-APX-hard Thm. 6.5	$O(\ln(n))$ [20, 94, 185]	log-APX-hard [53, 169]	$1+\ln(n)$ [127]
split (also: chordal)	log-APX-hard Thm. 6.8	$O(\ln(n))$ [20, 94, 185]	log-APX-hard [53, 169]	$1+\ln(n)$ [127]
split of max. CS-degree Δ	APX-hard $\Delta \geq 5$: Thm. 7.19	$O(\ln(\Delta))$ Thm. 7.12	APX-hard [21]	$1+\ln(\Delta)$ [127]
planar (∗)	NP-hard [9]	7 Cor. 2.53	NP-hard [88]	PTAS [15]
line	APX-hard Thm. 7.38	4 Cor. 7.21	APX-hard [52, 201]	2 [201]
induced $K_{1,\ell}$-free ($\ell \geq 3$)	log-APX-hard Thm. 6.14	$O(\ln(n))$ [20, 94, 185]	APX-hard [52, 201]	$\ell - 1$ [53]
max. degree Δ	APX-hard $\Delta \geq 3$: [9]	$O(\ln(\Delta))$ [94, 185]	APX-hard $\Delta \geq 3$: [162]	$1+\ln\Delta$ [127]
max. degree Δ and bipartite	APX-hard $\Delta \geq 4$: Thm. 7.8	$O(\ln(\Delta))$ [94, 185]	APX-hard $\Delta \geq 3$: [53]	$1+\ln\Delta$ [127]
unit disk (∗)	NP-hard [158]	$O(\ln(n))$ [20, 94, 185]	NP-hard [57]	PTAS [124]
co-bipartite (also: DSP, AT-free)	log-APX-hard Thm. 6.14	$O(\ln(n))$ [20, 94, 185]	P [136]	
strongly chordal (∗)	NP-hard Thm. 7.50	$O(\ln(n))$ [20, 94, 185]	P [82]	
undirected path (∗)	NP-hard Thm. 7.50	$O(\ln(n))$ [20, 94, 185]	APX-hard [29]	$1+\ln(n)$ [127]
directed path (∗)	NP-hard Thm. 7.50	$O(\ln(n))$ [20, 94, 185]	P [29]	
interval (∗)	NP-hard Thm. 7.50	$O(\ln(n))$ [20, 94, 185]	P [29]	
unit interval (∗)	OPEN	PTAS Cor. 8.8	P [29]	
permutation (∗)	OPEN	$O(\ln(n))$ [20, 94, 185]	P [83]	

Table 1.9: Comparison of complexities and approximation ratios of optimization problems Min Id Code and Min Dom Set for selected graph classes. Red entries are new results proved in this thesis. Graph classes for which the precise complexity class of Min Id Code is not fully determined are marked with (∗). For all these classes, the exact complexity class for Min Dom Set is known, except (up to our knowledge) for undirected path graphs.

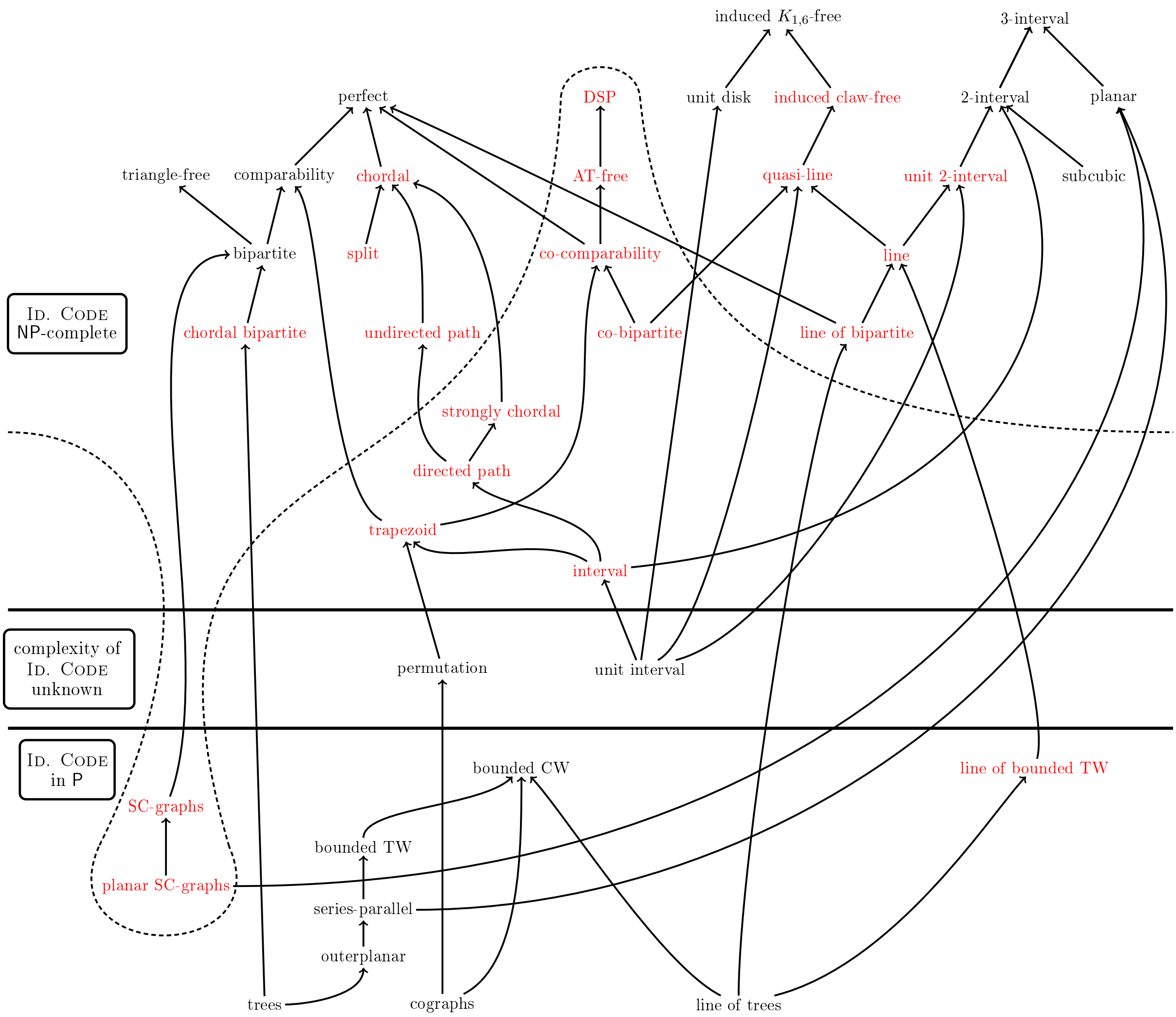


Figure 1.10: Inclusion diagram of selected graph classes with known complexity of decision problem Identifying Code, divided into three categories by the two horizontal full lines. For a comparison, the dashed curve indicates the separation between classes for which Dominating Set is NP-complete and classes where it is in P. Classes in red are those for which the complexity was unknown prior to the results of this thesis. An arrow from a class B to a class A means that A contains B. Abbreviations TW and CW stand for tree-width and clique-width, respectively. Definitions of graph classes that are not defined in this dissertation can be found in [31, 73].

Chapter 2
Definitions, notations and related work

IN this chapter, we first recall a set of basic notions in mathematics, graph theory, combinatorics and computational complexity theory, which constitute the framework of the studies of this thesis (Sections 2.1, 2.2 and 2.3). We then present various identification problems that are related to identifying codes in Section 2.4. Finally, in Section 2.5, we survey some of the known results of the literature that are relevant to this thesis.

2.1 A few mathematical notations

Let us first fix a small number of general mathematical notations that will be used in this thesis. Notations related to other notions will be introduced in the corresponding parts of this introduction.

We will use the standard *asymptotic notations* $o, \omega, O, \Omega, \Theta$ throughout the thesis. Given two non-zero functions $f, g : \mathbb{R} \to \mathbb{R}$ or $f, g : \mathbb{N} \to \mathbb{N}$, we note $f(x) = o(g(x))$ $\lim_{x\to+\infty} \frac{f(x)}{g(x)} = 0$ and $f(x) = \omega(g(x))$ if $\lim_{x\to+\infty} \frac{f(x)}{g(x)} = +\infty$ (i.e. when $g(x) = o(f(x))$). We note $f(x) = O(g(x))$ if there exists an $x_0 \in \mathbb{R}$ and a positive constant c such that $f(x) \leq c \cdot g(x)$ when $x \geq x_0$. Similarly, $f(x) = \Omega(g(x))$ if there exists an $x_0 \in \mathbb{R}$ and a positive constant c such that $f(x) \geq c \cdot g(x)$ when $x \geq x_0$. When $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$, we note $f(x) = \Theta(g(x))$.

We also use a (non-standard) modified version of these notations when functions f, g have multiple variables, i.e. the notations $o_n, \omega_n, O_n, \Omega_n, \Theta_n$. They indicate that variable n is the variable to be considered in the asymptotics in the previous definitions.

Given two sets A and B, we will denote their symmetric difference by $A \ominus B$.

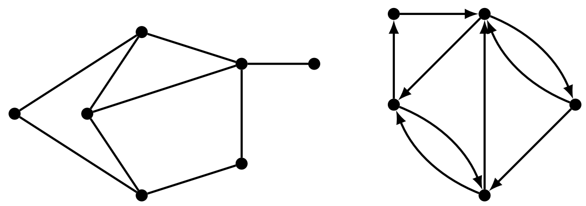
When speaking about probabilities, we denote the proability of a given event A by $Pr(A)$. The expectance of a given random variable X is denoted by $\mathbb{E}(X)$.

2.2 Graphs and hypergraphs

2.2.1 Basic definitions

Undirected graphs

An (undirected) graph G is composed of a set $V(G)$ of elements called *vertices*, together with a symmetric binary relation $E(G)$ over $V(G)$ called *adjacency relation*. The set $V(G)$ is called the *vertex set* of G. Relation $E(G)$ can be represented by a multiset of non-ordered pairs of $V(G)$, called *edges* of G. Set $E(G)$ is called the *edge set* of G. The cardinality of the vertex set is the *order* of G, and the cardinality of the edge set is the *size* of G. We sometimes call a pair of vertices that is not in $E(G)$ a *non-edge* of G. The two vertices of an edge are called its *endpoints*. Two vertices u, v of $V(G)$ (or two edges e, f of $E(G)$) are said to be *adjacent* if $\{u, v\} \in E(G)$ ($e \neq f$ and $e \cap f \neq \emptyset$, respectively). Adjacency between two vertices or two edges u, v is denoted $u \sim v$; non-adjacency is denoted $u \nsim v$. Sometimes, to shorten our notations, we will denote an edge $\{u, v\}$ by uv. A vertex u and an edge e are *incident* if $u \in e$. A *loop* is an edge whose two endpoints are the same vertex. An edge that exists twice in a graph is called *multiple*. A graph is called *simple* if it does neither have loops nor multiple edges. In the following, by *graph* and unless otherwise stated, we will mean undirected simple graph over a finite vertex set. By *multigraph*, we mean undirected graph over a finite vertex set (i.e. multiple edges and loops are allowed). An example of a graph is represented in Figure 2.1(a), where black dots represent vertices and lines represent edges.



(a) An undirected graph. (b) A digraph.

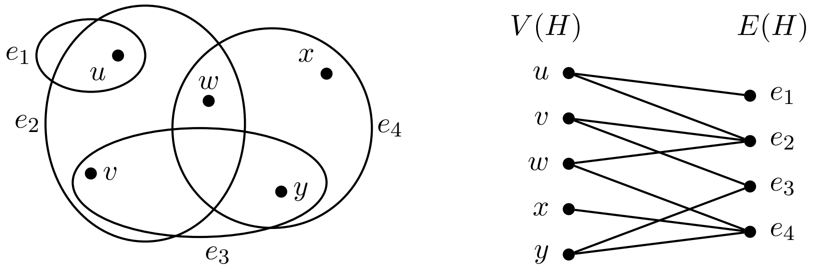
Figure 2.1: An undirected and a directed graph.

Directed graphs

The concept of a graph can be generalized to the one of a *directed graph* (commonly called *digraph*). As in a graph, in a digraph D we also have a set $V(D)$ of vertices, but instead of a symmetric adjacency relation E we have a (not necessarily symmetric) binary relation $A(D)$ over $V(D)$. The elements of $A(D)$ are called *arcs* and $A(D)$ is called the *arc set* of D: $A(D)$ can be seen as a set of ordered pairs from $V(D)$. We will usually denote an arc (u, v) by $\overrightarrow{uv}$. An undirected graph can be seen as a digraph in which the existence of an arc $\overrightarrow{uv}$ implies the existence of the symmetric arc $\overrightarrow{vu}$. Such a digraph is called *symmetric*. A digraph is called an *oriented graph* if it does not contain any pair of symmetric arcs. The *underlying graph* of a digraph D is the undirected graph over the same set of vertices as D, in which the edge uv is present whenever at least one of the arcs $\overrightarrow{uv}$ and $\overrightarrow{vu}$ exists in D. An example of a digraph is represented in Figure 2.1(b), where dots represent vertices and arrows represent arcs.

Hypergraphs

Another widely studied generalization of the concept of a graph is the one of a *hypergraph*. A hypergraph H is also formed by a vertex set $V(H)$, but now the edges of H (also called *hyperedges*) are subsets of $V(H)$ of any cardinality. The set of hyperedges of H is denoted $E(H)$. A hypergraph can also be viewed as a *set system*, i.e. a collection of subsets (the hyperedges) of a set (the vertices). The concepts of adjacency and incidence carry over to hypergraphs: two vertices are adjacent if they both belong to some hyperedge, and a vertex v is incident to some hyperedge e from $E(H)$ if $v \in e$. A hypergraph H is said to be *uniform* if all hyperedges have the same cardinality m — in this case, we can say that H is *m-uniform*. A simple undirected graph is then a 2-uniform hypergraph. An example of two representations of the same hypergraph H with $V(H) = \{u, v, w, x, y\}$ and $E(H) = \{e_1 = \{u\}, e_2 = \{u, v, w\}, e_3 = \{v, y\}, e_4 = \{w, x, y\}\}$ is given in Figure 2.2. In Figure 2.2(a), dots represent vertices and ellipses represent hyperedges; Figure 2.2(b) represents the *bipartite incidence graph* of H (the bipartite incidence graph of a hypergraph H is the graph $\mathcal{B}(V(H), E(H))$ with vertex set $V(H) \cup E(H)$, with an edge between vertex v and edge e if and only if $v \in e$).



(a) Graphical representation of H. (b) Bipartite incidence graph of H.

Figure 2.2: Two representations of hypergraph H.

Isomorphisms

Two graphs G and H are said to be *isomorphic* if there exists a bijection between $V(G)$ and $V(H)$ which preserves the adjacency relation, i.e. two vertices of G are adjacent if and only if their images in H are adjacent. We will denote that G and H are isomorphic by $G \cong H$; if G and H are *not* isomorphic, we note $G \ncong H$.

Neighbourhoods and degree parameters

In an undirected graph G, given a vertex v, the vertices which are adjacent to v are called *neighbours* of v. The set of neighbours of v is called the *(open) neighbourhood* of v and is denoted $N(v)$. The *closed neighbourhood* $N[v]$ of v is the set $N(v) \cup \{v\}$. We generalize the notion of open and closed neighbourhoods of single vertices to the one of sets of vertices: given a set X of vertices, we let $N(X) = \cup_{x \in X} N(x)$ and $N[X] = \cup_{x \in X} N[x]$. The *degree* of v is the number of its neighbours, and is denoted $deg(v)$. The *minimum degree* of G (*maximum degree* of G, respectively) is the minimum value (maximum value, respectively) of the degree of a vertex among all vertices in G, and is denoted $\delta(G)$ ($\Delta(G)$, respectively). Similarly, the *average degree* of G, denoted $\overline{d}(G)$, is the average value of the degrees of all vertices of G. If all vertices of G have the same degree k, then G is said to be *regular*, or, more precisely, k-regular. A 3-regular graph is called *cubic*, and a graph with maximum degree 3 is *subcubic*. A vertex x is said to be *universal* in G if $N[x] = V(G)$, that is, all other vertices are neighbours of x. More generally, given a set S of vertices of G, we call vertex x *S-universal* if $S \subseteq N(x)$. On the contrary, vertex x is *isolated* if it has no neighbour.

For a digraph D and a vertex v of D, we distinguish the *in-neighbours* (*out-neighbours*) of v, that is, the vertices u such that $(u, v) \in A(D)$ ($(v, u) \in A(D)$, respectively). The *in-neighbourhood* of v (*out-neighbourhood* of v, respectively) is the set of its in-neighbours (out-neighbours, respectively); it is denoted $N^-(v)$ ($N^+(v)$, respectively), and its cardinality is the *in-degree* of v, denoted $d^-(v)$ (*out-degree* of v denoted $d^+(v)$, respectively). We also have the notions of *closed in-neighbourhood* $N^-[v]$ of v, and *closed out-neighbourhood* $N^+[v]$ of v. A vertex with no incoming arc is called a *source*, and a vertex with no outgoing arc is called a *sink*.

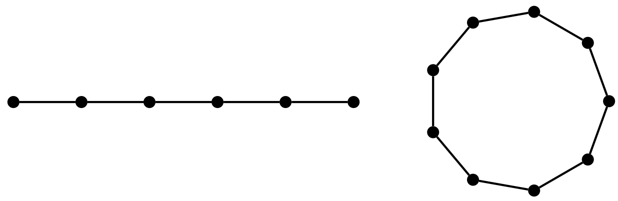
Paths and distance

In an undirected graph G, a *path* between two vertices u and v is a sequence $P = (u, \ldots, v)$ of at least two distinct vertices such that two consecutive vertices are adjacent in G. In this case we say that u and v are *joined* by path P. The *length* of a path is its cardinality minus one, i.e. the number of its edges. The *distance* between u and v is the smallest length of a path joining u and v, and is denoted $d(u, v)$.

We define the *distance-k closed neighbourhood* of some vertex x, denoted $N_k[x]$, to be the set of vertices that are at distance at most k from x in G.

In a digraph D, a *directed path* between two vertices u and v is a sequence $P = (u, \ldots, v)$ of at least two distinct vertices such that for each two consecutive vertices x, y of P, $(x, y) \in A(D)$.

The *path graph* (usually simply called *path*) P_n is the graph which consists only of a path of n vertices. The path P_6 is depicted in Figure 2.3(a).



(a) The path P_6. (b) The cycle C_9.

Figure 2.3: A path and a cycle.

Cycles and girth

In an undirected graph, a *cycle* is a path whose first and last vertices u, v are adjacent. Similarly, in a digraph, a *directed cycle* is a directed path whose first and last vertices u, v are joined by an arc $\overrightarrow{vu}$. The *length* of a cycle is the number of its vertices. The *girth* of an undirected graph G is the length of one of its shortest cycles, and is denoted $g(G)$. A cycle having an even (odd, respectively) number of vertices is called an *even cycle* (*odd cycle*, respectively).

The *cycle graph* (usually simply called *cycle*) C_n is the graph which consists of a unique cycle on n vertices. The cycle C_9 is depicted in Figure 2.3(b).

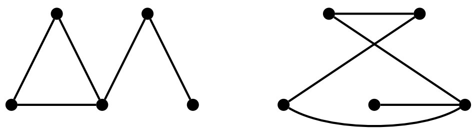
Connected graphs

A graph G is said to be *connected* if there exists a path between each pair of vertices in G. If G is not connected, its maximal connected subgraphs are called *connected components*.

2.2.2 Operations, transformations and substructures for graphs and hypergraphs

Complement of a graph

The *complement* of an undirected graph G is the graph denoted $\overline{G}$ having $V(G)$ as its vertex set and all non-edges of G as its edge set. An example of a graph and its complement are given in Figure 2.4.



G $\overline{G}$

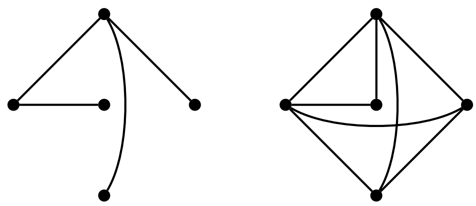
Figure 2.4: A graph and its complement.

Power of a graph

For an integer $r \geq 1$, the r^{th} *power* of a graph G, denoted G^r, is the graph with vertex set $V(G^r) = V(G)$ and where $\{u, v\} \in E(G^r)$ if and only if $d_G(u, v) \leq r$. Conversely, G is called an r^{th} *root* of G^r. A graph G and its 2^{nd} power are depicted in Figure 2.5.

Disjoint union of two (di)graphs

Given two disjoint graphs G_1 and G_2, the *disjoint union* of G_1 and G_2, denoted $G_1 \oplus G_2$, is the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The same concept carries over to digraphs.

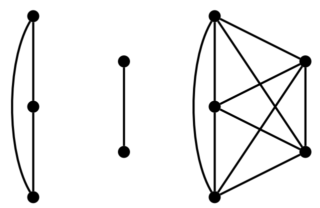


G G^2

Figure 2.5: A graph and its 2^{nd} power.

Complete join of two graphs

Given two disjoint graphs G_1 and G_2, the *complete join* of G_1 and G_2, denoted $G_1 \bowtie G_2$, is the disjoint union of G_1 and G_2 where we add all possible edges between G_1 and G_2: it has vertex set $V(G_1) \cup V(G_2)$ and its edge set contains $E(G_1)$, $E(G_2)$ and all edges $\{u_1, u_2\}$ with $u_1 \in V(G_1)$ and $u_2 \in V(G_2)$. Two graphs G_1, G_2 and their complete join are shown in Figure 2.6.



G_1 G_2 $G_1 \bowtie G_2$

Figure 2.6: Two graphs and their complete join.

Neighbourhood hypergraph of a (di)graph

Given a graph G, the *open/closed neighbourhood hypergraph* of G is the hypergraph with vertex set $V(G)$ and whose edges are all closed/open neighbourhoods in G. The open/closed in- or out-neighbourhood hypergraphs of a digraph D are defined similarly.

Subgraphs of a graph

Given a graph G, a graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H)$ is a subset of edges of $E(G)$ with both endpoints in $V(H)$.

Given a subset V' of $V(G)$, the subgraph *induced* by V', denoted $G[V']$, is the subgraph of G with vertex set V' and having as its edge set all edges of G with both endpoints in V'. Similarly, given a subset E' of $E(G)$, the subgraph *induced* by E', denoted $G[E']$, is the subgraph of G with vertex set $\bigcup_{e \in E'} e$ and edge set E'.

Given a vertex x or a set X of vertices, the subgraphs $G[V(G) - x]$ and $G[V(G) - X]$ are denoted $G - x$ and $G - X$, respectively.

A *spanning subgraph* of a graph G is a subgraph of G which has the same vertex set as G.

Minors of a graph

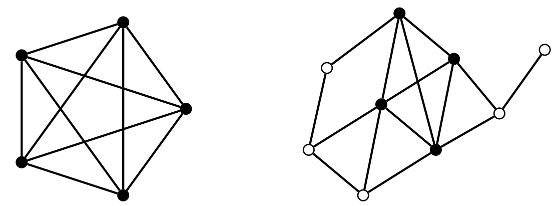
Given two graphs G and H, H is said to be a *minor* of G if one can obtain H from G using the following operations: deleting an edge of G, contracting an edge of G,[1] and deleting an isolated vertex of G.

[1] The contraction of an edge $\{u, v\}$ corresponds to the removal of edge $\{u, v\}$ and vertex u from G, and each other edge $\{u, x\}$ is replaced by $\{v, x\}$.

Cliques and complete graphs

The *complete graph* on n vertices, denoted K_n, is the graph having the set of all pairs of vertices as its edge set. The graph K_5 is depicted in Figure 2.7(a). We also denote by K_n^- the graph obtained from K_n by removing one edge.

A *clique* in a graph G is a subset of vertices inducing a complete subgraph in G. A graph with a clique of four vertices is given in Figure 2.7(b). A clique on three vertices is called a *triangle*. The *clique number* of G, denoted $\omega(G)$, is the maximum size of a clique in G. A graph having clique number at most 2 is called *triangle-free*.



(a) The graph K_5.

(b) A graph with a clique (black vertices).

Figure 2.7: A complete graph and a clique.

Independent sets

An *independent set* of a graph G (sometimes called *stable set*) is a set $\mathcal{S}$ of vertices inducing an edgeless graph, that is, $G[\mathcal{S}] \cong \overline{K_{|\mathcal{S}|}}$. An example is given in Figure 2.8 (white vertices). The *independence number* $\alpha(G)$ of G is the maximum size of an independent set in G. Note that $\alpha(G) = \omega(\overline{G})$. This notion can be generalized to the one of a *distance-k-independent set* $\mathcal{S}$, where it is asked that the distance between every two vertices of $\mathcal{S}$ is at least k. An independent set is then a distance-2-independent set, and the maximum size of a distance-k-independent set, $\alpha_k(G)$, is equal to $\alpha(G^{k-1})$.

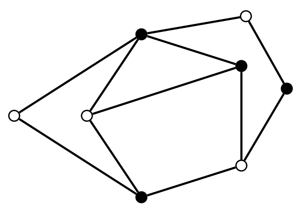


Figure 2.8: A graph with a vertex cover (black vertices) and an independent set (white vertices).

Hypergraph transversals and set covers

Given a hypergraph H, a *transversal* of H is a subset $\mathcal{T}$ of $V(H)$ such that for each edge e of $E(H)$, $e \cap \mathcal{T} \neq \emptyset$. A transversal is sometimes called a *hitting set* when the hypergraph is viewed as a set system $(V(H), E(H))$. Finding small transversals in hypergraphs is a well-studied problem in combinatorics (see e.g. [25]).

The related notion of a *set cover* of a given set system (or hypergraph) $(X, \mathcal{S})$ is defined as a set $\mathcal{C}$ of $\mathcal{S}$ such that each element of X belongs to at least one set of $\mathcal{C}$ [88, 127]. In fact, a set cover of a set system $(X, \mathcal{S})$ is exactly a transversal of the *dual* of the hypergraph $(X, \mathcal{S})$.[2]

[2]The dual of a hypergraph H is the hypergraph H' where vertex set and edge set are swapped, that is, each vertex of H' corresponds to a hyperedge of H, each hyperedge of H' corresponds to a vertex of H, and each hyperedge $e_{H'} = v_H$ of H' contains all vertices of H' (hyperedges of H) which are incident to v_H in H.

Vertex covers

A *vertex cover* of a graph G is a subset $\mathcal{C}$ of $V(G)$ such that for each edge e, $e \cap \mathcal{C} \neq \emptyset$. An example is given in Figure 2.8 (black vertices). The minimum size of a vertex cover in G, the *vertex cover number* of G, is denoted $\tau(G)$. Equivalently, a vertex cover is a transversal of G (when G is viewed as a 2-uniform hypergraph).

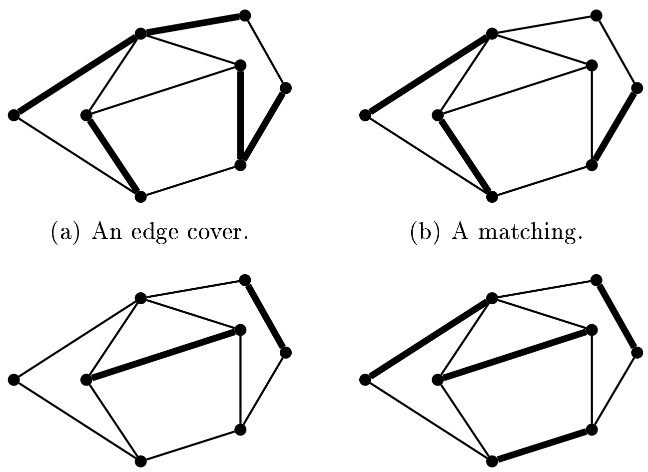
We remark that any vertex cover is the complement of an independent set (and vice-versa); hence, we have $\alpha(G) + \tau(G) = |V(G)|$ for any graph G.

Proper colourings

A *proper (vertex-)k-colouring* of a graph G is a function $c : V(G) \to \{1, \ldots, k\}$ such that for any two adjacent vertices u, v, we have $c(u) \neq c(v)$. Equivalently, a k-colouring is a partition of $V(G)$ into k independent sets. A graph having a k-colouring is called *k-colourable*. The minimum k such that G is k-colourable is called the *chromatic number* of G and is denoted $\chi(G)$.

Matchings and edge covers

An *edge cover* of a graph G is a subset $\mathcal{S}$ of its edges such that the union of the endpoints of $\mathcal{S}$ equals $V(G)$. A *matching* of G is a set of pairwise non-adjacent edges of G; an *induced matching* is a matching whose edges induce a graph of maximum degree 1; a *perfect matching* is a matching which is also an edge cover. Note that in order to admit a perfect matching, a graph must have an even order. A graph G with an edge cover and different types of matchings is shown in Figure 2.9.



(a) An edge cover. (b) A matching.

(c) An induced matching. (d) A perfect matching.

Figure 2.9: A graph with matchings and an edge cover (thick edges).

Locating-dominating sets

The concept of a *locating-dominating set* (sometimes called locating-dominating *code*) was introduced in [182] and has a very rich literature. A subset $\mathcal{D}$ of vertices of G is a locating-dominating set if it is a dominating set and if for every pair u, v of distinct vertices of $V(G) \setminus \mathcal{D}$, the property $N(u) \cap \mathcal{D} \neq N(v) \cap \mathcal{D}$ holds.

It is easily observed that any identifying code is also a locating-dominating set, though the converse is not necessarily true. As a matter of fact, each graph admits a locating-dominating set (e.g. the whole vertex set).

An example is shown in Figure 2.10, where the vertices are labelled and the sets $N(x) \cap \mathcal{D}$ are indicated for vertices that are not in the dominating set. The *location-domination number* of a graph G is the minimum size of a locating-dominating set in G and is denoted $\gamma^{\text{LD}}(G)$. Since any locating-dominating set is a dominating set, we have $\gamma(G) \leq \gamma^{\mathrm{LD}}(G)$. The fact that any identifying code is a locating-dominating set implies that $\gamma^{\mathrm{LD}}(G) \leq \gamma^{\mathrm{ID}}(G)$.

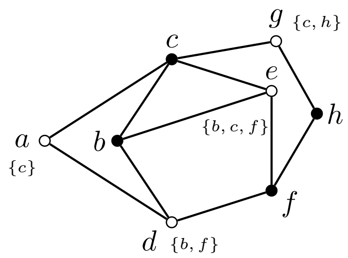


Figure 2.10: A graph with a locating-dominating set (black vertices).

2.2.3 Graph classes

Graphs with forbidden substructures

Given a (not necessarily finite) set $\mathcal{F}$ of graphs, we say that a graph G is *$\mathcal{F}$-free* if it does not contain any of the graphs of $\mathcal{F}$ as a subgraph. Similarly, G is *induced $\mathcal{F}$-free* if it does not contain any of the graphs of $\mathcal{F}$ as an *induced* subgraph, and G is *$\mathcal{F}$-minor-free* if it does not contain any of the graphs of $\mathcal{F}$ as a minor.

If the forbidden set $\mathcal{F}$ is composed of one single graph F, we say that G is F-free (induced F-free and F-minor-free, respectively).

Many important graph classes can be defined by means of such sets of forbidden substructures. For example, graphs of clique number at most k are exactly the K_{k+1}-free graphs; forests are cycle-free; graphs of maximum degree Δ are $K_{1,\Delta+1}$-free. We will see below that perfect graphs, chordal graphs, line graphs and other classes can be defined by lists of forbidden induced subgraphs; trees, planar graphs, outerplanar graphs, series-parallel graphs, can be defined by a list of forbidden minors. For a survey on the theory of minors, see [145]. For a survey on the case of induced subgraphs, see [56].

A graph class $\mathcal{G}$ is called *minor-closed* if for any graph G of $\mathcal{G}$ and any minor H of G, H belongs to $\mathcal{G}$. An important theorem of N. Robertson and P. Seymour (proved in a series of about twenty papers whose publication spans over about twenty years [172, 173]) connects minor-closed classes to classes defined by a list of forbidden minors. Indeed, they show that any minor-closed class of graphs $\mathcal{G}$ is the class of $\mathcal{F}(\mathcal{G})$-minor-free graphs, for some finite list of graphs $\mathcal{F}(\mathcal{G})$.[3]

Bipartite graphs, forests, trees

A graph is *bipartite* if its vertex set can be partitioned into two sets both inducing an independent set. An example of a bipartite graph was already given in Figure 2.2(b), and two further examples can be seen in Figure 2.11.

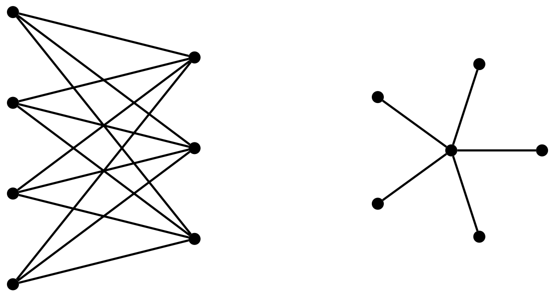
As an extremal case, the *complete bipartite graph* with parts of sizes n and m, denoted $K_{n,m}$, is built from two independent sets A and B of sizes n and m, with its edge set containing all pairs $\{a, b\}$ with $a \in A$ and $b \in B$. See Figure 2.11(a) for a drawing of $K_{4,3}$.

It is known that a graph is bipartite if and only if it has no odd cycle as a subgraph, see e.g. [75, Proposition 1.6.1]. A graph is called *chordal bipartite* if it is bipartite and it has no induced cycle of length more than 4.

An important theorem related to bipartite graphs, that will be used several times in this thesis, is Hall's marriage theorem:

Theorem 2.1 ([103]). *A bipartite (multi)graph G with parts A and B admits a perfect matching if and only if for every subset of A, we have $|A| \leq |N(A)|$.*

[3]We refer to [75, Chapter 12] for an overview of this result and its proof. In fact, this theorem is a corollary of a stronger result that we do not detail here.



(a) The graph $K_{4,3}$. (b) The star $K_{1,5}$.

Figure 2.11: Two bipartite graphs.

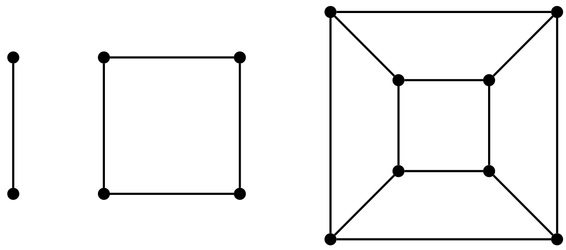
An important subclass of bipartite graphs is the one of *forests*: a graph is a forest if it does not contain any cycle as a subgraph. Equivalently, one can easily see that forests are exactly the K_3-minor-free graphs. A forest is a *tree* if it is connected. An example of a tree is the graph G from Figure 2.5. The vertices of degree 1 of a tree are called its *leaves*. Special trees are the *star* on n vertices, $K_{1,n-1}$ (as an example, the star $K_{1,5}$ is depicted in Figure 2.11(b)) and the previously defined path.

Given a connected graph G, a *spanning tree* of G is a spanning subgraph of G that is also a tree.

A *rooted tree* is a tree with one of its vertices marked as its *root*. A rooted tree T with root r can be decomposed into *levels* according to the distance to r: each level ℓ consists of all vertices having the same distance d_ℓ to r, and the levels are numbered according d_ℓ. The *height* of T is the number of its levels minus one. Given a vertex u of T at level ℓ, a *child* of u is a neighbour v of u at level $\ell + 1$; u is called the *parent* of v. More generally, if two vertices u, v have levels $\ell_u < \ell_v$, u is called an *ancestor* of v in T, and v is called a *descendant* of u in T.

A *rooted oriented tree* is an oriented graph whose underlying tree is rooted, and all arcs are oriented away from the root.

Another important subclass of bipartite graphs is the one of *hypercubes*. For any integer $d \geq 1$, the hypercube of dimension d, denoted $\mathcal{H}_d$, is the graph obtained from the cartesian product of $\mathcal{H}_{d-1}$ with K_2 (where $\mathcal{H}_1$ is isomorphic to K_2 itself). Alternatively, $\mathcal{H}_d$ can be seen as the graph whose vertices are elements of $\mathbb{Z}_2^d$ with two vertices being adjacent if their difference belongs to the standard basis $\{(1,0,0,\dots,0),(0,1,0,\dots,0),\dots,(0,0,\dots,0,1)\}$. The first three hypercubes are depicted in Figure 2.12.



$\mathcal{H}_1$ $\mathcal{H}_2$ $\mathcal{H}_3$

Figure 2.12: The three hypercubes $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$.

Another graph of importance is the *grid* $G_{a,b}$ which is isomorphic to the cartesian product of the paths P_a and P_b.

Planar graphs, series-parallel graphs, outerplanar graphs, genus of a graph

A graph G is called *planar* if it is possible to draw it on the plane such that no edges intersect each other. Such a drawing is called an *embedding* of G in the plane. Given an embedding of G, a *face* of G is defined by a closed region of the plane delimited by a set of edges of G. If G can be drawn in such a way that all vertices lie on the same face, G is said to be *outerplanar*. The classes of planar and outerplanar graphs can be defined by a set of forbidden minors. Indeed, a famous theorem of K. Wagner states that the class of planar graphs is exactly the class of $\{K_5, K_{3,3}\}$-minor-free graphs. Similarly, the class of outerplanar graphs is exactly the class of $\{K_4, K_{2,3}\}$-minor-free graphs (see [75, Exercise 4.20]). There also exists the intermediate class of *series-parallel graphs*, that is, the class of K_4-minor-free graphs.[4]

For some examples, note that the graph from Figure 2.1(a) and graph G^2 from Figure 2.5 are planar, but not series-parallel (they both contain a K_4-minor), whereas the graphs from Figure 2.5 are outerplanar. The graph K_5 from Figure 2.7(a) is not planar.

We also point out that planarity can be generalized as follows. A graph is said to be of *genus* at most g if it can be embedded on a surface of genus g without edge crossings. We recall that the plane has genus 0 and the torus has genus 1. We refer the reader to [75] for more details.

Intersection graphs

Let $\mathcal{S}$ be a collection of sets (they can be of any type). The *intersection graph* of $\mathcal{S}$ is the graph with vertex set $\mathcal{S}$ and where S_1 and S_2 are adjacent if and only if $S_1 \cap S_1 \neq \emptyset$. This notion is an important one, as many special cases are of particular interest and have been studied extensively. We present some classes that will be mentioned or studied in this thesis.

Let $\mathcal{C}_{4+}$ be the set of all cycles of length at least 4. A graph is *chordal* if it is induced $\mathcal{C}_{4+}$-free, but it is also known that a graph G is chordal if and only if it is the intersection graph of a set of vertex sets of subtrees of a tree [89]. If the subtrees are paths, then G is called an *undirected path graph*. If the tree is a rooted top-down oriented tree and the subtrees are directed paths, then G is called a *directed path graph*. An *interval graph* is the intersection graph of a set of intervals of the real line. An example of a collection of intervals and the corresponding interval graph are shown in Figure 2.13. A *unit interval graph* is the intersection graph of a set of *unit length* intervals of the real line.

It follows from all the previous definitions that these graph classes are included one in the other in the following sequence: unit interval graphs, interval graphs, directed path graphs, undirected path graphs, chordal graphs.

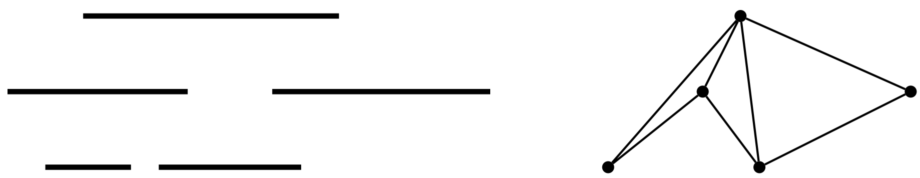


Figure 2.13: A collection of intervals and the corresponding interval graph.

A generalization of interval graphs is the one of *t-interval graphs* for $t \geq 1$, which are intersection graphs of a collection of sets each containing t intervals of the real line. If each interval has unit length, we have *unit t-interval graphs*. It is known that line graphs are 2-interval [86] and that planar graphs are 3-interval [175].

Another important class of intersection graphs is the class of *unit disk graphs*: a graph is unit disk if it is the intersection graph of a collection of disks all having unit radius and positioned in the plane.

A graph is a *permutation graph* if, given two parallel lines, it is the intersection graph of segments starting at one line and stopping at the other line.

Perfect graphs

A graph G is *perfect* if for every induced subgraph H of G, we have $\omega(H) = \chi(H)$. For example, any bipartite graph is perfect, since each such graph has clique number 2 and is 2-colourable;

[4] Series-parallel graphs are sometimes defined slightly differently; we refer to [31, Chapter 11.2] for a survey of several different (but non-equivalent) definitions.

odd cycles are easily seen not to be perfect, since they also have clique number 2, but are not 2-colourable.

The definition of a perfect graph is due to C. Berge, who conjectured that a graph is perfect if and only if its complement is perfect. This conjecture was proved by L. Lovász [143]. C. Berge also conjectured that a graph G is perfect if and only if neither G nor $\overline{G}$ contain an odd cycle of length at least 5 as an induced subgraph [19]. One implication was proved by C. Berge, and the other implication was proved to be true roughly fourty-five years later in [54]; this major result in graph theory is now known under the name of *Strong Perfect Graph Theorem*.

Perfect graphs are not only important because of their nice definition, but also because many important classes of graphs are subclasses of the class of perfect graphs, e.g. bipartite graphs, chordal graphs, interval graphs, permutation graphs. Moreover, the problems of determining the independence number and the chromatic number of a perfect graph are solvable in polynomial time [100]. For all these reasons, perfect graphs are widely studied in the literature. In this thesis, we also study various subclasses of perfect graphs.

Line graphs, quasi-line graphs, induced claw-free graphs

The *line graph* of a graph G, denoted $\mathcal{L}(G)$, is the graph with vertex set $E(G)$, where two vertices of $\mathcal{L}(G)$ are adjacent if the corresponding edges are adjacent in G. In other words, $\mathcal{L}(G)$ is the intersection graph of the edge set of G. An example of a graph and its line graph is given in Figure 2.14. In this thesis, we will study identifying codes in line graphs under various perspectives.

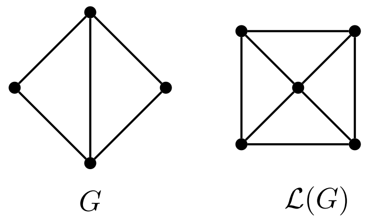
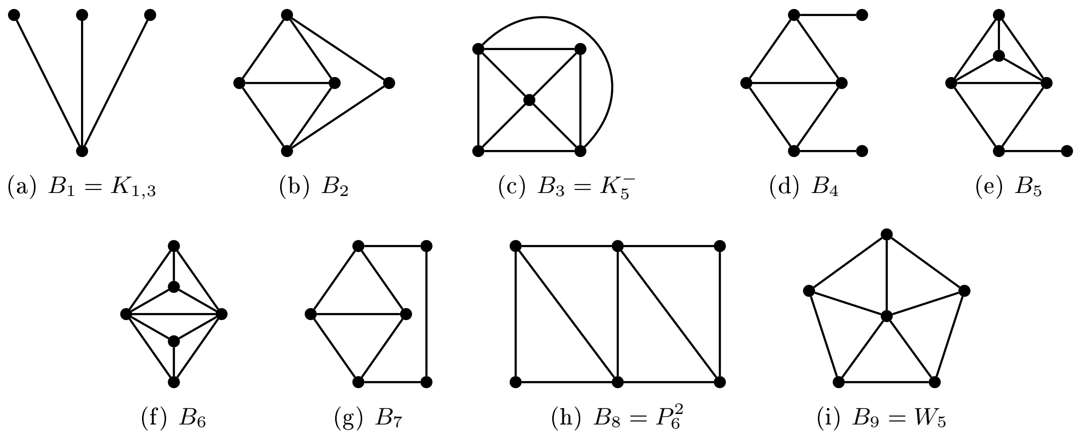


Figure 2.14: A graph and its line graph.

Let $\mathcal{B} = \{B_1 = K_{1,3}, B_2, B_3 = K_5^-, B_4, B_5, B_6, B_7, B_8 = P_6^2, B_9 = W_5\}$ be the list of graphs depicted in Figure 2.15.[5] We have the following theorem, due to L. W. Beineke:

Theorem 2.2 ([17]). *A graph G is a line graph if and only if it does not contain any of the graphs of $\mathcal{B}$ as an induced subgraph.*



(a) $B_1 = K_{1,3}$ (b) B_2 (c) $B_3 = K_5^-$ (d) B_4 (e) B_5

(f) B_6 (g) B_7 (h) $B_8 = P_6^2$ (i) $B_9 = W_5$

Figure 2.15: L. W. Beineke's list $\mathcal{B}$ of forbidden induced subgraphs

[5]The notation W_5 denotes the *wheel* on six vertices, which might be familiar to the reader.

The following relation between the classes of line graphs and perfect graphs, due to L. E. Trotter, is known:

Theorem 2.3 ([193])**.** *A line graph $\mathcal{L}(G)$ is perfect if and only if G contains no odd cycles of length more than 3 as a subgraph.*

A superclass of line graphs is the one of *quasi-line* graphs: a graph is quasi-line if the neighbourhood of each vertex is the union of two cliques. It follows that quasi-line graphs are *induced claw-free*[6], that is, induced $K_{1,3}$-free. The structure of induced claw-free graphs and quasi-line graphs is well-studied, see e.g. [55]. One can define a whole hierarchy of classes defined by forbidden induced subgraphs by considering induced $K_{1,\ell}$-free graphs, for $\ell \geq 3$. Unit disk graphs, for example, are known to be induced $K_{1,6}$-free.

Graphs of given tree-width and clique-width

An important graph parameter is the one of *tree-width*, which intuitively measures the similarity of a graph to a tree. This parameter has been introduced in [102] (cited in [75, Notes to Chapter 12]) and plays an important role in the aforementioned theory of graph minors [172, 173]. We refer to [75] for a definition. It is well-known that trees have tree-width at most 1 and series-parallel graphs have tree-width at most 2. Planar graphs, however, are in general not of bounded tree-width since the grid $G_{a,a}$ has $n = a^2$ vertices but tree-width $\Omega(\sqrt{n})$.

Similar width parameters have been introduced in the literature; we mention only one further parameter, introduced in [65]: the *clique-width* of a graph (however, we do not give a definition here). A *cograph* is a graph that can be built from single vertices using the disjoint union and complete join operations (cographs are also the induced P_4-free graphs). The class of cographs (which includes all cliques) is a subset of the class of graphs having clique-width at most 2. Trees have clique-width at most 3. In fact, if a graph has tree-width at most k, it has also clique-width at most $O(2^k)$ [66]. Parameters tree-width and clique-width have gained much interest because of their connection with computational complexity, as we will discuss in Section 2.3.4.

Random graphs and the probabilistic method

There are several classic ways to define *random graphs*, i.e. graphs generated using some specific random process. The usual question, given a class of random graphs, is to determine what kind of properties hold for one of these graphs *in average*, *with high probability* (for some valid definitions of these concepts), etc. A classical model is the so-called *Erdős-Rényi model* $\mathcal{G}_{n,p}$ [78], where for some real probability $p \in [0, 1]$ a graph $G \in \mathcal{G}_{n,p}$ on n (labelled) vertices[7] is constructed by setting each of its potential $\binom{n}{2}$ edges with uniform probability p.

We will not use the Erdős-Rényi model in this thesis; in Section 5.1.3, we will however consider *random regular graphs*, which are defined using the *configuration model* [26]. In this model, a d-regular multigraph on n vertices[8] is obtained by selecting some perfect matching of K_{nd} at random (see [26] for further reference). We will only consider cases where nd is even, as otherwise there does not exist any d-regular graph on n vertices. In the Configuration Model, the set of vertices in K_{nd} is partitioned into n cells of size d, and each cell W_v is associated to a vertex v of the random regular graph. An edge e of a perfect matching of K_{nd} induces either a loop in v (if it connects two elements of W_v) or an edge between v and u (if it connects a vertex from W_v to a vertex in W_u).

In general, this model may produce graphs with loops and multiple edges. We will denote by $\mathcal{G}^*(n, d)$ the former probability space and by $\mathcal{G}(n, d)$ the same probability space conditioned

[6] The standard denomination of these graphs is *claw-free*, however in this thesis, to avoid confusion with (non-induced) claw-free graphs, we call them *induced claw-free*.

[7] A graph is *labelled* if its n vertices are each assigned a distinct label, conveniently, integers from 1 to n.

[8] As opposed to the remaining of this thesis, when dealing with this model we use d to designate the degree of a random d-regular graph, as this notation is standard.

on the event that G is simple. It is shown in [147] that the following holds:

$$\Pr\big(G \in \mathcal{G}(n,d) \mid G \in \mathcal{G}^*(n,d)\big) = (1+o(1))e^{\frac{1-d^2}{4}} \qquad \text{if } d = o(\sqrt{n}).$$

Thus, for relatively small d (and in particular when d does not depend on n), any property that holds with probability tending to 1 for $\mathcal{G}^*(n,d)$ as $n \to \infty$, will also hold with probability tending to 1 for $\mathcal{G}(n,d)$. In this case we will say that the property holds *with high probability* (w.h.p.).

The *probabilistic method* is a technique where a property $\mathcal{P}$ of a class $\mathcal{C}$ of objects is proved by selecting at random an object O from some probabilistic universe over $\mathcal{C}$ (for example, a random graph). Then, one estimates the probability that $\mathcal{P}$ holds for O; if this probability is strictly positive, one concludes that an object satisfying $\mathcal{P}$ necessarily exists in $\mathcal{C}$. A classic reference about the probabilistic method is the textbook by N. Alon and J. Spencer [2]. In this thesis, we will use the probabilistic method in Sections 4.4 and 5.1.2.

Other important graph classes

Let us mention a few additional graph classes that will be studied or mentioned in this thesis.

A graph is a *split graph* if its vertex set can be partitioned into two sets, one of them being a clique, and the other one being an independent set. Split graphs form a subclass of chordal graphs.

A graph is a *Dominating Shortest Path* graph (*DSP graph* for short) if it contains a shortest path whose vertices form a dominating set. This class has been introduced in [136], where it was shown to be a superclass of the well-studied class of *asteroidal triple-free graphs*.

For some $k \geq 1$, a graph is called *k-degenerate* if its vertices can be ordered $v_1, v_2, \ldots, v_n$ such that each vertex v_i is adjacant to at most k vertices from $\{v_1, v_2, \ldots, v_{i-1}\}$.

2.3 Computational complexity

In this section, we introduce basic notions of computational complexity theory (also called the theory of computation) that will mainly be needed in the second part of this thesis.

2.3.1 Computational problems and algorithms

The field of *computational complexity* aims at classifying *computational problems* according to how efficiently they can be solved under various computational models. We refer to the books [4, 13, 88, 91, 161] for a detailed introduction to the topic.

A *computational problem* is a task which is defined by an input (called an instance) and a desired output related to the input. In this thesis, we consider two sorts of problems: *decision problems* and *optimization problems*, which we will define in the following subsections. The notion of an *algorithm* is a central one here. An algorithm is a sequence of instructions that can be used to solve a computational problem.

The *(worst-case) running time* of an algorithm $\mathscr{A}$, expressed as a function of $|I|$, the size of the instance I, is the (worst-case) number of steps that are performed by $\mathscr{A}$ on I. Usually we are interested only in the asymptotic running time of $\mathscr{A}$. Within the last decades, it has been assumed that an algorithm can be efficiently used in practice if its running time is polynomial in the size of the instance, that is, the running time is of the order $O(|I|^k)$ for some constant k. This idea is generally attributed to A. Cobham [58] and J. Edmonds [76], independently (cited in [91, Chapter 2.1]).

2.3.2 Decision problems and related classes

A *decision problem* P is defined by an infinite set $\mathcal{I}(P)$ of *instances*, and, given one instance, a question relative to this instance that can be answered either by “YES” or by “NO”. Given an instance I_P of P, an algorithm $\mathscr{A}_P$ for P will be able to access I_P, and will return either “YES” or "NO". The *size* $|I_P|$ of I_P is the number of bits that are needed to represent I_P in binary. We give the following examples of decision problems.

SAT
INSTANCE: A collection $\mathcal{Q}$ of clauses of literals from a set X of boolean variables (a literal is a variable x or its negation $\overline{x}$).
QUESTION: Can $\mathcal{Q}$ be satisfied, i.e. is there a truth assignment of the variables of X such that each clause contains at least one true literal?

3-SAT
INSTANCE: A collection $\mathcal{Q}$ of clauses of literals from a set X of boolean variables, where each clause contains at most three distinct literals.
QUESTION: Can $\mathcal{Q}$ be satisfied, i.e. is there a truth assignment of the variables of X such that each clause contains at least one true literal?

MATCHING
INSTANCE: A graph G and an integer k.
QUESTION: Does G have a matching of size at least k?

The definition of an efficient algorithm as one that has polynomial running time has led to the definition of the complexity class P, consisting of all decision problems for which there exists a polynomial-time algorithm solving it. The class P is a subset of the class NP, consisting of all decision problems P that can be verified in polynomial time, that is, there exists for each instance I_P of P, a so-called *certificate* $C(I_P)$ of size polynomial in $|I_P|$ with a *verifying algorithm* V of polynomial running time taking I_P and $C(I_P)$ as an input, and outputting "YES" if and only if I_P is a "YES"-instance (otherwise it outputs "NO").

One of the biggest (and most famous) open problems in computational complexity, introduced by S. Cook [62] is the one of determining whether P=NP. It is widely believed (or hoped) that the problems in NP are more difficult than those of P (that is, there would exist problems in NP that cannot be solved in polynomial time).

To this end, the framework of *polynomial-time reductions* between decision problems was introduced. The most widely used type of reductions for decision problems is the one of *Karp-reductions*, introduced in [130]. Given two decision problems P and Q, a Karp-reduction from P to Q is a polynomial-time computable function f that transforms any instance I_P of P into an instance $I_Q = f(I_P)$ of Q and such that I_P is a "YES"-instance of P if and only if I_Q is a "YES"-instance of Q. As an immediate consequence, note that if Q is in P, then P is also in P. A decision problem P is called NP-complete if it is in NP, and P is NP-hard, that is, each problem in NP can be Karp-reduced to P. We note that, assuming that P is not equal to NP, the existence of an infinite number of so-called NP-intermediate problems (that are neither in P nor NP-complete) was proved by R. Ladner [137].

For the examples we gave earlier, SAT (and its restricted version 3-SAT) is known to be NP-complete [62], whereas MATCHING is in P [76].

A first consequence of this definition is that if an NP-complete problem was proved to belong to P, then P=NP. Another consequence is that if an NP-complete problem P can be Karp-reduced to some problem Q of NP, then Q is also NP-complete. S. Cook proved in [62] that SAT is NP-complete, and R. M. Karp reduced SAT to twenty-one other fundamental combinatorial problems including 3-SAT, SET COVER, VERTEX COVER [130] (see later sections for their definitions). When facing with a (new) decision problem in NP, one wants to know whether it is in P or it is NP-complete. Once this is settled, one can investigate the same question when the set of instances is restricted. In the case of graph problems, there are many graph classes for which some NP-complete problems become polynomial-time solvable, or remain NP-complete. Summarizing and extending this classification for IDENTIFYING CODE is one of the goals of this thesis.

2.3.3 Optimization problems, approximation algorithms and related classes

Many decision problems have a natural optimization variant, where it is not just asked to separate "YES"-instances from "NO"-instances, but where one wants to compute a "good" solution to a given question. For instance, in the case of SAT, instead of deciding whether a boolean formula is satisfiable, one would ask to output a truth assignment satisfying the maximum possible number of clauses. If the solution to the optimization problem is optimal (according to some measure), the solution is said to be *exact*; otherwise it is called *approximate*. The goal of the field of *approximation* is to classify optimization problems for which it is possible to provide efficient methods for finding provably good approximate solutions. We give some formal definitions but refer to [13] for all details. For a recent (and regularly updated) survey of the concept of approximation, see [192].

An *optimization problem* P (either a minimization or a maximization problem) is defined by a set of instances and a desired solution. The goal is to optimize (minimize or maximize) the measure of the solution. The set of instances of P is denoted $\mathcal{I}(P)$, and the set of solutions of an instance I_P of P is denoted $\mathcal{SOL}(I_P)$; the size of an optimal solution for I_P is denoted $OPT(I_P)$ (we may note $OPT_P(I_P)$ to stress the fact that I_P is an instance of P). We give the following examples, from which the first two are special optimization variants of the first two examples from the previous section, SAT:

MAX SAT
INSTANCE: A collection $\mathcal{Q}$ of clauses of literals over a set X of boolean variables (a literal is a variable x or its negation $\overline{x}$).
SOLUTION: A boolean assignment $s : X \to \{0, 1\}$.
MEASURE: The number of clauses which are satisfied by s.

MAX (≤ 3,≤ 3)-SAT
INSTANCE: A collection $\mathcal{Q}$ of clauses over a set X of boolean variables, where each clause contains at most three distinct literals. Moreover, each variable appears in at most three clauses.
SOLUTION: A boolean assignment $s : X \to \{0, 1\}$.
MEASURE: The number of clauses which are satisfied by s.

MIN SET COVER
INSTANCE: A set system $(X, \mathcal{S})$.
SOLUTION: A set cover $S \subseteq \mathcal{S}$ of $(X, \mathcal{S})$.
MEASURE: The cardinality $|S|$ of the set cover.

Similar to decision problems, optimization problems for which the measure of a solution is polynomial-time computable and it can be checked in polynomial time whether a given answer is a valid solution, belong to the class NPO. If, moreover, an optimization problem can be solved exactly by a polynomial-time algorithm (that is, it always outputs an optimal solution), it belongs to the class PO.

The interest of studying optimization problems rather than decision problems comes from the possibility of judging the quality of a solution. A polynomial-time algorithm $\mathscr{A}_P$ for an optimization problem P is said to be an *α-approximation algorithm* for P if, given an instance I_P, it returns a solution SOL_{I_P} whose measure is at most $\alpha{\cdot}OPT(I_P)$ if P is a minimization problem, and at least $\frac{OPT(I_P)}{\alpha}$ if P is a maximization problem, respectively. Note here that α can either be a constant, or a function of the instance size. The value $\alpha = \frac{|SOL_{I_P}|}{OPT(I_P)}$ ($\alpha = \frac{OPT(I_P)}{|SOL_{I_P}|}$, respectively) is the *performance ratio* of $\mathscr{A}_P$.[9] An optimization problem is said to be *α-approximable* if it

[9]Some authors always take $\frac{|SOL_{I_P}|}{OPT(I_P)}$ as the performance ratio, even for maximization problems. We follow the style of [13] in this matter. Some authors, such as in [161], also use the *relative error* $\frac{|OPT(I_P)-|SOL_{I_P}||}{\max\{OPT(I_P),|SOL_{I_P}|\}}$ to measure the performance of an approximation algorithm, but the use of the performance ratio is more common in the literature.

admits an α-approximation algorithm. These notions have first been proposed by D. Johnson in [127].

Using this formalism, one can now classify optimization problems according to the kind of performance ratios that they can be approximated within. For example, the class APX is the class of optimization problems that are c-approximable for some constant c. More generally, for a family $\mathcal{F}$ of functions from $\mathbb{N}$ to $\mathbb{N}$, one can define the class $\mathcal{F}$-APX as the class of optimization problems having an $f(|I|)$-approximation algorithm, where I is an instance of P and $f \in \mathcal{F}$ [68].

With respect to this definition, the main complexity classes (sorted by increasing approximation hardness) that have been considered are APX, $\mathsf{log\text{-}APX}$ (the class $\mathcal{F}$-APX with $\mathcal{F}$ being the family of all polylogarithmic functions, i.e. those being of the form $O(\ln(n)^c)$ for some constant $c > 0$) and $\mathsf{poly\text{-}APX}$ (the class $\mathcal{F}$-APX with $\mathcal{F}$ being the family of all polynomials), see e.g. [68, 132].

Another class of interest is the class PTAS of optimization problems P having a *polynomial-time approximation scheme*, that is, an approximation algorithm taking an instance I_P of P and a constant $\epsilon > 0$ as its input, and which provides a $(1+\epsilon)$-approximation of I_P in time polynomial in $|I_P|$ (but not necessarily in $\frac{1}{\epsilon}$: for example, the algorithm could have running time $O\left(f\left(\frac{1}{\epsilon}\right) \cdot |I_P|^{g\left(\frac{1}{\epsilon}\right)}\right)$ for arbitrary functions f, g). Its subclass FPTAS contains problems having a ***fully** polynomial-time approximation scheme*, where the dependence on *both* $|I_P|$ and $\frac{1}{\epsilon}$ in the time complexity is polynomial (i.e. the running time can be of the form $O\left(\left(\frac{1}{\epsilon}\right)^{c_1} \cdot |I_P|^{c_2}\right)$ for constants c_1, c_2).

We have the following sequence of inclusions between complexity classes of optimization problems:[10,11]

$$\mathsf{PO} \subseteq \mathsf{FPTAS} \subseteq \mathsf{PTAS} \subseteq \mathsf{APX} \subseteq \mathsf{log\text{-}APX} \subseteq \mathsf{poly\text{-}APX} \subseteq \mathsf{NPO}.$$

In this thesis, we will mainly consider the classes PTAS, APX and $\mathsf{log\text{-}APX}$.

We will use different types of reductions between optimization problems that will be defined thereafter. Given a complexity class C of optimization problems and a type of reductions, an optimization problem P is said to be *C-hard* with respect to this type of reductions if each problem in C has a reduction of the given type to P. If moreover, P belongs to C, then P is said to be *C-complete.*

The following type of reductions, called *L-reductions* (for "linear reductions") was introduced in [162] for reducing problems in the class MAXSNP whose problems are expressed using specific logical statements. In fact the closure of MAXSNP under some natural type of reductions was shown to be computationally equivalent to the class APX [132]; these reductions are widely used to prove APX-hardness of optimization problems.

Definition 2.4 ([162])**.** *Let P and Q be two optimization problems. An* L-reduction *from P to Q is a four-tuple (f, g, α, β) where f and g are polynomial time computable functions and α, β 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:*
$$|OPT_P(I_P) - |SOL_{I_P}|| \leq \beta \cdot |OPT_Q(f(I_P)) - |SOL_{f(I_P)}||.$$

[10]The names of some of these classes (except for PTAS and APX) are not consensual. In [114, Chapter 10], APX, $\mathsf{log\text{-}APX}$ and $\mathsf{poly\text{-}APX}$ are called Class I, Class II and Class IV, respectively (the authors of [114, Chapter 10], S. Arora and C. Lund, also consider a Class III lying between Class II and Class IV, that is however believed to be equal to Class IV). In a more recent textbook [123], FPTAS, PTAS, APX, $\mathsf{log\text{-}APX}$ and $\mathsf{poly\text{-}APX}$ are called NPO(I), NPO(II), NPO(III), NPO(IV) and NPO(V), respectively.

[11]Recent research has shown that there exist much more complexity classes in NPO than those mentioned here (even if one considers only classes containing "natural" optimization problems); see the recent survey of L. Trevisan [192] for further details.

L-reductions are useful due to the following fact:

Theorem 2.5 ([162]). *Let P and Q be two optimization problems. If there exists an L-reduction from P to Q with parameters α and β and Q has a $(1+\epsilon)$-approximation algorithm for some $\epsilon > 0$, then P has a $(1+\alpha\beta\epsilon)$-approximation algorithm.*

It was shown in [162] that Max 3-SAT is APX-complete with respect to L-reductions. Given an optimization problem P_O, it can be shown, assuming that P≠NP, that there is no approximation algorithm for P_O with a given performance ratio r. In that case we say that it is NP-hard to approximate P_O within r. The following result is known:

Theorem 2.6 ([162]). *Any optimization problem P that is APX-hard with respect to L-reductions is NP-hard to approximate within a factor c, for some constant $c > 1$.*

As a consequence of Theorems 2.5 and 2.6, no APX-hard optimization problem belongs to PTAS, unless P=NP. We can state this fact in the following way:

Corollary 2.7 ([162]). *Let P and Q be two optimization problems. If there exists an L-reduction from P to Q with parameters α and β and it is NP-hard to approximate P within ratio $r_P = 1+\delta$, then it is NP-hard to approximate Q within ratio $r_Q = 1 + \frac{\delta}{\alpha\beta}$.*

We will use L-reductions in Chapter 7 to show that Min Id Code is APX-hard for several graph classes.

The drawback of L-reductions is that one cannot use them for defining a notion of completeness for other classes than APX [68]. One may use the stronger

AP-reductions (for "approximation preserving") instead, which is now accepted as one of the most suitable kind of reductions for preserving approximability factors [13, Chapter 8.6]. This notion was introduced in [68], which also contains a discussion about the power of different kinds of approximation-preserving reductions that can be found in the literature. We give the definition that can be found in [13, Definition 8.3]:

Definition 2.8 ([68, 13]). *Let P and Q be two optimization problems. An* AP-reduction *from P to Q is a triple (f, g, α) where f, g are functions and α 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:*

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

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

Theorem 2.9 ([68]). *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 $c > 0$.*

For an example, it is known that Min Set Cover is log-APX-complete [132]; it can be approximated within a factor of $1+\ln(|I|)$ for instances I [127], but it is NP-hard to approximate it within a factor $c\ln(|I|)$, for some constant $c > 0$ [169].

We will use AP-reductions in Chapter 6 to show that Min Id Code is log-APX-hard for several graph classes.

2.3.4 Complexity classes defined using logic and Courcelle's theorem

We already saw that complexity classes can be defined by means of properties expressable in a certain kind of logic, as for example the class of optimization problems MAXSNP [162, 132]. When considering decision problems, the class NP itself has been related to logic by R. Fagin, who proved that NP is exactly the set of decision problems asking whether a property expressible in *existential second-order logic*[12] is true or not.

A restriction of second-order logic is *monadic* second-order logic. Regarding graph properties, we distinguish τ_1*-monadic second-order logic* and τ_2*-monadic second-order logic*, defined as follows:

A graph property $\mathcal{P}$ is expressible in τ_1*-monadic second-order logic*, MSOL(τ_1) for short, if $\mathcal{P}$ can be defined using:

- vertices or sets of vertices of a graph G,
- the binary adjacency relation *adj* where $adj(u, v)$ holds if and only if u, v are two adjacent vertices of G,
- the unary cardinality operator *card* for sets of vertices of G,
- the logical operators OR ($\vee$), AND ($\wedge$), NOT ($\neg$), and
- the logical quantifiers $\exists$ and $\forall$ over vertices or sets of vertices of G.

The τ_2*-monadic second-order logic* (MSOL(τ_2) for short) is an extension of MSOL(τ_1), allowing the use of the binary incidence relation *inc*, where $inc(v, e)$ holds if and only if edge e is incident to vertex v in G. Moreover, MSOL(τ_2) allows the use of quantifiers $\exists$ and $\forall$ over edges and sets of edges of G.

The following results are extensions of B. Courcelle's theorem (which was originally stated for MSOL(τ_1) in [64]) and show that many graph properties can be checked in linear time for graphs of bounded tree-width or clique-width.

Theorem 2.10 ([3, 67])**.** *Let $\mathcal{P}$ be a graph property expressible in MSOL(τ_2) and let c be a constant. Then, for any graph G of tree-width at most c, it can be checked in linear time whether G has property $\mathcal{P}$.*

Theorem 2.11 ([66])**.** *Let $\mathcal{P}$ be a graph property expressible in MSOL(τ_1) and let c be a constant. Then, for any graph G of clique-width at most c, it can be checked in linear time whether G has property $\mathcal{P}$.*

However, it is very unlikely that the latter theorem can be extended to MSOL(τ_2):

Theorem 2.12 ([66])**.** *There exists a graph property $\mathcal{P}$ expressible in MSOL(τ_2) such that if it can be checked in polynomial time whether a complete graph has property $\mathcal{P}$, then P=NP (recall that complete graphs have clique-width 2).*

2.3.5 Other decision and optimization problems that we will use

In this thesis, we will reduce various decision and optimization problems to Identifying Code and Min Id Code, respectively, in order to demonstrate the complexity of solving these problems in various graph classes. We define these problems in this section.

Let us first describe restricted versions of boolean satisfiability problems. We define $(\leq r, \leq s)$-SAT as the usual SAT problem where each variable appears at most r times in the formula, and each clause has at most s literals. We denote by $(= r, \leq s)$-SAT the same problem where each variable appears *exactly* r times in the formula. We will use the following version of $(= 3, \leq 3)$-SAT:

[12] An existential second-order logic formula is a formula starting with an existential quantifier over some entities (possibly variables, sets of variables, relations, functions) followed by a first-order formula.

PLANAR $(= 3, \leq 3)$-SAT
INSTANCE: A collection $\mathcal{Q}$ of clauses over a set X of boolean variables, where each clause contains at least two and at most three distinct literals (a variable x or its negation $\overline{x}$). Moreover, each variable appears in exactly three clauses: twice in its non-negated form, and once in its negated form. Finally, the bipartite incidence graph of $\mathcal{Q}$, denoted $B(\mathcal{Q}, X)$, is planar.
QUESTION: Can $\mathcal{Q}$ be satisfied, i.e. is there a truth assignment of the variables of X such that each clause contains at least one true literal?

PLANAR $(= 3, \leq 3)$-SAT is known to be NP-complete [69]. We remark that some versions of $(\leq r, \leq s)$-SAT are known to be in P:

Theorem 2.13 ([191]). *For all $k \geq 1$, any instance of* $(\leq k, \leq k)$-SAT *is satisfiable.*

We also use the problem MAX $(\leq 3, \leq 3)$-SAT that we have defined earlier. MAX $(\leq 3, \leq 3)$-SAT is APX-complete [162].

Problem MIN VERTEX COVER will also be used. It is a special case of SET COVER that is 2-approximable (a result attributed to F. Gavril in [88]) and APX-complete [162].

MIN VERTEX COVER
INSTANCE: A graph G.
SOLUTION: A vertex cover $\mathcal{C}$ of G.
MEASURE: The cardinality $|\mathcal{C}|$ of the vertex cover.

We will also reduce from dominating set problems:

DOMINATING SET
INSTANCE: A graph G and an integer k.
QUESTION: Does G have a dominating set of size at most k?

MIN DOM SET
INSTANCE: A graph G.
SOLUTION: A dominating set $\mathcal{D}$ of G.
MEASURE: The cardinality $|\mathcal{D}|$ of the dominating set.

Let us formally define the optimization problems related to matchings and edge covers:

MAX MATCHING
INSTANCE: A graph G.
SOLUTION: A matching $\mathcal{M}$ of G.
MEASURE: The size $|\mathcal{M}|$ of the matching.

MIN EDGE COVER
INSTANCE: A graph G.
SOLUTION: An edge cover $\mathcal{C}$ of G.
MEASURE: The size $|\mathcal{C}|$ of the cover.

It is a famous result due to J. Edmonds [76] that MAX MATCHING is solvable in polynomial-time; later, the running time of his original algorithm has been improved:

Theorem 2.14 ([76, 148]). MAX MATCHING *can be solved in time $O\left(\sqrt{|V(G)|} \cdot |E(G)|\right)$ for an input graph G.*

In fact, the complexity of MIN EDGE COVER is the same as the one of MAX MATCHING; indeed, the best approach to solve the former problem is to compute a maximum matching and to greedily cover the edges that remain uncovered [138] (cited in [88, Problem GT1]). Hence we have the following corollary:

Corollary 2.15. Min Edge Cover *can be solved in time* $O\left(\sqrt{|V(G)|}\cdot|E(G)|\right)$ *for an input graph* G.

We will also use a generalization of Matching, that is known to be NP-complete [88]. It is defined as follows:

3-Dimensional Matching
INSTANCE: Three disjoint sets $A=\{a_1,\dots,a_n\}, B=\{b_1,\dots,b_n\}, C=\{c_1,\dots,c_n\}$ each of size n, and a set $\mathcal{T}=\{T_1,\dots,T_t\}$ of t triples of $A\times B\times C$.
QUESTION: Is there a perfect 3-dimensional matching $\mathcal{M}\subseteq\mathcal{T}$, i.e. a set of disjoint triples from $\mathcal{T}$ of size n (i.e. such that each element of $A\cup B\cup C$ belongs to some triple of $\mathcal{M}$).

We will also study the new decision and optimization problems associated to the concept of an edge-identifying code:

Edge-Identifying Code
INSTANCE: A graph G and an integer k.
QUESTION: Does G have an edge-identifying code of size at most k?

Min Edge-Id Code
INSTANCE: A graph G.
SOLUTION: An edge-identifying code $\mathcal{C}_E$ of G.
MEASURE: The size $|\mathcal{C}_E|$ of the code.

2.4 Identification problems that are related to identifying codes

In this section, we present a number of problems related to identifying codes. Some of these concepts are generalizations or restrictions of identifying codes; all deal with the identification of certain combinatorial structures using certain kinds of (sub)structures.

2.4.1 Codes identifying sets of vertices at a given distance

The notion of an identifying code of a graph G has been generalized to the one of an r*-identifying code*. Given an integer $r\geq 1$, an r-identifying code $\mathcal{C}$ of G is a subset of $V(G)$ such that for every vertex x, there is a vertex of $\mathcal{C}$ at distance at most r from x, and for each pair x,y of distinct vertices, there is a vertex of $\mathcal{C}$ that is at distance at most r from exactly one of x,y. In other words, all vertices are dominated and separated at distance r. We remark that an r-identifying code of G is an identifying code of the power G^r. Graphs admitting an r-identifying code (i.e. graphs G such that G^r is identifiable) are called r*-twin-free*.

This notion has subsequently been generalized to the one of an $(r,\leq\ell)$*-identifying code*, in which all *sets* of vertices of size at most ℓ are dominated and separated from each other at distance r. Graphs admitting an $(r,\leq\ell)$-identifying code have been called $(r,\leq\ell)$*-twin-free* [10].

We will not study these notions in this thesis, and refer to e.g. [9, 44, 85, 118, 153] for the interested reader.

2.4.2 Test covers, discriminating codes and Bondy's theorem

A problem that is much related to identifying codes and which has been studied for several decades is the *test cover problem*, which generalizes separating codes. Let $\mathcal{I}$ be a set of elements ("individuals") and $\mathcal{A}$, a set of subsets of $\mathcal{I}$ ("attributes"). We say that an attribute a of $\mathcal{A}$ *separates* two distinct elements I,I' of $\mathcal{I}$ if a belongs to exactly one of I,I'. A *test cover* of the set system $(\mathcal{I},\mathcal{A})$ is a set $\mathcal{T}\subseteq\mathcal{A}$ such that each pair of distinct sets of $\mathcal{I}$ is separated by some element of $\mathcal{T}$. Note that, as in the case of separating codes, a test cover may only exist if all pairs of individuals can actually be separated; we say that the set system $(\mathcal{I},\mathcal{A})$ is $\mathcal{I}$*-identifiable*.

The notion of a test cover appears in a large number of papers under different denominations (test cover in [71], *test collection* in [88], and *test set* in [156]). In fact, a well-celebrated theorem of J. A. Bondy on *induced subsets* [27] (usually referred to as "Bondy's theorem") can be seen as the first study of this problem.

Theorem 2.16 (Bondy's theorem [27]). *Let $\mathcal{A} = \{A_1, A_2, \cdots, A_n\}$ be a collection of n distinct subsets of an n-set X. Then there exists an element x of X such that the sets $A_1 - x, A_2 - x, \cdots, A_n - x$ are all distinct.*

Since its publication in 1972, this theorem has received a lot of attention. The original proof of [27] is a nice one and uses a graph-theoretic argument. Various alternative proofs of different nature have been provided for it, we refer to [14, Exercise 4.1.1],[25, Chapter 2],[144, Exercise 13.10] and [197] for some other interesting proofs. We also note that the theorem is valid even when the cardinality of $\mathcal{A}$ is strictly larger than the one of X. We will study extremal cases of Bondy's theorem in Section 3.2.3.

We observe that Bondy's theorem can be rephrased in the language of test covers as follows:

Theorem 2.17. *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system. Then there is a test cover of $(\mathcal{I}, \mathcal{A})$ of at most $|\mathcal{I}| - 1$ elements of $\mathcal{A}$.*

The analogy between test covers and identifying codes is however limited to some extent: the test cover problem does not ask for each individual to actually belong to an attribute of the test cover (i.e. there is no domination condition). However we have the following notion, which slightly differs from the one of a test cover.

Given an $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$, a subset $\mathcal{C}$ of $\mathcal{A}$ is a *discriminating code* of $(\mathcal{I}, \mathcal{A})$ if it is a test cover of $(\mathcal{I}, \mathcal{A})$ and each element of $\mathcal{I}$ belongs to some set of $\mathcal{C}$. The notion of a discriminating code was introduced in [36] and further studied in [37].

We point out that the exact formalism used in [36, 37] is slightly different, as the authors study discriminating codes in bipartite graphs, that is, instead of considering set systems, they consider their bipartite incidence graphs defined earlier. However the two points of view are easily seen to be equivalent.

We note that identifying codes in (di)graphs are a special case of discriminating codes:

Observation 2.18. *An identifying code in a graph G (digraph D, respectively) is precisely a discriminating code of the closed neighbourhood hypergraph of G (closed out-neighbourhood hypergraph of D, respectively).*

In fact, just as some particular cases of the set cover problem arising from specific combinatorial structures have gained a lot of interest (consider for example all variants of the dominating set problem, or the vertex cover problem), it is of interest to investigate special cases of the discriminating code problem having a particular structure. In this line of research and in the flavour of identifying codes, many other specific cases arising from graph theory are of particular interest. We will mention some of these problems in the next sections, where the sets $\mathcal{I}$ and $\mathcal{A}$ are families of substructures of a graph G.

The following theorem contains a generalization of Theorem 2.24 to discriminating codes:

Theorem 2.19 ([36]). *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system, and let $\mathcal{C}$ be a test cover of $(\mathcal{I}, \mathcal{A})$. Then $|\mathcal{C}| \geq \log_2(|\mathcal{I}|)$. If $\mathcal{C}$ is also a discriminating code of $(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}| \geq \log_2(|\mathcal{I}| + 1)$.*

The following upper bound can be seen as a direct corollary of Bondy's theorem. We refer to [36] for a formal proof in the context of discriminating codes.

Theorem 2.20 ([27, 36]). *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system, and let $\mathcal{C}$ be an inclusionwise minimal discriminating code of $(\mathcal{I}, \mathcal{A})$. Then $|\mathcal{C}| \leq |\mathcal{I}|$.*

In Chapter 3, we will characterize those instances that reach the bound of Theorem 2.20 and that are the closed neighbourhood hypergraphs of some digraph. We will also consider the case of infinite graphs and digraphs.

We make the following observation on the difference between the cardinalities of a minimum test cover and a minimum discriminating code, which can be at most 1.

Observation 2.21. *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system, and let $\mathcal{C}$ be a test cover of $(\mathcal{I}, \mathcal{A})$. If $\mathcal{C}$ is not a discriminating code of $(\mathcal{I}, \mathcal{A})$, there is an element $A \in \mathcal{A}$ such that $\mathcal{C} \cup \{A\}$ is one.*

Proof. All pairs of distinct individuals are separated by $\mathcal{C}$. Observe that at most one individual I may remain uncovered by $\mathcal{C}$. It is then sufficient to add to $\mathcal{C}$ an arbitrary attribute A covering I to get a valid discriminating code. ☆

Observe that the problem of finding a test cover or discriminating code of a set system $(\mathcal{I}, \mathcal{A})$ is the same as the one of finding a set cover of a related set system.

Reduction 2.22 (Min Test Cover $\to$ Min Set Cover). *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system. Let $(\mathcal{D}, \mathcal{A})$ be the set system where $\mathcal{D}$ consists of all pairs of distinct elements of $\mathcal{I}$, and an element A of $\mathcal{A}$ contains a pair $\{I, I'\}$ of $\mathcal{D}$ if A separates I and I' in $(\mathcal{I}, \mathcal{A})$.*

Reduction 2.23 (Min Discrim Code $\to$ Min Set Cover). *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system. Let $(\mathcal{D}, \mathcal{A})$ be the set system built from $(\mathcal{I}, \mathcal{A})$ using Reduction 2.22. Define $(\mathcal{D}^*, \mathcal{A})$ by adding the set $\mathcal{I}$ to $\mathcal{D}$, as well as any element I of $\mathcal{I}$ to any set $A \in \mathcal{A}$ containing I in $(\mathcal{I}, \mathcal{A})$.*

It is easily observed that finding a test cover (discriminating code, respectively) of $(\mathcal{I}, \mathcal{A})$ is equivalent to finding a set cover of $(\mathcal{D}, \mathcal{A})$ as built in Reduction 2.22 (of $(\mathcal{D}^*, \mathcal{A})$ from Reduction 2.23, respectively).

Also recall that finding a set cover of a set system is the same as finding a transversal of the dual of the corresponding hypergraph. Similarly, the notion of a *distinguishing transversal* of a hypergraph $(\mathcal{I}, \mathcal{A})$ has been recently introduced in [112]. Finding a distinguishing transversal in the dual of $(\mathcal{I}, \mathcal{A})$ is a reformulation of the problem of finding a discriminating code of $(\mathcal{I}, \mathcal{A})$.

2.4.3 Identifying open codes

One immediate variation of the notion of identifying codes is the one where separation and domination are considered with respect to *open* neighbourhoods rather than closed neighbourhoods. This notion has been first studied in [119] under the name of *strongly identifying codes*, and further studied under the name of *open neighborhood locating-dominating sets* [176, 177] and very recently, *identifying open codes* [112].

One can see a relation to the classic notion of a *total dominating set*, where domination in the closed neighbourhood is replaced by domination in the open neighbourhood, see e.g. [108]. Moreover, such as for any classical identifying code, any identifying open code is also a locating-dominating set.

In this setting, a graph admits an identifying open code if and only if it has no *false* twins; such graphs are studied in [187] under the name of *point-determining* graphs. Similarly, when considering digraphs, an interesting series of papers [149, 150, 151, 160] has considered the notion of *extensional acyclic digraphs*, that is, digraphs in which no two vertices have the same *open out-neighbourhood*.[13]

2.4.4 Identification of vertices using stars, cycles and paths

The case where vertices of a graph G are identified using partial closed neighbourhoods (i.e. $\mathcal{I} = V(G)$ and $\mathcal{A}$ is the set of all stars in G) is called *watching systems*; it was introduced in [11] and further studied in [12]. See also the thesis of D. Auger [7].

The case where $\mathcal{I} = V(G)$ ($\mathcal{I} = V(G) \cup E(G)$, respectively) and $\mathcal{A}$ is the set of all sets of vertices (vertices and edges, respectively) inducing a cycle in G has also been considered in [117] ([174], respectively). Similarly, we studied the case where $\mathcal{A}$ is the set of paths of a graph [FK12].

[13] Following the definition of an identifying code in a digraph, a natural definition of open-identifiable digraphs would rather consider open *in*-neighbourhoods, but of course the two concepts are equivalent: just reverse all arcs of the digraphs to switch from one definition to the other.

2.4.5 Identifying the edges of a graph

Consider the case of an *identifying vertex cover* of a graph G, that is, a subset $\mathcal{C}$ of vertices of G that is a vertex cover of G, and where for each pair e, e' of distinct edges of G, $e \cap \mathcal{C} \neq e' \cap \mathcal{C}$. This notion was very recently studied in [113]. It was also briefly investigated in [153, Theorem 1.2], where this problem was shown to be equivalent to the one of finding the complement of a distance-3-independent set of G.

One can also study *edge-identifying codes*, that is, the identification of edges using edges. We introduced this concept in [FGN+12]; Section 5.3 (where we also define this notion in detail) is devoted to its study.

2.4.6 Resolving sets and metric dimension

One of the earliest concepts related to identifying vertices in graphs is the one of a *resolving set*, introduced independently in [104] and [180] (under the name of *locating set*). In a graph G, a subset $\mathcal{C}$ of vertices of G is a resolving set if for any pair of distinct vertices of G, there is an element c of $\mathcal{C}$ for which the distances from c to both vertices of the pair are distinct (in this context we may say that c separates the pair). The smallest cardinality of a resolving set of a graph is its *metric dimension*. Results on this topic have been surveyed in [47].

We note that this problem does not have a natural formulation as a sub-problem of the discriminating code problem. However, it is equivalent to the set cover problem in the set system $(\mathcal{I}, \mathcal{A})$ with $\mathcal{I}$ being the set of pairs of distinct vertices of $V(G)$, and where each set A_x of $\mathcal{A}$ corresponds to a distinct vertex x of G and contains all pairs which are separated by x [133].

2.4.7 Identifying colourings

Several notions of identification of *adjacent* vertices using colours have been studied. Recently, the notion of *locally identifying colouring* has been introduced [80]; a proper vertex-colouring c of a graph G is locally identifying if for each pair of adjacent vertices u, v which are not twins, the sets $\{c(x) \mid x \in N[u]\}$ and $\{c(x) \mid x \in N[v]\}$ are distinct. We further studied this notion in [FHL+12].

A similar notion has been considered with respect to edge-colourings: a proper edge-colouring c is *adjacent vertex-distinguishing* if for each pair of adjacent vertices u, v, the sets $\{c(e) \mid e$ is incident to $u\}$ and $\{c(e) \mid e$ is incident to $v\}$ are distinct [115] (the concept was introduced under the name of *strong edge-colouring* in [203]).

Other kinds of identifying colourings have been studied, such as *locating colourings*, that are proper colourings where for each pair u, v of adjacent vertices, there is a colour class C whose members c_u, c_v that, among all vertices of C, are closest to u and to v respectively, have the property that $d(u, c_u) \neq d(v, c_v)$ [48].

2.5 Existing work on identifying codes in (di)graphs related to this thesis

In this section, we present known results about identifying codes that are relevant with respect to the topics addressed in this thesis.

2.5.1 General bounds on the identifying code number

The first part of this thesis is dedicated to study lower and upper bounds on the identifying code number in restricted graph classes. We review the known bounds from the literature.

A lower bound

One of the first results about identifying codes is the following bound. It follows easily from the fact that when having an identifying code $\mathcal{C}$, we assign to each of the n vertices of our graph, a nonempty and distinct subset of $\mathcal{C}$; there are at most $2^{|\mathcal{C}|} - 1$ such sets, hence $n \leq 2^{|\mathcal{C}|} - 1$.

Theorem 2.24 ([131])**.** *For any identifiable graph G on n vertices, $\gamma^{ID}(G) \geq \lceil \log_2(n+1) \rceil$.*

In fact, all graphs reaching the bound of Theorem 2.24 have been characterized as follows:

Theorem 2.25 ([154])**.** *Let G be an identifiable graph on n vertices with $\gamma^{ID}(G) = \lceil \log_2(n+1) \rceil$ and a minimum identifying code $\mathcal{C}$ of G. Then $\mathcal{C}$ induces an identifiable graph in G, and each vertex of $V(G) \setminus \mathcal{C}$ has a distinct nonempty set of $\{\mathcal{C}' \mid \mathcal{C}' \subseteq \mathcal{C}\} \setminus \{N_{G[\mathcal{C}]}[x] \mid x \in \mathcal{C}\}$ as its neighbourhood within $\mathcal{C}$. The subgraph induced by $V(G) \setminus \mathcal{C}$ may be arbitrary.*

Theorem 2.25 gives a way of constructing all graphs reaching the bound of Theorem 2.24. Observe that in this construction, whenever $G[\mathcal{C}]$ and $G[V(G) \setminus \mathcal{C}]$ have no edges, G is bipartite; whenever $G[\mathcal{C}]$ has no edge but $G[V(G) \setminus \mathcal{C}]$ is a complete graph, G is a split graph:

Corollary 2.26. *The bound of Theorem 2.24 is tight for infinitely many bipartite graphs and split graphs (and therefore chordal graphs).*

An example of a bipartite graph on seven vertices reaching the bound of Theorem 2.24 is given in Figure 2.16.

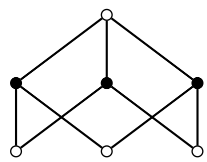


Figure 2.16: A graph reaching the bound of Theorem 2.24. Black vertices belong to an optimal identifying code.

An upper bound

Regarding upper bounds, observe that, as it is the case for dominating sets, the edgeless graph $\overline{K_n}$ has $\gamma^{\mathrm{ID}}(G) = \gamma(G) = n$. However the following theorem shows that it is the only such (finite) example.

Theorem 2.27 ([22, 96])**.** *Let G be an identifiable graph on n vertices. If G has at least one edge, then $\gamma^{ID}(G) \leq n - 1$. This bound is tight for any value of $n \geq 3$.*

Examples of graphs reaching the bound of Theorem 2.27 are the star $K_{1,n-1}$ and the path P_4. A part of this thesis (Section 3.4) will be dedicated to the classification of all graphs reaching this bound.

In fact, we note that Theorem 2.27 has been extended to infinite graphs as follows.

Theorem 2.28 ([96])**.** *Let G be a (not necessarily finite) identifiable graph with at least one edge and having all its vertices with a finite degree. Then there exists a vertex x of G such that $V(G) \setminus \{x\}$ is an identifying code of G.*

We will strengthen this theorem by characterizing all infinite graphs having their whole vertex set as their only identifying code in Section 3.5.

2.5.2 Bounds in specific graph classes

One of the goals of this thesis is to investigate lower and upper bounds for the identifying code number in various graph classes such as graphs of given maximum degree, graphs with girth 5, interval graphs or line graphs. We first review bounds that can already be found in the literature.

Graphs of given maximum degree

A lower bound depending on the maximum degree and the order of the graph is known:

Theorem 2.29 ([131]). *Let G be an identifiable graph on n vertices having maximum degree Δ. Then $\gamma^{ID}(G) \geq \frac{2n}{\Delta+2}$.*

Notice that the bound of Theorem 2.29 is an improvement over the bound of Theorem 2.24 whenever $\Delta \leq \frac{2n}{\lceil \log_2(n+1) \rceil} - 2$. The bound is tight, and we characterize all graphs reaching it in Section 4.1.

It seems that no *upper* bound on parameter γ^{ID} involving only the order and the maximum degree of a graph was known prior to our work. Chapter 4 is dedicated to this matter, where we present a conjecture and several bounds approximating the conjectured one in various graph classes.

Paths and cycles

The identifying code numbers of paths and cycles have been fully determined. We present the known results.

Theorem 2.30 ([23, 90]). *We have:*

$$\gamma^{ID}(P_n) = \begin{cases} \frac{n+1}{2}, & \text{if } n \text{ is odd,} \\ \frac{n}{2} + 1, & \text{if } n \text{ is even and } n \geq 4. \end{cases}$$

Theorem 2.31 ([23, 90, 97]). *We have:*

$$\gamma^{ID}(C_n) = \begin{cases} 3, & \text{if } 4 \leq n \leq 5, \\ \frac{n}{2}, & \text{if } n \text{ is even and } n \geq 6, \\ \frac{n+1}{2} + 1, & \text{if } n \text{ is odd and } n \geq 7. \end{cases}$$

We also note that the identifying code numbers of powers of paths and cycles (i.e. their r-identifying code numbers, see Section 2.4.1) have all been determined, see [23, 49, 90, 97, 129, 171, 199].

Trees

The following lower bound on the identifying code number of a tree is an improved bound over the general bound of Theorem 2.24:

Theorem 2.32 ([24]). *Let T be an identifiable tree on n vertices. Then $\gamma^{ID}(T) \geq \frac{3(n+1)}{7}$. This bound is tight for infinitely many trees.*

The *complete k-ary tree of height h* is the rooted tree T_k^h of height h where each non-leaf vertex of T_k^h has k children (hence $\Delta(T_k^h) = k+1$). We have $|V(T_k^h)| = \frac{k^h-1}{k-1}$. The following theorem is known:

Theorem 2.33 ([24]). *The bound $\gamma^{ID}(T_2^h) = \left\lceil \frac{20|V(T_k^h)|}{31} \right\rceil$ holds. For any $k \geq 3$, it holds that $\gamma^{ID}(T_k^h) = \left\lceil \frac{k^2|V(T_k^h)|}{k^2+k+1} \right\rceil = \left\lceil |V(T_k^h)| - \frac{|V(T_k^h)|}{\Delta(T_k^h)-1+1/\Delta(T_k^h)} \right\rceil$.*

Hypercubes

As many code-like structures, identifying codes have been much studied in the hypercube, and it has been a challenging topic. The area is rich of results, but as it is not one of the main topics of this thesis, we only mention a few of them. As to now, only the identifying code number of the first seven hypercubes is known, see [38]. It has been proved as a non-trivial result, that the identifying code number grows with the dimension of the hypercube: for any $d \geq 1$, $\gamma^{\mathrm{ID}}(\mathcal{H}_d) \leq \gamma^{\mathrm{ID}}(\mathcal{H}_{d+1})$ [155]. Though precise formulas are not known, the asymptotic order of magnitude of $\gamma^{\mathrm{ID}}(\mathcal{H}_d)$ is known to be given by $\gamma^{\mathrm{ID}}(\mathcal{H}_d) = O\left(\frac{2^d}{d}\right) = O\left(\frac{|V(\mathcal{H}_d)|}{d}\right)$ [120]. Checking whether a given set of vertices of a hypercube is an identifying code is a computationally difficult task [121, 120]. We refer to the PhD theses [128, 153, 166] for further reference on the matter.

Grids

Identifying codes have been studied in many infinite grid-like structures, where one asks for the minimum *density* of an infinite identifying code (the identifying code number is not well-defined in an infinite graph); we do not give a formal definition here, but intuitively, it measures the average local ratio of the number of code vertices over the total number of vertices. Infinite graphs that have been considered are the classic infinite grid $P_\infty \Box P_\infty$ [18, 41, 131, 184], the infinite strip $P_m \Box P_\infty$ for small values of m [30, 107, 153], the infinite d-dimensional grid $P_\infty^{(1)} \Box \ldots \Box P_\infty^{(d)}$ [184], the infinite king grid (i.e. the grid with diagonals) [59, FLP12], the infinite hexagonal grid [41, 131, 184] and the infinite triangular grid [41, 131]. It turns out that finite versions of these infinite grid-like graphs are even harder to study, as it is also the case for the domination number of classical grids (which was determined completely only very recently [93]).

Planar and outerplanar graphs

The following theorem was originally stated for the location-domination number γ^{LD}. However, it is also valid for the identifying code number since $\gamma^{\text{LD}}(G) \leq \gamma^{\text{ID}}(G)$ for any identifiable graph G.

Theorem 2.34 ([183]). *Let G be an identifiable planar graph on n vertices. If $\gamma^{LD}(G) \geq 4$, then $\frac{n+10}{7} \leq \gamma^{LD}(G) \leq \gamma^{ID}(G)$. If G is outerplanar, then $\frac{2n+3}{7} \leq \gamma^{LD}(G) \leq \gamma^{ID}(G)$.*

Random graphs

In Section 5.1.3, we will study the identifying code number of random regular graphs. Identifying codes have already been studied in several models of random graphs. The authors of [85] study identifying codes in the Erdős-Rényi model. They prove the following theorem:

Theorem 2.35 ([85]). *Let p be a real from $\left[\frac{4\ln(\ln(n))}{\ln(n)}, 1 - \frac{4\ln(\ln(n))}{\ln(n)}\right]$ and let $G \in \mathcal{G}_{n,p}$. Then w.h.p. G is identifiable and $\gamma^{ID}(G) = (1 \pm o(1))\frac{2\ln(n)}{\ln\left(\frac{1}{p^2+(1-p)^2}\right)}$.*

In [85] are also given precise *threshold functions* according to $p(n)$ for the property that $G \in \mathcal{G}_{n,p}$ is identifiable.[14] The authors also study $(1, \leq \ell)$-identifying codes in $\mathcal{G}_{n,p}$.

Another model of random graphs has been studied in [158] in connection with identifying codes: the model of *random geometric graphs* (for which the textbook [165] is a classic reference). In this model, given an integer $d \geq 1$ (the dimension), an integer n and a *radius* $r = r(n)$, n random vertices of $[0,1]^d$ are considered, and two vertices are adjacent if and only if their euclidean distance within $\mathbb{R}^d$ is at most r. The authors of [158] have studied the case where $d = 2$ (i.e. the case of *random unit disk graphs*). They give threshold functions according to $r(n)$ for the property that a random unit disk graph G on n vertices with radius $r(n)$ is identifiable. In contrast to the threshold functions for $\mathcal{G}_{n,p}$ from [85], except when $r(n) = o\left(\frac{1}{n}\right)$, the probability that G is identifiable is at most $\frac{1}{2}$. The authors also turn their attention to locating-dominating sets in the same setting, which are very close to identifying codes but have the advantage of existing in any graph. They prove the following tight bounds on $\gamma^{\text{LD}}(G)$:

Theorem 2.36 ([158]). *Let G be a random unit disk graph on n vertices and radius $r(n)$. If $r(n) = o\left(\sqrt{\frac{\ln(n)}{n}}\right)$, then w.h.p. $\gamma^{LD}(G) = n^{1-o(1)}$. If for some $\epsilon > 0$, $r(n) \leq \frac{\sqrt{2}}{2} - \epsilon$ and $r(n) = \omega\left(\frac{1}{\sqrt{n}}\right)$, then w.h.p. $\gamma^{LD}(G) = \left(\frac{n}{r(n)}\right)^{2/3+o(1)}$.*

[14] A *threshold function* $f(n)$ for property $\mathcal{P}(n)$ according to $p(n)$ is a function such that, if $p(n) = o(f(n))$, then w.h.p. $\mathcal{P}(n)$ does not hold, and if $p(n) = \omega(f(n))$, then w.h.p. $\mathcal{P}(n)$ holds (or vice-versa). See e.g. [2] for more details.

2.5.3 Complexity of Identifying Code, Min Id Code and related problems

In this section, we gather results from the literature about the decision and optimization problems Identifying Code and Min Id Code and related problems. This is relevant since in the second part of this thesis, we will consider these two problems for various restricted graph classes, extending the work from the literature.

Test covers and discriminating codes

Since identifying codes are special cases of discriminating codes (who are related to test covers), we first present complexity results for the related computational problems. We first define the following restriction of Min Set Cover:

k-bounded Min Set Cover
INSTANCE: A set system $(X, \mathcal{S})$ such that each set of $\mathcal{S}$ has size at mostr k.
SOLUTION: A set cover $S \subseteq \mathcal{S}$ of $(X, \mathcal{S})$.
MEASURE: The cardinality $|S|$ of the set cover.

As mentioned before, Min Set Cover is log-APX-complete. Its variant k-bounded Min Set Cover is known to be $(1 + \ln(k))$-approximable [127], but it is APX-hard [162].

We also define decision and optimization problems for the notion of a discriminating code and a test cover, as well as restrictions similar to the one of Min Set Cover:

Test Cover
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$ and an integer k.
QUESTION: Is there a test cover of $(\mathcal{I}, \mathcal{A})$ having size k?

Min Test Cover
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$.
SOLUTION: A test cover $\mathcal{T} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$.
MEASURE: The cardinality $|\mathcal{T}|$ of the test cover.

k-bounded Min Test Cover
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$ such that each element of $\mathcal{A}$ has size at most k.
SOLUTION: A test cover $\mathcal{T} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$.
MEASURE: The cardinality $|\mathcal{T}|$ of the test cover.

Discriminating Code
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$ and an integer k.
QUESTION: Is there a discriminating code of $(\mathcal{I}, \mathcal{A})$ having size k?

Min Discrim Code
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$.
SOLUTION: A discriminating code $\mathcal{D} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$.
MEASURE: The cardinality $|\mathcal{D}|$ of the discriminating code.

k-bounded Min Discrim Code
INSTANCE: An $\mathcal{I}$-identifiable set system $(\mathcal{I}, \mathcal{A})$ such that each element of $\mathcal{A}$ has size at most k.
SOLUTION: A discriminating code $\mathcal{D} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$.
MEASURE: The cardinality $|\mathcal{D}|$ of the discriminating code.

The two decision problems related to discriminating codes and test covers are NP-complete:

Theorem 2.37 ([88]). Test Cover *is NP-complete.*

Theorem 2.38 ([37]). *Discriminating Code is NP-complete, even when the bipartite incidence graph of the instance set system is planar.*

Because of the immediate proximity of their definitions, Min Test Cover and Min Discrim Code have the same computational complexity. The following is known.

Theorem 2.39 ([71, 156]). *Both Min Test Cover and Min Discrim Code are log-APX-complete.*

Theorem 2.40 ([71, 156]). *For any $k \geq 1$, both k-bounded Min Test Cover and k-bounded Min Discrim Code are $O(\ln(k))$-approximable. For any $k \geq 2$, they are APX-hard.*

These results follow from reductions to and from Min Set Cover and k-bounded Min Set Cover. The positive approximation results follow from the reductions from problems Min Test Cover and Min Discrim Code to Min Set Cover described in Reductions 2.22 and 2.23.

We note that by Observation 2.18, the problem of finding an identifying code of a graph G can be reduced to the one of finding a discriminating code of a given set system. This implies that, as a corollary of Theorems 2.39 and 2.40, Min Id Code is $O(\ln(n))$-approximable (where n is the number of vertices of the instance graph), and that Min Id Code is $O(\ln(\Delta))$-approximable for graphs of maximum degree Δ. However, the hardness results from Theorems 2.39 and 2.40 so not apply to Min Id Code. We will see that similar results hold indeed for this problem.

Decision problem Identifying Code

We define the decision problem related to finding an r-identifying code of a given size:

r-Identifying Code
INSTANCE: An r-identifiable graph G and an integer k.
QUESTION: Does G have an r-identifying code of size at most k?

The decision problem Identifying Code was first proved to be NP-complete in [60] by a reduction from 3-SAT. This result was then improved by the following theorem, also by reduction from 3-SAT:

Theorem 2.41 ([44]). *r-Identifying Code is NP-complete for any $r \geq 1$, even for bipartite graphs.*

Reductions from Vertex Cover have led to the two following theorems, showing that Identifying Code is NP-complete in restricted subclasses of planar graphs:

Theorem 2.42 ([9]). *Identifying Code is NP-complete, even when restricted to planar graphs of maximum degree 3 and with girth 9. r-Identifying Code is NP-complete for any $r \geq 1$ when restricted to planar graphs of maximum degree 3.*

Theorem 2.43 ([6]). *Identifying Code is NP-complete, even when restricted to planar graphs of maximum degree 4 and with arbitrarily large girth.*

The following theorem (via a reduction from Planar 3-SAT) showed that Identifying Code is hard, even in *planar bipartite* unit disk graphs. It is a well-known fact that unit disk graphs are induced $K_{1,6}$-free.

Theorem 2.44 ([158]). *Identifying Code is NP-complete, even when restricted to planar bipartite unit disk graphs, and hence to planar bipartite induced $K_{1,6}$-free graphs.*

The problem, when extended to digraphs, is also intractable in several special cases.

Directed Identifying Code
INSTANCE: An identifiable digraph D and an integer k.
QUESTION: Does D have an identifying code of size at most k?

Theorem 2.45 ([43]). Directed Identifying Code *is NP-complete, even when restricted to strongly connected asymmetric bipartite digraphs and to asymmetric bipartite digraphs without directed cycles.*[15]

On the positive side, we have the following theorems regarding trees:

Theorem 2.46 ([6]). *There is a linear-time algorithm to solve* Identifying Code *for the class of trees.*

Theorem 2.47 ([40]). *There is a linear-time algorithm to solve* Directed Identifying Code *for the class of oriented trees.*

In fact, using the framework of monadic second-order logic, one can generalize Theorem 2.46 using the following proposition:

Proposition 2.48 ([153, Chapter 2]). *Given a graph G and an integer k, let $\mathcal{ID}(G,k)$ be the property that $\gamma^{ID}(G) \leq k$. Property $\mathcal{ID}(G,k)$ can be expressed in MSOL(τ_1).*

We get the following corollary of Theorem 2.11:

Corollary 2.49. Identifying Code *can be solved in linear time when restricted to classes of graphs having their clique-width (and hence tree-width) bounded by a constant.*

We remark however, that Corollary 2.49 does not provide practical linear-time algorithms, as the algorithms given by this technique have huge complexities, due to constants that are exponential in the length of the MSOL(τ_1) formula.

Optimization problem Min Id Code

Regarding the optimization problem Min Id Code, the following theorems are known:

Theorem 2.50 ([20, 140, 185]). Min Id Code *is log-APX-complete.*

Theorem 2.51 ([94]). Min Id Code *is APX-hard, even in graphs of maximum degree 8.*

In fact, we note that the reduction from Vertex Cover of [9] giving Theorem 2.42 can also be seen as an L-reduction when applied to non-planar subcubic graphs; this strengthens Theorem 2.51 in the following way:

Theorem 2.52 ([9]). Min Id Code *is APX-hard, even in graphs of maximum degree 3 and girth 9.*

When considering Min Id Code for planar graphs, we can make use of Theorem 2.34, stating that $\frac{n+10}{7} \leq \gamma^{\mathrm{LD}}(G)$ for any identifiable planar graph G (recall that if G is identifiable, $\gamma^{\mathrm{LD}}(G) \leq \gamma^{\mathrm{ID}}(G)$), since $V(G)$ is always an identifying code of any identifiable graph G:

Corollary 2.53. Min Id Code *and* Min Loc-Dom Set *are 7-approximable for planar graphs.*

Locating-dominating sets

In the literature, the computational complexity of finding an identifying code is often studied together with the one of finding a locating-dominating set. We briefly recall some of the results, before mentioning that when dealing with approximation theory, these problems are closely related. But first, we formally define the computational problems related to locating-dominating sets:

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

[15] A digraph is *strongly connected* if for every two vertices x, y there is a directed path from x to y.

MIN LOC-DOM SET
INSTANCE: A graph G.
SOLUTION: A locating-dominating set $\mathcal{C}$ of G.
MEASURE: The cardinality $|\mathcal{C}|$ of the set.

We have the following theorems:

Theorem 2.54 ([181]). LOCATING-DOMINATING SET *has a linear-time algorithm for trees.*

Theorem 2.55 ([61]). LOCATING-DOMINATING SET *is NP-complete, but polynomial-time solvable in the class of series-parallel graphs.*

We note that similar methods than the ones used for Corollary 2.49 for the case of identifying codes can be used to extend the previous positive complexity results to all graphs of bounded clique-width.

Theorem 2.56 ([185]). MIN LOC-DOM SET *is log-APX-complete.*

Theorem 2.57 ([94, 185]). MIN LOC-DOM SET *is APX-complete when restricted to graphs of maximum degree 5.*

We note that all (in)approximability results for MIN ID CODE carry over to MIN LOC-DOM SET with a multiplicative factor of 2 on the performance ratios:

Theorem 2.58 ([94]). *Let G be an identifiable graph. We have $\gamma^{LD}(G) \leq \gamma^{ID}(G) \leq 2\gamma^{LD}(G)$.*

The proof of Theorem 2.58 is constructive: given a locating-dominating set D of G, the authors of [94] show a simple way to obtain an identifying code of G of cardinality at most $2|D|$. The reverse construction is trivial, since any identifying code is a locating-dominating set. Using this result, we are able to tightly link the complexity of approximating MIN LOC-DOM SET and MIN ID CODE:

Corollary 2.59. *Any α-approximation algorithm for* MIN LOC-DOM SET *or* MIN ID CODE *can be transformed into a 2α-approximation algorithm for* MIN ID CODE *or* MIN LOC-DOM SET*, respectively.*

Corollary 2.60. *If it is NP-hard to α-approximate* MIN ID CODE *or* MIN LOC-DOM SET*, then it is NP-hard to $\frac{\alpha}{2}$-approximate* MIN LOC-DOM SET *or* MIN ID CODE*, respectively.*

Dominating sets

The computational complexity of finding a dominating set of a graph is a well-studied problem. We will relate this complexity to the one of finding an identifying code in Tables 1.8 and 1.9 for various graph classes.

In general, DOMINATING SET is NP-complete [88] and MIN DOM SET is log-APX-complete [127, 169]. Of course, when restricted to some graph classes, these general complexities vary. We do not formally present all results regarding this topic, but refer to Figure 1.10 instead, which graphically represents some of the known results, and to Table 1.8 and 1.9, where references to the literature can be found for all covered graph classes. We also refer the reader to [53], [108, Chapter 12] and to [109, Chapters 8 and 9] for further reference.

Part I

Combinatorial aspects

Chapter 3
Extremal (di)graphs for identifying codes

THis chapter is dedicated to the study of those graphs and digraphs that reach the upper bound of Theorem 2.17 for the minimum size of a discriminating code[1] and the refined upper bound on the identifying code number of an undirected graph from Theorem 2.27. Recall that for a finite undirected graph G on n vertices, $\gamma^{\text{ID}}(G) = n$ if and only if $G \cong \overline{K_n}$; otherwise, $\gamma^{\text{ID}}(G) \leq n-1$ (Theorem 2.27). The case of digraphs was not much studied in the literature; we point the reader to [40, 43, 179].

We start with providing a general tool in Section 3.1 that we will use several times in this chapter (Proposition 3.1). This result will allow us to work using induction.

In Section 3.2, we show that the class of digraphs D with $\overrightarrow{\gamma^{\text{ID}}}(D) = |V(D)|$ is much richer than in the undirected case by exhibiting a new family of oriented graphs satisfying this property. This family can be described as the set of oriented graphs that are the closure of some top-down rooted oriented tree. We then give a full characterization of digraphs D for which $\overrightarrow{\gamma^{\text{ID}}}(D) = |V(D)|$ in Theorem 3.8 by showing that any such graph is the closure of an oriented tree as described before. We then apply this theorem to extremal cases in Bondy's theorem, by giving, in Theorem 3.11, a characterization of those set systems that have the set of all elements as only discriminating code (stated in the language of Bondy's theorem, they are such that any removal of an element will make one set empty).

In Section 3.3, we tackle the case of infinite digraphs having their whole vertex set as unique identifying code. We extend the previous class of closures of rooted oriented trees to infinite trees defined similarly. We then characterize all infinite *oriented* graphs having their whole vertex set as unique identifying code in Theorem 3.15 by proving that they must also be isomorphic to the closure of some infinite oriented tree. It seems that this result cannot be extended easily to *directed* graphs, a case that we leave open.

Then, in Section 3.4, we investigate those finite undirected graphs G with $\gamma^{\text{ID}}(G) = |V(G)|-1$. We introduce a new infinite family of such graphs in Definition 3.21, and show that, roughly speaking, the closure of this family under disjoint union and complete join still yields examples of graphs G with $\gamma^{\text{ID}}(G) = |V(G)|-1$. We then fully characterize all those graphs in Theorem 3.27, by showing that they are either a star or built in the way described above.

Finally, in Section 3.5, we extend Definition 3.21 to define an infinite undirected graph that we call A_∞. We show that this graph, as well as other infinite graphs built from it, need their

[1]This bound, of course, also holds for the identifying code number of a (di)graph.

whole vertex set in any identifying code. We show that this construction fully describes all such infinite undirected graphs in Theorem 3.31.

The results contained in this chapter appeared in [FGK+11] (joint work with E. Guerrini, M. Kovše, R. Naserasr, A. Parreau and P. Valicov) for those dealing with undirected graphs, and in [FNP12] (joint work with R. Naserasr and A. Parreau) for those about digraphs.

We point out that compared to the version of [FGK+11] that has been published, a minor flaw has been corrected in the proof of Theorem 3.27 (Case b).

3.1 A useful proposition

In this section, we provide a proposition that is useful for proving upper bounds on minimum identifying codes by induction. We will use it indeed in the proofs of several theorems of this chapter.

We give a general version of this proposition, for the case of discriminating codes. This version was given in A. Parreau's PhD thesis [164, Proposition 2.6]. As a matter of fact, this proposition is also valid for the cases of identifying codes in digraphs and in undirected graphs, since these are sub-problems of the discriminating code problem. Up to our knowledge, this proposition was first given in [FGK+11, Proposition 3] for the case of identifying codes in undirected graphs.

Proposition 3.1. *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system, and let $S \subseteq \mathcal{I}$. Denoting $(\mathcal{I}', \mathcal{A}')$ the set system obtained from $(\mathcal{I}, \mathcal{A})$ by removing all elements of S, let $\mathcal{C} \subseteq \mathcal{I}'$ be a discriminating code of $(\mathcal{I}', \mathcal{A}')$. Then, $(\mathcal{I}, \mathcal{A})$ has a discriminating code of size at most $|\mathcal{C}| + |S|$.*

Proof. First of all, note that since $(\mathcal{I}, \mathcal{A})$ is $\mathcal{I}$-identifiable, $(\mathcal{I}', \mathcal{A}')$ is $\mathcal{I}'$-identifiable (removing elements does not destroy this property). Let $\mathcal{C} \subseteq \mathcal{I}'$ be a discriminating code of $(\mathcal{I}', \mathcal{A}')$, and let us order the elements of S arbitraily $i_1, \ldots, i_{|S|}$. We set $\mathcal{C}_0 = \mathcal{C}$ and follow the given order in a step by step procedure. At each step k, we build the set $\mathcal{C}_k$ from $\mathcal{C}_{k-1}$ as follows. If $\mathcal{C}_{k-1}$ dominates and separates all the elements of $(\mathcal{I} \setminus S) \cup \{i_1, \ldots, i_k\}$ in the original set system $(\mathcal{I}, \mathcal{A})$, we let $\mathcal{C}_k = \mathcal{C}_{k-1}$. Otherwise, either i_k does not belong to any set of $\mathcal{C}_{k-1}$, or it is not separated from some unique element i. In the first case, we add any set of $\mathcal{I}$ containing i_k to $\mathcal{C}_{k-1}$ to build $\mathcal{C}_k$. In the other case, since $(\mathcal{I}, \mathcal{A})$ is $\mathcal{I}$-identifiable, there exists a set of $\mathcal{A}$ which separates i_k from i; we add this set to $\mathcal{C}_{k-1}$ to get $\mathcal{C}_k$. It is clear that in the end of the process, all elements are dominated and separated. Since at each step, we have only added at most one element to the solution, the claim follows. ☆

Of course, since an identifying code in a (di)graph is just a discriminating code in its closed (out-)neighbourhood hypergraph, Proposition 3.1 also holds for these special cases. However, we would like to apply Proposition 3.1 to (di)graphs G in such a way that we choose a set S of vertices, and consider a (minimum) identifying code of $G - S$. For this, we need the additional condition that $G - S$ is identifiable.

Proposition 3.2. *Let D be an identifiable digraph and let S be a subset of its vertices such that $D - S$ is identifiable. Then $\overrightarrow{\gamma^{ID}}(D) \leq \overrightarrow{\gamma^{ID}}(D - S) + |S|$.*

We state the previous proposition with digraphs, but of course it can be applied to undirected graphs since an identifying code of an undirected graph G is just an identifying code of the digraph obtained from G by replacing each edge by two symmetric arcs.

3.2 Digraphs with their whole vertex set as only identifying code

In this section, we classify all finite digraphs in which the whole vertex set is the only identifying code. We call such digraphs *extremal*.

3.2.1 A new family of extremal digraphs

As for the case of undirected graphs, a digraph D with no arc has $\overrightarrow{\gamma^{\text{ID}}}(D) = |V(D)|$. However, there are more examples for which this holds, such as the digraph on two vertices and with a unique arc, or the one on three vertices (a source and two sinks) and two arcs (see Figure 3.1).

Figure 3.1: Two small extremal digraphs

As we will see, there are much more such digraphs. We begin with the following definitions of operations on digraphs, illustrated in Figure 3.2, which will help us to describe them.

We recall that the disjoint union of two digraphs D_1 and D_2, $D_1 \oplus D_2$, is the digraph whose vertex set is $V(D_1) \cup V(D_2)$ and whose arc set is $A(D_1) \cup A(D_2)$.

Given a digraph D and a vertex x not in $V(D)$ we define $x\overrightarrow{\lhd}(D)$ to be the digraph with vertex set $V(D) \cup \{x\}$ whose arcs are the arcs of D together with each arc $\overrightarrow{xv}$ for every $v \in V(D)$.

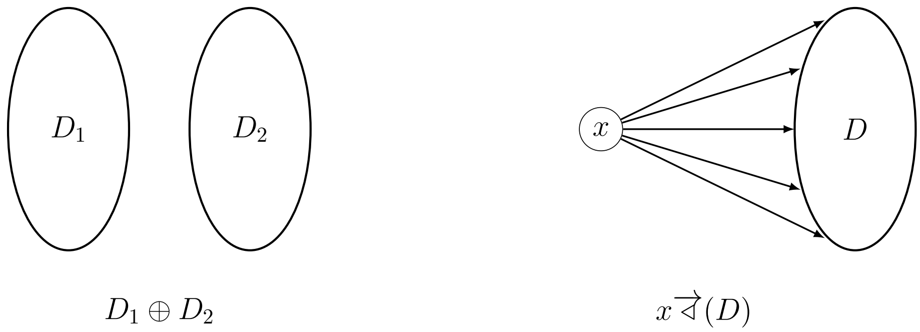


Figure 3.2: Illustrations of the two operations

Definition 3.3. *Let $(K_1, \oplus, \overrightarrow{\lhd})$ be the closure of the one-vertex graph K_1 with respect to the operations $\oplus$ and $\overrightarrow{\lhd}$. That is, the class of all graphs that can be built from K_1 by repeated applications of $\oplus$ and $\overrightarrow{\lhd}$.*

The *transitive closure* of a digraph D is a digraph obtained from D by adding the arc $\overrightarrow{xy}$ whenever there is a directed path from x to y. We recall that a *rooted oriented tree* is an oriented tree with a specific vertex v (called the root) such that for every other vertex u the path connecting u to v is a directed path from v to u. In such a tree, given an arc $\overrightarrow{xy}$, we say that x is the parent of y and that y is a child of x. The vertices with a directed path to x are called *ancestors* of x and vertices with a directed path from x are *descendants* of x. A *rooted oriented forest* is a disjoint union of rooted oriented trees.

By these definitions it is easy to check that:

Observation 3.4. *Every element of $(K_1, \oplus, \overrightarrow{\lhd})$ is the transitive closure of a rooted oriented forest.*

For a digraph D in $(K_1, \oplus, \overrightarrow{\lhd})$ let us denote by $\overrightarrow{F}(D)$ the rooted oriented forest whose transitive closure is D.

Proposition 3.5. *For every digraph D in $(K_1, \oplus, \overrightarrow{\lhd})$ we have $\overrightarrow{\gamma^{ID}}(D) = |V(D)|$. Furthermore, if a vertex x is a source in D, then $V(D) - x$ is a separating code of D. Otherwise, the vertex x and its parent in $\overrightarrow{F}(D)$ are the only ones not being separated from each other by the set $V(D) - x$.*

Proof. Let $\mathcal{C}$ be an identifying code of D. Except for its roots, each vertex of the forest $\overrightarrow{F}(D)$ must be in $\mathcal{C}$ in order to be separated from its parent. But the sources need also to be in $\mathcal{C}$ in order to be dominated. ☆

Let D be an identifiable digraph with vertex set $\{x_1, x_2, \cdots, x_n\}$ and let $\mathcal{A} = \{N^+[x_1], N^+[x_2], \ldots, N^+[x_n]\}$. Then $(\mathcal{A}, V(D))$ forms a set system satisfying the conditions of Bondy's theorem. Therefore we have:

Proposition 3.6. *Let D be a finite identifiable digraph. Then $\overrightarrow{\gamma^{S}}(D) \leq |V(D)| - 1$.*

The following corollary of the previous proposition will also be needed.

Proposition 3.7. *In a finite identifiable digraph D, $\overrightarrow{\gamma^{ID}}(D) = |V(D)|$ if and only if $\overrightarrow{\gamma^{S}}(D) = |V(D)|-1$ and for every minimum separating code of D, there is a vertex which is not dominated.*

Proof. Suppose $\overrightarrow{\gamma^{\mathrm{ID}}}(D) = |V(D)|$. By Proposition 3.6 we know that $\overrightarrow{\gamma^{S}}(D) \leq |V(D)|-1$. On the other hand, $\overrightarrow{\gamma^{\mathrm{ID}}}(D) \leq \overrightarrow{\gamma^{S}}(D)+1$ thus $\overrightarrow{\gamma^{S}}(D) = |V(D)|-1$. Now, if all the vertices were dominated by some minimum separating code of D, that code would also be identifying, a contradiction.

Conversely, if $\overrightarrow{\gamma^{S}}(D) = |V(D)| - 1$ and for any minimum separating code of D, there is a vertex which is not dominated, we are forced to take all the vertices in any identifying code in order to get the domination property. ☆

3.2.2 The characterization

The following theorem shows that the family of Definition 3.3 is exactly the class of finite digraphs in which the whole vertex set is the only identifying code.

Theorem 3.8. *Let D be a finite identifiable digraph. If $\overrightarrow{\gamma^{ID}}(D) = |V(D)|$ then $D \in (K_1, \oplus, \overrightarrow{\triangleleft})$.*

Proof. By contradiction, assume D is the smallest digraph for which $\overrightarrow{\gamma^{\mathrm{ID}}}(D) = |V(D)|$ but $D \notin (K_1, \oplus, \overrightarrow{\triangleleft})$. We consider two cases:

Case a. Assume that there exists a vertex x of D such that x has no out-neighbours. Then $D - x$ is an identifiable digraph. By Proposition 3.2 we have $\overrightarrow{\gamma^{\mathrm{ID}}}(D - x) = |V(D - x)|$ and, therefore, by the minimality of D, $D - x \in (K_1, \oplus, \overrightarrow{\triangleleft})$. Thus $\overrightarrow{F}(D - x)$ is well-defined. Since $V(D) - x$ is not an identifying code of D, in $V(D) - x$ either there is a vertex y which is not separated from x, or x is not dominated.

If x is not dominated, then x is an isolated vertex and, therefore, D is the disjoint union of two members of $(K_1, \oplus, \overrightarrow{\triangleleft})$, hence D is also in the family. So, there is a vertex y which is not separated from x. Therefore, $N^+(x) = N^+(y) \cup \{y\}$. Then D is the transitive closure of the oriented tree built from $\overrightarrow{F}(D - x)$ by adding x as a child of y, a contradiction.

Case b. Every vertex has at least one out-neighbour. By Proposition 3.6 we know that there exists a vertex x such that $D - x$ is identifiable and as in the previous case $\overrightarrow{F}(D - x)$ is well-defined.

Since every vertex, in particular, every leaf t of $\overrightarrow{F}(D - x)$ has an out-neighbour in D, we must have $\overrightarrow{tx} \in A(D)$. Thus $d^+(x) \geq 1$ and, therefore, $V(D) - x$ is a dominating set. But since it is not an identifying code there is a vertex $y \neq x$ which is not separated from x by $V(D) - x$, i.e., $N^+(x) = N^+(y) \cup \{y\}$.

We now claim that y is the only leaf of $\overrightarrow{F}(D - x)$. That is because if $t \neq y$ is a leaf then $t \in N^+(x)$ so $t \in N^+(y)$ which is a contradiction. But there has to be at least one leaf in $\overrightarrow{F}(D - x)$ thus y is the only leaf in $\overrightarrow{F}(D - x)$ and, therefore, $\overrightarrow{F}(D - x)$ is a path. Now since $d^-(x) > 0$, there is a vertex in $N^-(x)$. We have $y \notin N^-(x)$ since otherwise we would have $N^+(x) = N^+(y)$.

First, assume that there exists a vertex $t \in N^-(x)$ such that the parent of t in $\overrightarrow{F}(D - x)$ is not in $N^-(x)$. We claim that $\mathcal{C} = V(D) - t$ is an identifying code. Indeed, x is the only vertex dominated by all vertices of $\mathcal{C}$. Vertex t and its parent are separated by x. Finally, each other pair of vertices from $V(D) - x$ is separated by the one which is a descendant of the other in $\overrightarrow{F}(D - x)$.

Now, assume that there is no vertex in $N^-(x)$ with its parent in $\overrightarrow{F}(D - x)$ not belonging to $N^-(x)$. Let r be the root of $\overrightarrow{F}(D - x)$. In particular, $r \in N^-(x)$. We claim that $\mathcal{C} = V(D) - r$ is an identifying code. Indeed, x is the only vertex dominated by all vertices of $\mathcal{C}$. Each pair of vertices from $V(D) - x$ is separated by the one which is a descendant of the other in $\overrightarrow{F}(D - x)$. Finally, r is the only vertex which is dominated only by x. ☆

3.2.3 An application to extremal cases in Bondy's theorem

In this section, unless specifically mentioned, a set system is a pair $(\mathcal{A}, X)$ with X being any set of size n and $\mathcal{A}$ being a collection of n distinct subsets of X. When applying Bondy's theorem (Theorem 2.16) to a set system $(\mathcal{A}, X)$ where all subsets in $\mathcal{A}$ are distinct *and nonempty*, it is a natural request to be able to choose an element x of X such that $A_i - x \neq \emptyset$ for each $A_i \in \mathcal{A}$.[2] Such set systems will be called *extremal*. More precisely, an extremal set system is a set system $(\mathcal{A}, X)$ in which elements of $\mathcal{A}$ are all distinct and nonempty, and where for any element x of X either there is an element $A_i \in \mathcal{A}$ with $A_i - x = \emptyset$ or there is a pair of distinct sets $A_i, A_j \in \mathcal{A}$ such that $A_i - x = A_j - x$. In other words, they are the set systems that have only one discriminating code: the whole set X. In this section we characterize all such extremal cases.

We would like to mention that almost any proof of Bondy's theorem (e.g., see [14, 25, 27, 144]) works for an extension of this theorem in which we are allowed to have more elements in X than in $\mathcal{A}$. We then look for a subset $X' \subset X$ of size $|X| - |\mathcal{A}| + 1$ such that all the induced sets $A_i - X'$ are distinct. The following proposition is now an easy consequence of this general version of Bondy's theorem.

Proposition 3.9. *Let $(\mathcal{A}, X)$ be a set system with $|X| > |\mathcal{A}|$ where all the subsets in $\mathcal{A}$ are distinct and nonempty. There is a subset X' of X of size $|X| - |\mathcal{A}|$ such that all the subsets $A \cap (X - X')$ for $A \in \mathcal{A}$ are nonempty and distinct.*

Proof. Let X_0 be the subset of size $|X| - |\mathcal{A}| + 1$ found by extended version of Bondy's theorem i.e., all the subsets $A \cap (X - X_0)$ are distinct. Thus there is at most one subset A_0 such that $A_0 \cap (X - X_0)$ is empty. Let $x_0 \in X_0$ be an element of A_0. Then $X' = X_0 - \{x_0\}$ satisfies the proposition. ☆

As in the case of set systems, one can also define the *incidence bipartite graph* of a digraph. To this end, given a digraph D on a vertex set $\{x_1, x_2, \cdots, x_n\}$, we define $B(D)$ to be the bipartite graph on $S = \{x_1, x_2, \cdots, x_n\}$ and $T = \{x'_1, x'_2, \cdots, x'_n\}$ with x_i being adjacent to x'_j if either $\overrightarrow{x_i x_j} \in A(D)$ or $i = j$. The latter condition of adjacency implies that for a bipartite graph to be the incidence bipartite graph of some digraph it must admit a perfect matching. Below we prove that this is also a sufficient condition.

Lemma 3.10. *A bipartite graph $G = (S \cup T, E)$ is the incidence bipartite graph of some digraph D if and only if $|S| = |T|$ and G admits a perfect matching $\varphi : S \to T$.*

Proof. It follows from the definition of the incidence bipartite graph of a digraph that G must have both parts of the same size and admits a perfect matching which matches the two copies of each vertex.

Now, given a bipartite graph G with parts S and T of equal size together with a perfect matching φ, one can construct a digraph D with vertex set S in the following way. For each pair x, y of vertices if $x\varphi(y) \in E(G)$, then $\overrightarrow{xy}$ is an arc of D. The constructed digraph has G as its incidence bipartite graph. ☆

An example of the correspondence between a bipartite graph with a perfect matching (thick edges) and a digraph is given in Figure 3.3.

We are now ready to achieve our goal of classifying the extremal set systems in Bondy's theorem when $|\mathcal{A}| = |X|$.

Theorem 3.11. *A set system $(\mathcal{A}, X)$ with $|\mathcal{A}| = |X|$ is extremal if and only if its incidence bipartite graph $B(\mathcal{A}, X)$ is the incidence bipartite graph of a digraph in $(K_1, \oplus, \overrightarrow{\lhd})$.*

Proof. If $B(\mathcal{A}, X)$ is the incidence bipartite graph of a member D of $(K_1, \oplus, \overrightarrow{\lhd})$, then by Theorem 3.8 we have $\overrightarrow{\gamma^{\text{ID}}}(D) = |V(D)|$ thus by Proposition 3.7 with any separating code of size $|V(D)| - 1$ there must be a vertex which is not dominated. Any separating code $V(D) - x$ of D

[2]This is not always possible: consider the set system $\mathcal{A}$ consisting of all singletons of X.

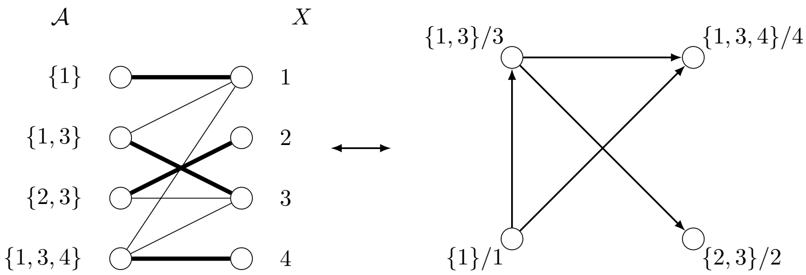


Figure 3.3: Digraph of a bipartite graph

corresponds to the choice of x in Bondy's theorem, and leaves some vertex in D undominated — that is, there exists $A_i \in \mathcal{A}$ such that $A_i - x = \emptyset$. Hence $(\mathcal{A}, X)$ is extremal.

For the other direction, we distinguish two cases.

Case a. $B(\mathcal{A}, X)$ admits a perfect matching. Then the directed graph D built from $B(\mathcal{A}, X)$ using this matching as in the method of Proposition 3.7 is such that $\overrightarrow{\gamma^{\mathrm{ID}}}(D) = |V(D)|$. Thus by Theorem 3.8, $D \in (K_1, \oplus, \overrightarrow{\triangleleft})$ and we are done.

Case b. $B = B(\mathcal{A}, X)$ does not contain a perfect matching. Then by P. Hall's marriage theorem (Theorem 2.1), there is a subset X' of X such that $|N_B(X')| < |X'|$. For $A_i \in \mathcal{A}$ we define $A'_i = A_i \cap X'$ and $\mathcal{A}' = \{A'_1 \cdots A'_{|\mathcal{A}|}\} - \emptyset$. Consider the set system $(\mathcal{A}', X')$ $(|\mathcal{A}'| < |X'|)$. By Proposition 3.9 there is an element x_0 in X' such that $A'_i - x_0$ are all nonempty and distinct as long as they induce distinct elements in $\mathcal{A}'$. Now it is easy to check that $X - x_0$ induces nonempty and distinct elements on $\mathcal{A}$. ☆

Using the previous theorem we can describe the extremal set systems $(\mathcal{A}, X)$ with $|\mathcal{A}| = |X|$ purely in the terminology of sets:

Corollary 3.12. *A set system $(\mathcal{A}, X)$ with $|\mathcal{A}| = |X|$ is extremal if and only if:*

- $\bigcup_{A_i \in \mathcal{A}} A_i = X$
- *for any subset A_i with at least two elements, there is an element x of A_i such that $A_i - x \in \mathcal{A}$*

Proof. If D is a digraph of $(K_1, \oplus, \overrightarrow{\triangleleft})$, it is easy to see that the set system corresponding to its incidence bipartite graph $B(D)$ has the properties.

Now, let $(\mathcal{A}, X)$ be a set system having the properties of the corollary. Assume there is an element x such that all $A_i - x$ are distinct and nonempty. Take A_i to be the smallest subset containing x, it exists because $\bigcup A_i = X$. Then $A_i \neq \{x\}$ and so there is an element y such that $A_i - y = A_j$. Then necessarily $x \neq y$ and A_j is a smaller set than A_i containing x. This is a contradiction. ☆

3.3 The case of infinite oriented graphs

In this section, we consider the case of infinite digraphs in which the whole vertex set is the only identifying code. To avoid set theoretic problems we only consider infinite graphs on a countable vertex set.

3.3.1 Families of extremal infinite oriented graphs

As we will see in Section 3.5, the set of symmetric digraphs that are extremal with respect to identifying codes is very rich. Hence the family of all such digraphs (not necessarily symmetric) seems to be too rich to characterize. In this section, we provide such a characterization for the class of all oriented graphs. We begin with some definitions.

A connected (and possibly infinite) oriented graph D is a *finite-source transitive tree* (f.s.t. tree for short) if:

(1) for each vertex v of D, $N^+[v]$ induces the transitive closure of a finite directed path, P_v, with v as its end vertex, and

(2) for each pair x, y of vertices, there is a vertex $z \in V(D)$ such that $P_x \cap P_y = P_z$.

Note that in an f.s.t. tree D, since each path P_x is finite, point (2) of the definition implies that for any pair x, y of vertices, P_x and P_y begin with the same vertex. Hence all paths P_x begin with the same vertex, which is the unique source of D.

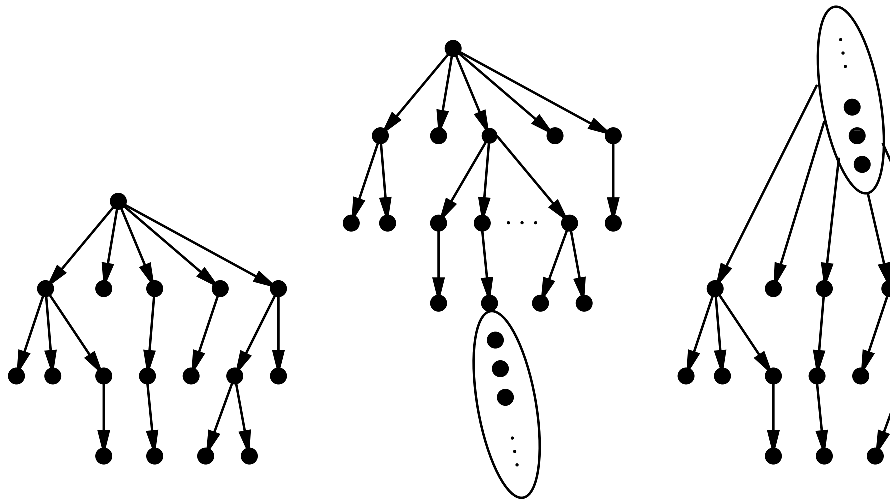
An oriented graph D is an *infinite-source transitive tree* (i.s.t. tree for short) if:

(1) for all vertices x of D, $N^+[x]$ induces the transitive closure of an infinite path (we denote it by P_x), and

(2) for any pair x, y of vertices of D, there is a vertex $z \in V(D)$ such that $P_x \cap P_y = P_z$.

Note that an i.s.t. tree has no source vertex but one can imagine infinity as its source.

Finally we say that an oriented graph D is a *source transitive tree* if it is either an f.s.t. tree or an i.s.t. tree, see Figure 3.4 for examples. For each pair x, y of vertices of a source transitive tree D, paths P_x and P_y share the vertices of a path P_z which includes the "beginning" of both P_x and P_y. Hence, all arcs of D can be oriented in the same direction. Moreover, there cannot be any cycle in the union of all paths P_x. This implies that each source transitive tree D is the transitive closure of a finite or infinite "rooted" oriented tree (even if an i.s.t. tree has no properly defined root vertex, one can regard infinity as its root) which we call the *underlying tree* of D. Notice that the collection of f.s.t. trees on a finite set of vertices is exactly the set of connected elements of $(K_1, \oplus, \overrightarrow{\triangleleft})$.



fst-tree on a finite set $\quad$ fst-tree on an infinite set $\quad$ ist-tree

Figure 3.4: Underlying trees of source transitive trees

Proposition 3.13. *The only identifying code of a source transitive tree is its whole set of vertices.*

Proof. Let $D = (V, A)$ be a source transitive tree and x be any vertex of D. If x is a source (it can only happen if D is an f.s.t. tree), then x must be in any identifying code of D in order to be dominated. If x is not a source, then to separate x from its parent in the underlying tree of D, x itself must be in any identifying code. ☆

We are now ready to build the whole family of oriented graphs that need their whole vertex set to be identified. To this end given any oriented graph H we first build the family $\Psi(H)$ of extremal oriented graphs as follows:

For each vertex x of H if x is a source of H, then we assign an f.s.t. tree T_x to x. If x is not a source, then we assign an i.s.t. tree T_x to x. The choice of T_x is free but each T_x has its distinct set of vertices. For each arc $\overrightarrow{xy}$ of H we also associate a subset $V_{\overrightarrow{xy}}$ of $V(T_x)$ (the choice of $V_{\overrightarrow{xy}}$ is also free). We now build a member of $\Psi(H)$ by taking $\cup V(T_x)$ as the vertex set, arcs of T_x are also arcs of the new graph and, furthermore, for any $z \in V_{\overrightarrow{xy}}$ and any $t \in V(T_y)$, we add an arc $\overrightarrow{zt}$.

Proposition 3.14. *Given an oriented graph H, any digraph D in $\Psi(H)$ can only be identified by its whole vertex set.*

Proof. The sources of D are exactly the sources of the f.s.t. trees T_x for source-vertices x of H and need to be in any identifying code in order to be dominated. If a vertex u of D is not a source, then it is in an fst or i.s.t. tree T_x and there is, like in the proof of Proposition 3.13, a vertex v of T_x such that $N^+[u] \cap V(T_x) = (N^+[v] \cap V(T_x)) \cup \{u\}$ (v simply is the parent of u in the underlying tree of T_x). By our construction, any incoming neighbour of u not in T_x is also an incoming neighbour of v so $N^+[u] = N^+[v] \cup \{u\}$ and u must be in any identifying code of D. ☆

3.3.2 The characterization

Theorem 3.15. *Let D be an infinite identifiable oriented graph. Then a proper subset of $V(D)$ is an identifying code of D unless $G \in \Psi(H)$ for some finite or infinite oriented graph H.*

Proof. Let D be an infinite oriented graph that needs its whole vertex set to be identified. Let x be a vertex of D. The set $V(D) - \{x\}$ is not an identifying code. Either x is not dominated and so, x is a source or there is a pair of vertices, say u and v, such that $N^+[v] = N^+[u] \cup \{x\}$. If $x \neq v$, we must have $\overrightarrow{uv} \in A$ and $\overrightarrow{vu} \in A$. Since D has no symmetric arc, this is not possible so, necessarily, $x = v$, $N^+[x] = N^+[u] \cup \{x\}$ and u is the only vertex such that $N^+[x] = N^+[u] \cup \{x\}$. So for any vertex x of D which is not a source there is a unique vertex we call x_{-1} such that $N^+[x] = N^+[x_{-1}] \cup \{x\}$. We may repeat this argument on x_{-i} to get x_{-i-1} for $i = 1, 2, \cdots$ as long as x_{-i} is not a source. This will result in a well defined set $\{\cdots, x_{-i}, \cdots, x_0 = x\}$ which induces a transitive closure of a finite or infinite path, which we denote by P_x (if x is a source itself, then $P_x = \{x\}$).

Assume that for two vertices x and y, $P_x \cap P_y \neq \emptyset$. Then let x_i be the first (in the order defined by the path P_x) vertex of P_x in $P_x \cap P_y$. We have $P_x \cap P_y = P_{x_i}$.

We now define an equivalence relation on the vertices of D: $x \equiv y$ if and only if $P_x \cap P_y \neq \emptyset$. This gives us the equivalence class of x: $T_x = \{y \in V(D) | x \equiv y\}$. The set of vertices of T_x induces either an f.s.t. tree or an i.s.t. tree in D. Furthermore, if $u \notin T_x$, then $\overrightarrow{uv}$ is an arc of D for either every $v \in V(T_x)$ or no $v \in V(T_x)$. In fact, if there is an arc $\overrightarrow{uy}$ in D for $y \in T_x$, then if $z \in T_x$ we have $P_y \cap P_z = P_{y_i}$. But $N^+[y] = N^+[t] \cup \{y, y_{-1}, ..., y_{i+1}\}$ so $u \in N^+[t]$. Now, $N^+[t] \subset N^+[z]$ so $u \in N^+[z]$ and $\overrightarrow{uz}$ is also an arc of D.

We construct a graph H as follows: the vertices of H are the equivalence classes T_x and there is an arc $\overrightarrow{T_xT_y}$ if there is an arc $\overrightarrow{uv}$ of D such that $u \in V(T_x)$ and $v \in V(T_y)$. It is now clear that $D \in \Psi(H)$. ☆

We conclude this section by the following remarks:

- The proof of Theorem 3.15 also works for the classification of finite oriented graphs with $\overrightarrow{\gamma}^{\text{ID}}(D) = |V(D)|$. But the characterization of Theorem 3.8 is for all digraphs, thus it is a stronger statement.

- The oriented graphs for which the only separating code is their whole vertex set are graphs in $\Psi(H)$ as long as H has no source vertex.

3.4 Undirected graphs having as identifying code number their order minus one

In this section, we characterize all graphs G for which $\gamma^{\mathrm{ID}}(G) = |V(G)|-1$. As already mentioned, it was known that $\gamma^{\mathrm{ID}}(G) = |V(G)|$ if and only if $G \cong \overline{K_n}$. Graphs with identifying code number $|V(G)| - 1$ were already studied in [45, 179]. Some of them where known: the star $K_{1,n-1}$ ($n \geq 3$) [45], the complete graph minus the edges of a maximum matching M, $K_n \setminus M$ ($n \geq 3$) [45], and the complete join of at least one copy of P_4 with a possible additional universal vertex [179]. Moreover, two conjectures had been made about them:

Conjecture 3.16 ([45, Conjecture 1])**.** *The only connected graphs with n vertices such that there is a minimum identifying code with size $n-1$ are the star and the complete graph minus a maximum matching.*

Conjecture 3.17 ([179, Conjecture 3.13])**.** *If $\gamma^{ID}(G) = |V(G)|-1$, then either G is the complete join of at least one copy of P_4 with a possible additional universal vertex, or $\overline{G}$ is not identifiable.*

In the following, we disprove both these conjectures by providing new examples of graphs with identifying code number $|V(G)| - 1$ (whose complement is identifiable); special powers of paths are the basic examples of such graphs. We then prove that any other example is mainly obtained from the complete join of some of these basic elements.

3.4.1 Preliminary tools

The following proposition is useful when considering the join operation between two graphs. We recall that $\gamma^{\mathrm{S}}(G)$ denotes the separating code number of an identifiable graph G.

Proposition 3.18. *Let G_1 and G_2 be identifiable graphs such that for every minimum separating code S there is an S-universal vertex. If $G_1 \bowtie G_2$ is identifiable, then we have $\gamma^{S}(G_1 \bowtie G_2) = \gamma^{S}(G_1) + \gamma^{S}(G_2) + 1$. Furthermore, if S is a separating code of size $\gamma^{S}(G_1) + \gamma^{S}(G_2) + 1$ of $G_1 \bowtie G_2$, then there is an S-universal vertex.*

Proof. Let S be a minimum separating code of $G_1 \bowtie G_2$. Since vertices of G_2 do not separate any pair of vertices in G_1 then $S \cap V(G_1)$ is a separating code of G_1. By the same argument $S \cap V(G_2)$ is a separating code of G_2. Therefore, $|S| \geq \gamma^{\mathrm{S}}(G_1) + \gamma^{\mathrm{S}}(G_2)$. But if $|S| = \gamma^{\mathrm{S}}(G_1) + \gamma^{\mathrm{S}}(G_2)$, then there is an $[S \cap V(G_1)]$-universal vertex x in G_1 and an $[S \cap V(G_2)]$-universal vertex y in G_2. But then, x and y are not separated by S.

Given a separating code S_1 of G_1 and a separating code S_2 of G_2, the set $S_1 \cup S_2$ separates all pairs of vertices except the S_1-universal vertex of G_1 from the S_2-universal vertex of G_2. But since $G_1 \bowtie G_2$ is identifiable, we could add one more vertex to $S_1 \cup S_2$ to obtain a separating code of $G_1 \bowtie G_2$ of size $\gamma^{\mathrm{S}}(G_1) + \gamma^{\mathrm{S}}(G_2) + 1$.

For the second part assume S is a separating code of size $\gamma^{\mathrm{S}}(G_1) + \gamma^{\mathrm{S}}(G_2) + 1$ of $G_1 \bowtie G_2$. Then we have either $|S \cap V(G_1)| = \gamma^{\mathrm{S}}(G_1)$ or $|S \cap V(G_2)| = \gamma^{\mathrm{S}}(G_2)$. Without loss of generality assume the former. Then there is an $[S \cap V(G_1)]$-universal vertex z of G_1. Since z is also adjacent to all the vertices of G_2, it is an S-universal vertex of $G_1 \bowtie G_2$. ☆

We remark that in Proposition 3.18, if $G_1 \not\cong K_1$ and $G_2 \not\cong K_1$, then $\gamma^{\mathrm{ID}}(G_1 \bowtie G_2) = \gamma^{\mathrm{S}}(G_1 \bowtie G_2) = \gamma^{\mathrm{S}}(G_1) + \gamma^{\mathrm{S}}(G_2) + 1$.

We will also need the following strengthening for the case $|S| = 1$ of Proposition 3.2 for identifying codes in connected undirected graphs:

Corollary 3.19. *Let G be a connected identifiable graph with $\gamma^{ID}(G) = |V(G)| - 1$, $G \not\cong K_{1,2}$, then there is a vertex x of G such that $G - x$ is still connected and $\gamma^{ID}(G - x) = |V(G - x)| - 1$.*

Proof. If $G \cong K_{1,n-1}$, $n \neq 3$, then any leaf vertex works. Thus, we may suppose $G \not\cong K_{1,n-1}$. Then by Theorem 2.27, there is a vertex x of G such that $V(G - x)$ is an identifying code of G and thus $G - x$ is identifiable and $G - x \not\cong \overline{K_n}$. By Proposition 3.2, we have $\gamma^{\mathrm{ID}}(G - x) \geq$

$\gamma^{\text{ID}}(G) - 1 = |V(G - x)| - 1$. Equality holds since otherwise $\gamma^{\text{ID}}(G) = |V(G)|$. To complete the proof, we show that x can be chosen such that $G - x$ is connected. To see this, assume $G - x$ is not connected. Since $\gamma^{\text{ID}}(G - x) = |V(G - x)| - 1$, except one component, every component of $G-x$ is an isolated vertex. If there are two or more such isolated vertices, then either one of them can be the vertex we want. Otherwise there is only one isolated vertex, call it y. Now if $G - y$ is identifiable, then y is the desired vertex, else there is a vertex x' such that $N[x'] = N[x] - y$. Then $G - x'$ is connected and identifiable. ☆

The following lemma will also be useful.

Lemma 3.20. *Let G be an identifiable graph and let $v \in V(G)$ be a vertex forced by a pair x, y of vertices of G. If x or y are also forced by some pair, then v must be one of the vertices of this pair.*

Proof. Since v separates x and y, it is adjacent to one of them (say x) and not to the other. Suppose z, t are twins in $G - x$. Suppose z is adjacent to x and t is not. If $z \neq v$ then y is also adjacent to z and, therefore, t is also adjacent to y which implies x being adjacent to t. This contradicts the fact that x separates z and t. The other case is proved similarly. ☆

3.4.2 New constructions

Definition 3.21. *For an integer $k \geq 1$, let $A_k = (V_k, E_k)$ be the graph with vertex set $V_k = \{x_1, \ldots, x_{2k}\}$ and edge set $E_k = \{x_i x_j \mid |i - j| \leq k - 1\}$.*

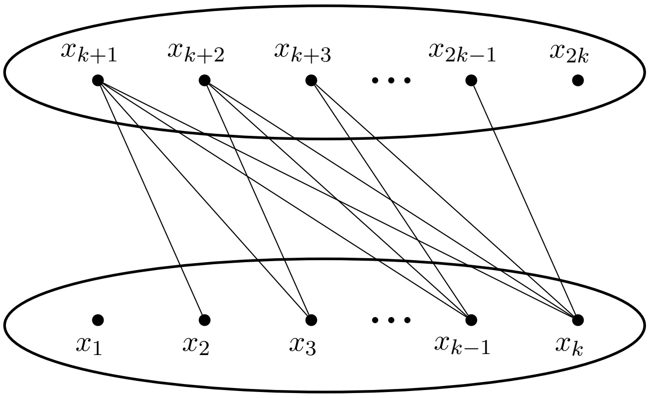


Figure 3.5: The graph A_k which needs $|V(A_k)| - 1$ vertices for any identifying code

We note that for $k \geq 2$ we have $A_k = P_{2k}^{k-1}$ (hence $A_2 = P_4$) and $A_1 = \overline{K_2}$. It is also easy to check that the only nontrivial automorphism of A_k is the mapping $x_i \to x_{2k+1-i}$. It is not hard to observe that A_k is identifiable, $\Delta(A_k) = 2k - 2$ and that A_k and $\overline{A_k}$ are connected if $k \geq 2$.

Proposition 3.22. *For $k \geq 1$, we have $\gamma^{S}(A_k) = 2k - 1$ with $N[x_k]$ and $N[x_{k+1}]$ being the only separating codes of size $2k - 1$ of A_k. Furthermore, if $k \geq 2$, $\gamma^{ID}(A_k) = 2k - 1$.*

Proof. Let $\mathcal{S}$ be a separating code of A_k. For $i < k$, we have $N[x_i] \ominus N[x_{i+1}] = \{x_{i+k}\}$ and for $k < i \leq 2k - 1$, we have $N[x_i] \ominus N[x_{i+1}] = \{x_{i-k+1}\}$. Thus, $\{x_2, \ldots, x_{2k-1}\} \subset \mathcal{S}$. But to separate x_k and x_{k+1}, we must add x_1 or x_{2k}. It is now easy to see that $V_k \setminus \{x_1\} = N[x_{k+1}]$ and $V_k \setminus \{x_{2k}\} = N[x_k]$, each is a separating code of size $2k - 1$. If $k \geq 2$, then they both dominate A_k and therefore are also identifying codes. ☆

In the previous proof in fact we have also proved that:

Corollary 3.23. *For $k \geq 1$ every minimum separating code $\mathcal{S}$ of A_k has an $\mathcal{S}$-universal vertex.*

Let $\mathcal{A}$ be the closure of $\{A_i \mid i = 1, 2, \ldots\}$ with respect to the complete join operation $\bowtie$. It is shown below that elements of $\mathcal{A}$ are also extremal graphs with respect to both separating codes and identifying codes.

Proposition 3.24. *For every graph $G \in \mathcal{A}$, we have $\gamma^S(G) = |V(G)| - 1$. Furthermore, every minimum separating code $\mathcal{S}$ of G has an $\mathcal{S}$-universal vertex.*

Proof. The proposition is true for basic elements of $\mathcal{A}$ by Proposition 3.22 and by Corollary 3.23. For a general element $G = G_1 \bowtie G_2$ it is true by Proposition 3.18 and by induction. ☆

Corollary 3.25. *If $G \in \mathcal{A}$ and $G \ncong A_1$, then $\gamma^{ID}(G) = |V(G)| - 1$.*

Further examples of graphs extremal with respect to separating codes and identifying codes can be obtained by adding a universal vertex to each of the graphs in $\mathcal{A}$, as we prove below. We let $\mathcal{A} \bowtie K_1$ denote the set of all graphs of $\mathcal{A}$ with an additional universal vertex.

Proposition 3.26. *For every graph G in $\mathcal{A} \bowtie K_1$ we have $\gamma^{ID}(G) = \gamma^S(G) = |V(G)| - 1$.*

Proof. Assume $G = G_1 \bowtie K_1$ with $G_1 \in \mathcal{A}$, and assume u is the vertex corresponding to K_1. Suppose $\mathcal{S}$ is a minimum separating code of G. We first note that since $\mathcal{S} \cap V(G_1)$ is a separating code of G_1, we have $|\mathcal{S} \cap V(G_1)| \geq |V(G_1)| - 1$. But if $|\mathcal{S} \cap V(G_1)| = |V(G_1)| - 1$, then by Proposition 3.24, there is an $[\mathcal{S} \cap V(G_1)]$-universal vertex y of G_1. Then y is not separated from x. Thus $|\mathcal{S} \cap V(G_1)| = |V(G_1)|$ and therefore $\mathcal{S} = V(G_1)$. It is easy to check that $\mathcal{S}$ is also an identifying code. ☆

It was proved in [45] that $\gamma^{\mathrm{ID}}(K_n \setminus M) = n - 1$ where $K_n \setminus M$ is the complete graph minus a maximal matching. We note that this graph, for even values of n, is the join of $\frac{n}{2}$ disjoint copies of A_1, thus it belongs to $\mathcal{A}$. For odd values of n, it is built from the previous graph by adding a universal vertex.

So far we have seen that $\gamma^{\mathrm{ID}}(G) = |V(G)| - 1$ for $G \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$, $G \ncong A_1$. We also know that $\gamma^{\mathrm{ID}}(\overline{K}_n) = n$. More examples of graphs with $\gamma^{\mathrm{ID}}(G) = |V(G)| - 1$ can be obtained by adding isolated vertices. In the next subsection we show that for any other identifiable graph G we have $\gamma^{\mathrm{ID}}(G) \leq |V(G)| - 2$.

3.4.3 The characterization

Theorem 3.27. *Given a connected graph G, we have $\gamma^{ID}(G) = |V(G)| - 1$ if and only if $G \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$ and $G \ncong A_1$.*

Proof. The "if" part of the theorem is already proved. The proof of the "only if" part is based on induction on the number of vertices of G. For graphs on at most 4 vertices this is easy to check. Assume the claim is true for graphs on at most $n-1$ vertices and, by contradiction, let G be an identifiable graph on $n \geq 5$ vertices such that $\gamma^{\mathrm{ID}}(G) = n - 1$ and $G \notin \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$.

By Corollary 3.19 there is a vertex $x \in V(G)$ such that $G - x$ is connected and $\gamma^{\mathrm{ID}}(G - x) = |V(G - x)| - 1$. By the induction hypothesis we have $G - x \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$. Depending on which one of these three sets $G - x$ belongs to, we will have three cases.

Case a. $G - x \in \{K_{1,n-1} \mid n \geq 3\}$. In this case we consider a minimum identifying code $\mathcal{C}$ of $G - x$. If $\mathcal{C}$ does not already identify x then either $deg(x) \leq 3$ or $deg(x) \geq n - 2$. We leave it to the reader to check that in each of these cases, there is an identifying code of size $n - 2$.

Case b. $G - x \in \mathcal{A}$. We consider two sub-cases. Either $G - x \cong A_k$ for some k or $G - x = G_1 \bowtie G_2$, with $G_1, G_2 \in \mathcal{A}$.

(1) $G - x \cong A_k$, for some $k \geq 2$. If x is adjacent to all the vertices of $G - x$, then $G \in \mathcal{A} \bowtie K_1$ and we are done. Otherwise there is a pair of consecutive vertices of A_k, say x_i and x_{i+1}, such that one is adjacent to x and the other is not. By the symmetry of A_k we may assume $i \leq k$. We claim that one of the sets $\mathcal{C} = V(G) \setminus \{x_1, x\}$, $\mathcal{C}' = V(G) \setminus \{x_{2k}, x\}$ or $\mathcal{C}'' = V(G) \setminus \{x_k, x_{k+1}\}$ is an identifying code of G. This would contradict our assumption.

Note that for each of the sets $\mathcal{C}$, $\mathcal{C}'$ and $\mathcal{C}''$, the vertices of $V(G-x)$ are all separated. If x is also separated from all the vertices of $G-x$ then we are done. Otherwise there will be two possibilities.

First we consider the possibility: x is not adjacent to some x_i, but adjacent to x_{i+1}. Consider the set $\mathcal{C}$. Each vertex x_j, $j > i+k$, is separated from x by x_{i+1} and each vertex x_j, $j < i+k$, is separated from x by x_i. Thus x is not separated from x_{i+k} and therefore x must be adjacent to $x_{i+1}, \ldots, x_{2k-1}$. On the other hand, if $\mathcal{C}'$ is not an identifying code of G, it means that x is not separated only from x_k (it is the only vertex adjacent to x_{2k-1} but not to x_{2k}). Therefore, x is adjacent to exactly $x_2, \ldots, x_{2k-1}$. In this case $\mathcal{C}''$ is an identifying code of G. Indeed, x is separated from all other vertices of G. Since x_k and x_{k+1} were needed only to separate x_1 from x_2 and x_{2k-1}, x_{2k}, respectively, x is now separating these pairs.

In the other possibility, x is adjacent to x_i but not adjacent to x_{i+1}. If we consider C, a similar argument implies that x is separated from every vertex but x_{2k}. Then $\mathcal{C}'$ is an identifying code.

(2) $G - x \cong G_1 \bowtie G_2$ with $G_1, G_2 \in \mathcal{A}$. If x is adjacent to all the vertices of $G-x$, then $G \in \mathcal{A} \bowtie K_1$ and we are done. Thus there is a vertex, say y, that is not adjacent to x. Without loss of generality, we can assume $y \in V(G_1)$. Let $\mathcal{C}_1$ be an identifying code of size $\gamma^{\mathrm{ID}}(G_1) = |V(G_1)| - 1$ of G_1 which contains y. The existence of such an identifying code becomes apparent from the proof of Proposition 3.24. Then $\mathcal{C} = \mathcal{C}_1 \cup V(G_2)$ is an identifying code of $G_1 \bowtie G_2$ of size $|V(G_1 \bowtie G_2)| - 1 = |V(G)| - 2$. Thus $\mathcal{C}$ does not separate a vertex of $G_1 \bowtie G_2$ from x. Call this vertex z. Since $y \in \mathcal{C}$, z is not adjacent to y, hence $z \in V(G_1)$. Therefore, z is adjacent to all the vertices of G_2. So x should also be adjacent to all the vertices of G_2. Thus we have $G = (G_1 + x) \bowtie G_2$ and any minimum identifying code of $G_1 + x$ together with all vertices of G_2 would form an identifying code of G. This proves that $\gamma^{\mathrm{ID}}(G_1 + x) = |V(G_1 + x)| - 1$. Since $G_1 + x$ has less vertices than G, by induction hypothesis, we have $G_1 + x \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$ and $G \ncong A_1$. Since $G_1 \in \mathcal{A}$, and since x is not adjacent to a vertex of G_1, we should have $G_1 + x \in \mathcal{A}$ but all graphs in $\mathcal{A}$ have an even number of vertices and this is not possible.

Case c. $G - x \in \mathcal{A} \bowtie K_1$. Suppose $G - x \cong A_{i_1} \bowtie A_{i_2} \bowtie \ldots \bowtie A_{i_j} \bowtie K_1$ and let u be the vertex corresponding to K_1.

If x is also adjacent to u, then u is a universal vertex of G and $G-u$ is also identifiable. In this case we apply the induction on $G-u$: by Proposition 3.1, $\gamma^{\mathrm{ID}}(G-u) = |V(G-u)| - 1$ and by induction hypothesis $G - u \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$. But if $G - u \in \{K_{1,n-1} \mid n \geq 3\} \cup (\mathcal{A} \bowtie K_1)$, there will be two universal vertices, and therefore twins. Thus $G - u \in \mathcal{A}$ and $G \in \mathcal{A} \bowtie K_1$.

We now assume x is not adjacent to u and we repeat the argument with $G-u$ if it is identifiable. In this case if $G - u \in \{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A}$, we apply Case a or Case b. If $G - u \in \mathcal{A} \bowtie K_1$ with u' being the vertex of K_1, then u and u' induce an isomorphic copy of A_1 and $G \in \mathcal{A}$.

If $G-u$ is not identifiable then, by Lemma 3.20, x must be one of the twin vertices. Let x' be its twin and suppose $x' \in V(A_{i_1})$ with $V(A_{i_1}) = \{z_1, z_2, \ldots, z_{2k}\}$. Without loss of generality we may assume $x' = z_l$ with $l \leq k$. If $l \geq 2$, then we claim $\mathcal{C} = V(G) \setminus \{z_l, z_{2k}\}$ is an identifying code of G which is a contradiction. To prove our claim notice first that vertices of $A_{i_2} \bowtie \cdots \bowtie A_{i_j}$ are already identified from each other and from the other vertices. Now each pair of vertices of A_{i_1} is separated by a vertex in $V(A_{i_1}) \cap \mathcal{C}$ except z_{l+k-1} and z_{l+k} which are separated by x. The vertex x is also separated from all the other vertices by u. It remains to show that u is separated from vertices of A_{i_1}. It is separated from vertices in $\{z_1, \ldots, z_{l+k-1}\}$ by x and from $\{z_{k+1}, \ldots, z_{2k}\}$ by z_1 ($l \geq 2$). Thus $x' = x_1$ and now it is easy to see that the subgraph induced by $V(A_{i_1})$, u and x is isomorphic to A_{i_1+1} and, therefore, $G \cong A_{i_1+1} \bowtie A_{i_2} \bowtie \ldots \bowtie A_{i_j}$. ☆

3.4.4 Tightness of the bound of Theorem 2.27 in various graph classes

As we have classified all graphs reaching this bound, we can determine for which graph classes the bound $\gamma^{\mathrm{ID}}(G) \leq |V(G)|-1$ is tight. We can observe that the star is a tree, and therefore it is also outerplanar, series-parallel, planar, chordal, bipartite. One can check that each graph A_k is a unit interval graph and a permutation graph [80], and that the graphs $P_3, P_4, C_4, P_4 \bowtie K_1, C_4 \bowtie K_1$ and $K_6 \setminus M$ are the only line graphs in the extremal family.

Corollary 3.28. *There are infinitely many trees (and therefore chordal graphs, bipartite graphs, series-parallel graphs, (outer)planar graphs), unit interval graphs (and therefore interval graphs, (un)directed path graphs and induced claw-free graphs) and permutation graphs G such that $\gamma^{ID}(G) = |V(G)| - 1$. There are exactly six such line graphs.*

3.4.5 Answering two questions from the literature

The following two questions were asked by D. Skaggs in his PhD thesis [179]:

1. Do there exist k-regular graphs G of order n with $\gamma^{\mathrm{ID}}(G) = n - 1$ for $k < n - 2$?
2. Do there exist graphs G of odd order n and maximum degree $\Delta < n-1$ with $\gamma^{\mathrm{ID}}(G) = n-1$?

As a corollary of Theorem 3.27, we can now answer these questions in the negative. Indeed, for the first question, if G is a k-regular ($k \geq 2$) graph of order n with $\gamma^{\mathrm{ID}}(G) = n - 1$ then G is the join of k disjoint copies of A_1. For the second question, noting that each graph in $\mathcal{A}$ has an even order, we conclude that if a graph G on an odd number n of vertices has $\gamma^{\mathrm{ID}}(G) = n - 1$, then $G \in \{K_{1,n-1} \mid n \geq 3\} \cup (\mathcal{A} \bowtie K_1)$ and, therefore $\Delta(G) = n - 1$.

3.5 Infinite undirected graphs with their whole vertex set as only identifying code

It is shown in [45] that Theorem 2.27 does not have a direct extension to the family of infinite graphs. In other words, there are nontrivial examples of identifiable infinite graphs requiring the whole vertex set for any identifying code. The basic example of such infinite graphs, originally defined in [45], is given in Subsection 3.5.1. In Subsection 3.5.2, we classify all such infinite graphs. This strengthens a theorem of [96] (see Theorem 2.28), which claims that there are no such infinite graphs in which all vertices have finite degrees.

3.5.1 A family of infinite extremal graphs

Definition 3.29. *Let $X = \{\dots, x_{-1}, x_0, x_1, \dots\}$ and $Y = \{\dots, y_{-1}, y_0, y_1, \dots\}$. $A_\infty = (X \cup Y, E)$ is the graph on $X \cup Y$ having edge set $E = \{x_i x_j \mid i \neq j\} \cup \{y_i y_j \mid i \neq j\} \cup \{x_i y_j \mid i < j\}$.*

See Figure 3.6 for an illustration.

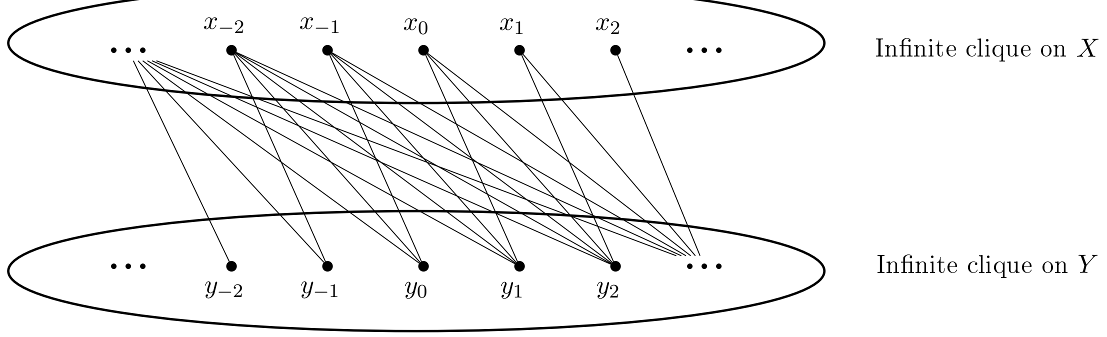


Figure 3.6: The graph A_∞ which needs all its vertices for any identifying code

It is shown in [45] that the only separating code of A_∞ is $V(A_\infty)$. One should note that the graph induced by $\{y_1, y_2, \ldots, y_k, x_1, x_2 \ldots, x_k\}$ is isomorphic to the graph A_k.

Before introducing our theorem let us see again why every separating code of A_∞ needs the whole vertex set: for every i, x_i and x_{i+1} are only separated by y_{i+1}, while y_i and y_{i+1} are separated only by x_i.

This property would still hold if we add a new vertex which is adjacent either to all vertices in X (similarly in Y) or to none. This leads to the following family:

Let H be a finite or infinite simple graph with a perfect matching ρ, that is a mapping $x \to \rho(x)$ of $V(H)$ to itself such that $\rho^2(x) = x$ and $x\rho(x)$ is an edge of H. We define $\Psi(H, \rho)$ to be the graph built as follows: for every vertex x of H we assign $\Phi(x) = \{\ldots x_{-1}, x_0, x_1, \ldots\}$. The vertex set of $\Psi(H, \rho)$ is $\bigcup\limits_{x \in V(H)} \Phi(x)$. For each edge $x\rho(x)$ of H we build a copy of A_∞ on $\Phi(x) \cup \Phi(\rho(x))$ and for every other edge xy of H we join every vertex in $\Phi(x)$ to every vertex in $\Phi(y)$. An example of such construction is illustrated in Figure 3.7.

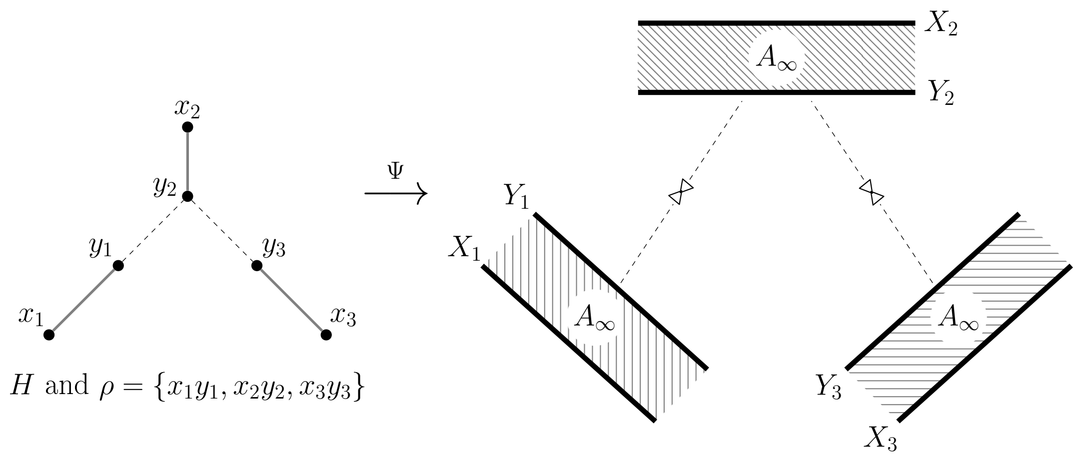


Figure 3.7: Construction of $\Psi(H, \rho)$ from (H, ρ)

We now have:

Proposition 3.30. *For every simple, finite or infinite, graph H with a perfect matching ρ, the graph $\Psi(H, \rho)$ can only be identified with $V(\Psi(H, \rho))$.*

Proof. Let A_x be the copy of A_∞ which corresponds to the edge $x\rho(x)$. Then for every vertex y in $V(\Psi(H, \rho)) \setminus V(A_x)$, either y is connected to every vertex in A_x or to neither of them. Thus to separate vertices in A_x, we need all the vertices of A_x. Since x is arbitrary, we need all the vertices in $V(\Psi(H, \rho))$ in any separating code. ☆

3.5.2 The characterization

In the next theorem, we prove that every extremal connected infinite graph is $\Psi(H, \rho)$ for some connected finite or infinite graph H together with a matching ρ.

Theorem 3.31. *Let G be an infinite connected graph. Then G admits only $V(G)$ as an identifying code if and only if $G \cong \Psi(H, \rho)$ for some finite or infinite graph H together with a perfect matching ρ.*

Proof. We already have seen that if $G \cong \Psi(H, \rho)$, then the only identifying code of G is $V(G)$. To prove the converse suppose $G - v$ has a pair of twin vertices for every vertex v of G. It is enough to show that every vertex v of G belongs to a unique induced subgraph A_v of G isomorphic to A_∞ and that if a vertex not in A_v is adjacent to a vertex in the X (respectively, Y) part of A_v then it is adjacent to all the vertices of the X (respectively, Y).

Let x_1 be a vertex of G. The subgraph $G - x_1$ has a pair of twins, let y_1 and y_2 be one such pair. Assume, without loss of generality, that x_1 is adjacent to y_2 and not to y_1. By Lemma 3.20, x_1 must be one of the vertices of a pair of twins in $G - y_1$. Let the other be x_2. Now consider the subgraph $G - y_1$. This subgraph must have a pair of twins and x_1 must be one of them. Let x_0 be the other one.

Continuing this process in both directions (with negative and positive indices) we build our $A_{x_1} \cong A_\infty$ as a subgraph of G. Since each consecutive pair of vertices in $X \subset A_{x_1}$ is separated only by a vertex in $Y \subset A_{x_1}$, every pair of vertices in X are twins in $G - Y$. Thus each vertex not in A_{x_1}, either is adjacent to all the vertices in X or to none of them. Similarly, every vertex in A_{x_1}, either is adjacent to all the vertices in Y or to none. Hence A_{x_1} is unique. This proves the theorem. ☆

3.6 Conclusion

In this chapter, we fully characterized the finite digraphs, infinite oriented graphs and infinite undirected graphs that have their whole vertex set as only identifying code. These characterizations are rich and nontrivial. However, we note that we have only characterized infinite *oriented* graphs having their whole vertex set as unique identifying code, giving rise to the following question.

Question 3.32. *Is it possible to give a nice characterization of all infinite digraphs having only their whole vertex set as an identifying code?*

The characterization for finite digraphs also led to an interesting application to Bondy's theorem. Moreover, it gives a nice corollary, by noticing that digraphs in $(K_1, \oplus, \overrightarrow{\lhd})$ have no symmetric arcs. Indeed, we have the following extension of Theorem 2.27, a direct consequence of Theorem 3.8:

Corollary 3.33. *Let D be a finite identifiable digraph having some symmetric arcs. Then $\overrightarrow{\gamma^{ID}}(D) \leq |V(D)| - 1$.*

Regarding finite undirected graphs, the class of graphs having their whole vertex set as unique identifying code consists only of all edgeless graphs. We however fully characterized those finite undirected graphs that have their order minus one as identifying code number, providing new constructions and thereby answering several questions from the literature. A further question to study could be the following one, even though we suspect it might not lead to a nice classification.

Question 3.34. *What can be said about graphs having their order minus* two *as minimum size of an identifying code?*

We note that the graphs with location-domination number equal to their order minus two have been recently characterized in [34].[3]

We make an interesting observation regarding our classification. Since every graph on n vertices from the family $\{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$ has maximum degree $n-2$, we have:

Corollary 3.35. *Let G be an identifiable connected graph on $n \geq 3$ vertices and maximum degree $\Delta \leq n-3$. Then $\gamma^{ID}(G) \leq n-2$.*

We remark that the graphs of $\{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$ can be recognized in polynomial time. Indeed, it is easy to check whether a graph is a star or a path power from $\mathcal{A}$. Moreover, one can check in polynomial time whether a graph is a complete join of two other graphs, and whether it has a universal vertex. Similarly, checking whether a digraph is the closure of a top-down rooted oriented tree can be done in polynomial time by comparing the out-degrees of all vertices.

[3]However, the set of graphs having location-domination number equal to their order minus one is much easier to describe than for the case of the identifying code number, as this set contains only the complete graphs and the stars [182].

The case of r-identifying codes is much studied in the literature. In order to give characterizations analogous to our results, one should find the r-roots of the graphs of our classifications. We tried to solve this question for the case of the graphs of $\{K_{1,n-1} \mid n \geq 3\}\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$, but it does not seem to be an easy task. In particular, if s divides $k-1$ and $r = \frac{k-1}{s}$, then the graph $G = P_{2k}^s$ is one of the r-roots of A_k. It is easy to see that, in most cases, one can remove many edges of G and still have $G^r \cong A_k$. The difficulty of the problem is that an r-root of A_k is not necessarily a subgraph of P_{2k}^s. An example of such a 2-root of A_5 is given in Figure 3.8.

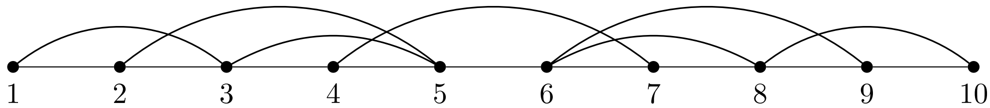


Figure 3.8: A 2-root of A_5 which is not a subgraph of P_{10}^2

Question 3.36. *What can be said about finite undirected graphs having their minimum r-identifying code of size their order minus one?*

For the case of infinite graphs, we note that there exists a 2-root of A_∞. This graph is defined as follows: it has the same vertex set $X \cup Y$ as A_∞ and the same edges between X and Y, but no edges within X or Y. However, we do not know whether there exist other roots of graphs described in Theorem 3.31.

Question 3.37. *What can be said about infinite graphs and digraphs having as only r-identifying code their whole vertex set?*

We conclude by relating our work to the case of discriminating codes. The extremal cases we have studied in Section 3.2, where a minimum separating code does always give an undominated vertex, is equivalent to the one where the only minimum discriminating code of a set system $(\mathcal{I}, \mathcal{A})$ consists in the whole set $\mathcal{A}$. Therefore Theorem 3.11 can be stated in the language of discriminating codes (recall that the closed out-neighbourhood hypergraph of a digraph D is the hypergraph with vertex set $V(D)$ and whose edges are all closed out-neighbourhoods in D):

Corollary 3.38. *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system with $|\mathcal{I}| = |\mathcal{A}|$. A minimum discriminating code $\mathcal{C} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$ has size $|\mathcal{A}|$ if and only if $(\mathcal{I}, \mathcal{A})$ is isomorphic to the closed out-neighbourhood hypergraph of some digraph in $(K_1, \oplus, \overrightarrow{\lhd})$.*

We remark that the class of set systems $(\mathcal{I}, \mathcal{A})$ in which any discriminating code has size at least $|\mathcal{A}|$ are classified in [36] in terms of trees obtained by observing the structure of symmetric differences between sets of $\mathcal{A}$. However, we believe that our classification is more explicit. The authors of [36] further asked the following question:

Question 3.39 ([36]). *Which are the set systems $(\mathcal{I}, \mathcal{A})$ for which every discriminating code has at least $|\mathcal{A}| - 1$ elements of $\mathcal{A}$?*

In Theorem 3.27, we answered this question for those set systems that are isomorphic to a closed neighbourhood hypergraph of some identifiable graph.

Corollary 3.40. *Let $(\mathcal{I}, \mathcal{A})$ be an $\mathcal{I}$-identifiable set system with $|\mathcal{I}| = |\mathcal{A}|$ which is isomorphic to the closed neighbourhood hypergraph of some identifiable graph G. Then a minimum discriminating code $\mathcal{C} \subseteq \mathcal{A}$ of $(\mathcal{I}, \mathcal{A})$ has size $|\mathcal{A}| - 1$ if and only if G belongs to the family $\{K_{1,n-1} \mid n \geq 3\} \cup \mathcal{A} \cup (\mathcal{A} \bowtie K_1)$ $(G \ncong A_1)$ defined in Section 3.4.2.*

However, to our knowledge, Question 3.39 remains open for the other cases.

Chapter 4
Identifying codes in graphs of given maximum degree

IN this chapter, we study the influence of the maximum degree $\Delta(G)$ on the identifying code number $\gamma^{\mathrm{ID}}(G)$ of an identifiable graph G.

In Section 4.1, we give a full characterization of all graphs G reaching the lower bound $\gamma^{\mathrm{ID}}(G) \geq \frac{2|V(G)|}{\Delta(G)+2}$ from Theorem 2.29. These results are an improved version from part of the author's master thesis [Fo09].

In Section 4.2, we discuss upper bounds on the identifying code number depending solely on the order and the maximum degree of the graph. To the best of our knowledge, a similar discussion cannot be found in the literature. In this regard, we conjecture in Conjecture 4.4 that for any connected identifiable graph G, the bound $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Delta} + c$ holds (for some constant c independent of n and Δ). We further discuss the tightness of this conjectured bound by providing constructions of families of graphs reaching it. We also study the structure of false twins and forced vertices in a graph, a study related to the conjecture.

In Section 4.3, we introduce a new technique to provide upper bounds that approximate the bound of Conjecture 4.4. This technique uses the construction of independent sets having a few special properties; taking the complement of this set yields an identifying code. We first introduce this idea by giving an easy bound of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Theta(\Delta^5)}$ in Subsection 4.3.1 (Theorem 4.24). We then refine the general idea in Subsection 4.3.2 to provide a general framework (Proposition 4.25) to build an identifying code using a (special) independent set. We apply this technique to get an improved bound of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Theta(\Delta^4)}$ (Theorem 4.27). Subsequently, in Subsection 4.3.3, these ideas are further refined for the special case of triangle-free graphs. We prove a bound of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Delta(1+o_\Delta(1))}$ for general triangle-free graphs, and of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{o(\Delta)}$ for triangle-free graphs without false twins (Corollary 4.40).

Finally, we use the probabilistic method (in form of a combination of L. Lovász' Local Lemma together with the Chernoff bound) in Section 4.4 to prove a bound of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Theta(\Delta)}$ for graphs having few forced vertices, which includes regular graphs (Corollary 4.50), and a bound of the form $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Theta(\Delta^3)}$ valid for all graphs (Corollary 4.51).

The results of this chapter have appeared in [FGK+11] for Subsection 4.3.1 (joint work with E. Guerrini, M. Kovše, R. Naserasr, A. Parreau and P. Valicov), [FKKR12] for Subsubsection 4.2.3.1, Subsection 4.3.3 and Conjecture 4.4 (joint work with R. Klasing, A. Kosowski and A. Raspaud), [FP12] for Subsection 4.2.2, Subsubsection 4.2.3 and for Section 4.4 (joint work with G. Perarnau).

4.1 Graphs reaching the lower bound of Theorem 2.29

We recall the bound $\gamma^{\text{ID}}(G) \geq \frac{2|V(G)|}{\Delta(G)+2}$ that was already mentioned in Chapter 2 (Theorem 2.29). This result was first proved in [131]; we give an easy proof (similar to the one of [131]) which will help us to characterize all graphs that reach this bound.

Proof of Theorem 2.29. Let G be an identifiable graph on n vertices with maximum degree Δ, and let $\mathcal{C} \subseteq V(G)$ be an identifying code of G of cardinality k. We partition $V(G)$ into the four following sets, as shown in Figure 4.1:

$$\mathcal{C}_1 = \Big\{x \in \mathcal{C} \mid |N[x] \cap \mathcal{C}| = 1\Big\}, \mathcal{C}_2 = \Big\{x \in \mathcal{C} \mid |N[x] \cap \mathcal{C}| \geq 2\Big\}$$

$$\mathcal{N}_1 = \Big\{v \in V \setminus \mathcal{C} \mid |N[v] \cap \mathcal{C}| = 1\Big\}, \text{ and } \mathcal{N}_2 = \Big\{v \in V \setminus \mathcal{C} \mid |N[v] \cap \mathcal{C}| \geq 2\Big\}.$$

It is easy to see that the following (in)equalities hold:

$$|\mathcal{C}_1| + |\mathcal{C}_2| = k \tag{4.1}$$
$$|\mathcal{C}_1| + |\mathcal{N}_1| \leq k \tag{4.2}$$

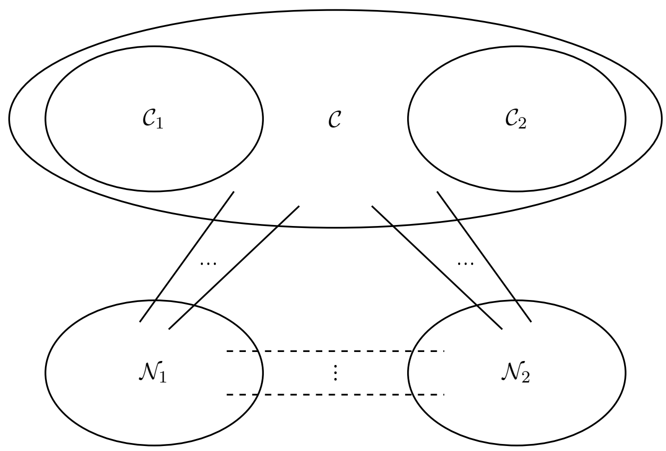


Figure 4.1: Partition of $V(G)$ in the proof of Theorem 2.29

Let m be the number of edges between $\mathcal{C}$ and $V(G) \setminus \mathcal{C} = \mathcal{N}_1 \cup \mathcal{N}_2$. Let us make the following observations:

- Every vertex of $\mathcal{C}_2$ is adjacent to at least one other vertex of $\mathcal{C}_2$. Moreover, it is not possible to have two vertices of $\mathcal{C}_2$ forming a connected component of $G[\mathcal{C}_2]$; indeed, this would imply that they are not separated by $\mathcal{C}$. Thus, there are at least $\frac{2|\mathcal{C}_2|}{3}$ edges within $\mathcal{C}_2$. So, $m \leq k \cdot \Delta - 2 \cdot \frac{2|\mathcal{C}_2|}{3} = k \cdot \Delta - \frac{4|\mathcal{C}_2|}{3}$ since the maximum degree of G is Δ.

- There are no edges from any vertex of $\mathcal{C}_1$ to another vertex of $\mathcal{C}$.

- There are at least $2|\mathcal{N}_2|$ edges between $\mathcal{C}$ and $\mathcal{N}_2$ since every vertex of $\mathcal{N}_2$ has at least two neighbours in the code, and there are exactly $|\mathcal{N}_1|$ edges between $\mathcal{C}$ and $\mathcal{N}_1$.

Summarizing, we have $k \cdot \Delta - \dfrac{4|\mathcal{C}_2|}{3} \geq m \geq |\mathcal{N}_1| + 2|\mathcal{N}_2|$. Using Inequalities (4.1) and (4.2), we get:

$$\begin{aligned}
k\Delta \geq & \frac{4|\mathcal{C}_2|}{3} + |\mathcal{N}_1| + 2|\mathcal{N}_2| = k - |\mathcal{C}_1| + |\mathcal{N}_1| + 2|\mathcal{N}_2| + \frac{|\mathcal{C}_2|}{3} \\
k\Delta \geq & 2(|\mathcal{N}_1| + |\mathcal{N}_2|) + \frac{|\mathcal{C}_2|}{3} = 2(n-k) + \frac{|\mathcal{C}_2|}{3} \\
k(\Delta+2) \geq & 2n + \frac{|\mathcal{C}_2|}{3} \geq 2n \qquad (4.3) \\
k \geq & \frac{2n}{\Delta+2} \qquad ☆
\end{aligned}$$

The lower bound of Theorem 2.29 is tight. Graphs which reach it can easily be constructed with the following construction.

Construction 4.1. *Let Δ be an integer; let H be a Δ-regular graph on k vertices, and let H' be a graph on $\frac{k\Delta}{2}$ vertices and of maximum degree $\Delta - 2$. Construct a graph $\mathcal{G}(H, H')$ with $\frac{k(\Delta+2)}{2}$ vertices as follows:*

1. *Subdivide each edge of H once.*
2. *Embed H' on the set of newly created vertices.*

It is clear that the graphs $\mathcal{G}(H, H')$ obtained from Construction 4.1 are identifiable and have maximum degree Δ, and that the set of original vertices of H forms an identifying code of size $\frac{2|V(\mathcal{G}(H,H'))|}{\Delta+2}$ of $\mathcal{G}(H, H')$.

For two examples, see Figure 4.2 which depicts $\mathcal{G}(K_5, C_{10})$, and Figure 4.3 which depicts $\mathcal{G}(P_{10}, \overline{K_{15}})$, where P_{10} is the Petersen graph.

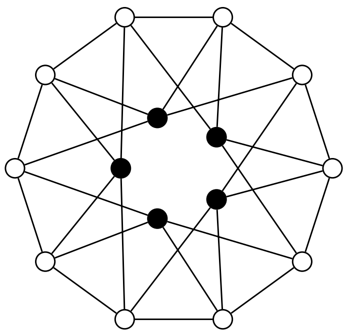


Figure 4.2: The graph $\mathcal{G}(K_5, C_{10})$

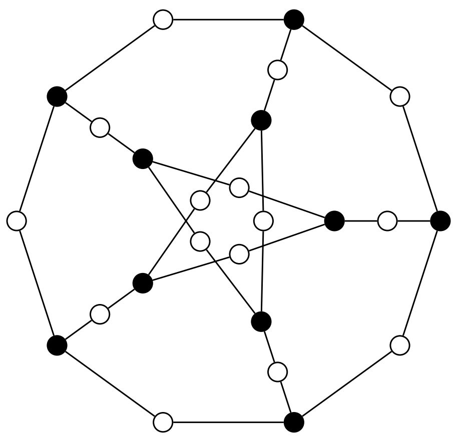


Figure 4.3: The graph $\mathcal{G}(P_{10}, \overline{K_{15}})$

We note the graphs of Construction 4.1 have no false twins. If H' is $(\Delta-2)$-regular, $\mathcal{G}(H, H')$ is regular. If H' has no edge, $\mathcal{G}(H, H')$ is bipartite. Moreover, for any $\Delta \geq 3$ and for arbitrarily large values of g and n, there exists a Δ-regular graphs G on n vertices having girth g [79]; we obtain the following proposition:

Corollary 4.2. *For any $\Delta \geq 3$ and arbitrarily large values of g and n, there are infinitely many connected bipartite graphs G on n vertices having maximum degree Δ, girth g (and hence, no forced vertex) with $\gamma^{ID}(G) = \frac{2n}{\Delta+2}$. Moreover there are infinitely many connected Δ-regular graphs G on n with $\gamma^{ID}(G) = \frac{2n}{\Delta+2}$.*

We remark however, that the bound is not tight for planar graphs. Indeed, first of all, the bound $\gamma^{\mathrm{ID}}(G) \geq \frac{|V(G)|+10}{7}$ holds for any identifiable planar graph G by Theorem 2.34 from [183]; it is stronger whenever $\Delta(G) \geq 12$. Furthermore, it is well-known that any planar graph is 5-degenerate, hence there are no Δ-regular planar graphs for any $\Delta \geq 6$ and Construction 4.1 cannot be applied in that case.

In fact, we can show that each graph reaching the bound of Theorem 2.29 can be constructed using Construction 4.1; this gives a full characterization of these graphs:

Theorem 4.3. *Let G be a graph on n vertices with maximum degree Δ and $\gamma^{ID}(G) = \frac{2n}{\Delta+2}$. Then, there exist two graphs H, H' such that $G = \mathcal{G}(H, H')$ as defined in Construction 4.1.*

Proof. Consider the partition of $V(G)$ illustrated in Figure 4.1, and let us recall Inequality (4.3) from the proof of Theorem 2.29:

$$k(\Delta+2) \geq 2n + \frac{|\mathcal{C}_2|}{3}.$$

Since we assume $k = \gamma^{\mathrm{ID}}(G) = \frac{2n}{\Delta+2}$, we obtain:

$$2n \geq 2n + \frac{|\mathcal{C}_2|}{3}.$$

That is, $\mathcal{C}_2 = \emptyset$. This implies that $|\mathcal{C}_1| = k$, and thus $\mathcal{N}_1 = \emptyset$ since $|\mathcal{C}_1| + |\mathcal{N}_1| = k$. Hence, a graph reaching the lower bound has an independent set as its code, and every non-code vertex is adjacent to exactly two code vertices, whereas all code vertices have Δ neighbours. It is now clear that G can be obtained using Construction 4.1. Graph H has vertex set $V(H) = \mathcal{C}$ and is obtained by first removing all edges between non-code-vertices and then, contracting one of the two edges $\{u, c\}$, $\{u, c'\}$ for each non-code vertex u (where c, c' are the two code neighbours of u in G). Graph H' is the graph induced by the set $V(G) \setminus \mathcal{C}$. ☆

4.2 Upper bounds depending on the order and the maximum degree - a conjecture and some constructions

In this section, we present a conjecture and some constructions of families of graphs which reach this conjectured bound. We then study the structure and maximum number of false twins and forced vertices in a graph.

The conjecture was published in [FKKR12]; some of the constructions were mentioned in the author's master thesis [Fo09], and were published in [FP12] together with some new ones. The studies of false twins and forced vertices are from [FKKR12, FP12].

4.2.1 A conjecture

We saw in Chapter 2.5.1 and in the previous section that there is a lower bound on $\gamma^{\mathrm{ID}}(G)$ depending on the order and the maximum degree of G, showing that the maximum degree has a strong influence on this parameter. When considering upper bounds in terms of n and Δ, we conjecture that the following bound holds.

Conjecture 4.4. *There exists a constant c such that for any nontrivial connected identifiable graph G of maximum degree Δ, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta} + c$.*

It is known that there exist examples of specific families of graphs such that $\gamma^{\mathrm{ID}}(G) = n - \frac{n}{\Delta}$ (e.g. the complete bipartite graph $K_{\Delta,\Delta}$, Sierpiński graphs [95] and other classes of graphs that will be described in Chapter 4.2.2. Other classes of graphs with slightly smaller values of parameter γ^{ID} are known, including graphs having high girth. For instance, it is shown in [24] that $\gamma^{\mathrm{ID}}(T^h_{\Delta-1}) = \lceil n - \frac{n}{\Delta-1+1/\Delta} \rceil$ for the complete $(\Delta-1)$-ary tree $T^h_{\Delta-1}$ of height h having n vertices (see Theorem 2.33).

As we saw in Chapter 2.5.1, for all identifiable graphs on n vertices having at least one edge, the upper bound $\gamma^{\mathrm{ID}}(G) \leq n-1$ holds (see Theorem 2.27). This bound is tight, as we have seen in Chapter 3.4, where we proposed a full characterization of these graphs. Hence, for graphs of very high maximum degree (say $\Delta = n-1$), the conjecture holds with $c \leq 1$ since $n-1 = n - \frac{n}{\Delta} + \frac{1}{n-1}$. We note that the graphs reaching the bound $\gamma^{\mathrm{ID}}(G) \leq n-1$ actually have maximum degree $n-2$ or $n-1$, as demonstrated in Chapter 3.

Moreover, for any connected graph G of maximum degree 2 (i.e. when G is either a path or a cycle), the exact value of $\gamma^{\mathrm{ID}}(G)$ is known (see Section 2.5.2). In this case, the bound $\gamma^{\mathrm{ID}}(G) \leq \frac{n}{2} + \frac{3}{2} = n - \frac{n}{2} + \frac{3}{2}$ holds and is reached for infinitely many values of n (more precisely, this is the case when G is a cycle of odd order $n \geq 7$). Hence, the conjecture holds for $\Delta = 2$ and $c = \frac{3}{2}$. In fact, we expect the constant c to be small. The previous example which shows $c \geq \frac{3}{2}$ is the worst case we know of.

There is some evidence that proving the conjecture even for the case $\Delta = 3$ might be challenging. Indeed, as mentioned in Chapter 2.4.3, the similar notion of *identifying open codes* (that is, identifying codes on *open* neighbourhoods rather than closed neighbourhoods, i.e. vertices do not dominate or identify themselves) was studied very recently in [112] for cubic graphs. Denoting $\gamma^{\mathrm{OID}}(G)$ the minimum size of an identifying open code of a graph G, the authors are able to prove that in a cubic graph G admitting an identifying open code, $\gamma^{\mathrm{OID}}(G) \leq \frac{3n}{4}$. Moreover, they conjecture that the only (connected) examples reaching the bound belong to a set of six graphs, and that otherwise, $\gamma^{\mathrm{OID}}(G) \leq \frac{3n}{5}$, which, if true, would be sharp. This result is proved by using a strong connection to *distinguishing transversals* of 3-uniform hypergraphs. It is worth noting that using the same technique in the case of (classic) identifying codes in cubic graphs would require to handle distinguishing transversals of 4-uniform hypergraphs, which seems to be a much more difficult task.

4.2.2 Extremal constructions

We now present some constructions which yield arbitrarily large graphs of given maximum degree and having large identifying code number (with respect to Conjecture 4.4).

Construction 4.5. *Given any Δ_H-regular multigraph H (without loops) on n_H vertices, let $\mathcal{G}_1(H)$ be the graph on $n = n_H(\Delta_H + 1)$ and maximum degree $\Delta = \Delta_H + 1$ constructed as follows:*

1. *Replace each vertex v of H by a clique $K(v)$ of $\Delta_H + 1$ vertices*
2. *For each vertex v of H, let $N(v) = \{v_1, \ldots, v_{\Delta_H}\}$ and $K(v) = \{k_0(v), \ldots, k_{\Delta_H}(v)\}$. For each $k_i(v)$ but one ($1 \leq i \leq \Delta_H$), connect it with an edge in $\mathcal{G}_1(H)$, to a unique vertex of $K(v_i)$, denoted $f(k_i(v))$.*

An example of this construction is given in Figure 4.4, where H is the hypercube of dimension 3, H_3, and the black vertices are those which belong to a minimum identifying code of $\mathcal{G}_1(H_3)$.

We have the following proposition:

Proposition 4.6. *Let H be a Δ-regular multigraph without loops having n_H vertices. The graph $G = \mathcal{G}_1(H)$ on $n = n_H(\Delta_H + 1)$ vertices obtained by applying Construction 4.5 on H has $\gamma^{ID}(G) = n_H\Delta = n - \frac{n}{\Delta(G)}$.*

Proof. For each vertex v of H and for each $1 \leq i \leq \Delta_H$, note that $f(k_i(v))$ is $k_0(v)k_i(v)$-forced. Therefore, G has $\Delta_H n_H = n - \frac{n}{\Delta}$ forced vertices. In fact one can check that these forced vertices form an identifying code, which completes the proof. ☆

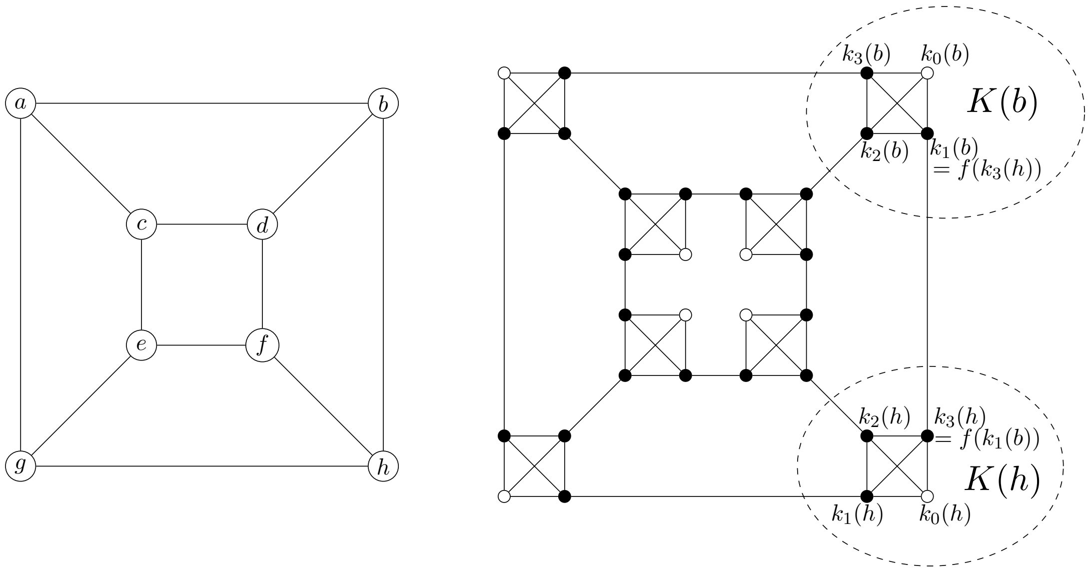


Figure 4.4: The graphs H_3 and $\mathcal{G}_1(H_3)$.

The following construction is very similar, but yields regular graphs. It was first introduced in the author's master thesis [Fo09].

Construction 4.7. *Given any Δ_H-regular multigraph H (without loops) on n_H vertices, let $\mathcal{G}_2(H)$ be the Δ-regular graph on $n_H\Delta_H$ vertices (where $\Delta = \Delta_H$) constructed as follows:*

1. *Replace each vertex v of H by a clique $K(v)$ of Δ_H vertices.*

2. *For each vertex v of H, let $N(v) = \{v_1, \dots, v_{\Delta_H}\}$ and $K(v) = \{k_1(v), \dots, k_{\Delta_H}(v)\}$. For each $k_i(v)$ ($1 \leq i \leq \Delta_H$), connect it with an edge in $\mathcal{G}_2(H)$, to a unique vertex of $K(v_i)$, denoted $f(k_i(v))$.*

An example of this construction is given in Figure 4.5, where H is the complete graph K_5, and the black vertices form a minimum identifying code of $\mathcal{G}_2(K_5)$.

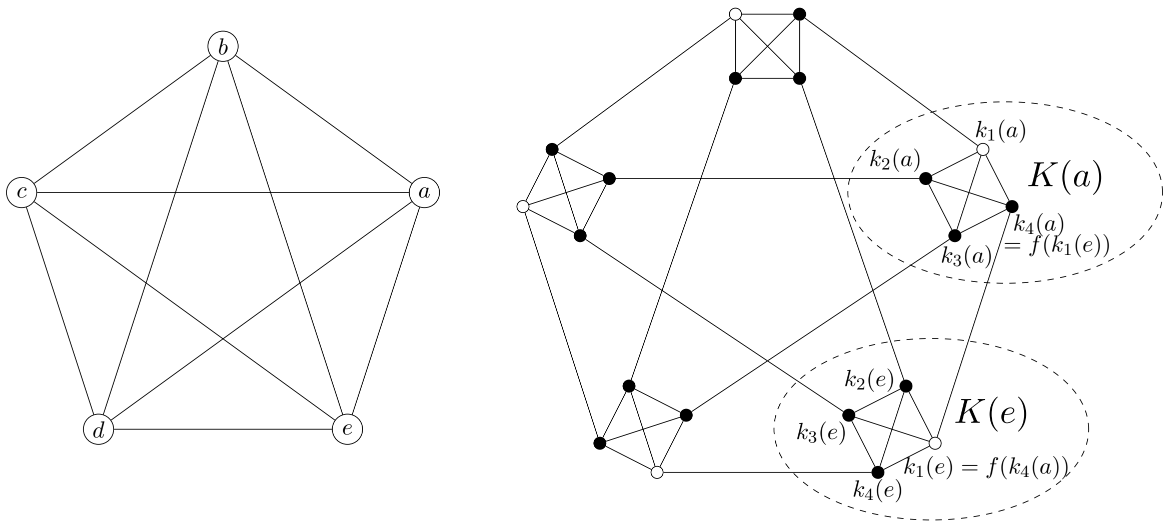


Figure 4.5: The graphs K_5 and $\mathcal{G}_2(K_5)$.

We have the following proposition:

Proposition 4.8. *Let H be a Δ-regular multigraph without loops having n_H vertices. The graph $G = \mathcal{G}_2(H)$ on $n = n_H\Delta_H$ vertices obtained by applying Construction 4.7 on H has $\gamma^{ID}(G) = n_H(\Delta - 1) = n - \frac{n}{\Delta(G)}$.*

Proof. For each vertex v of H, we let $A(v)$ be the set $\{f(k_i(v)) \mid 1 \le i \le \Delta\}$.

For each vertex v of H, in order to separate each pair of vertices $k_i(v), k_j(v)$ $(1 \le i < j \le \Delta)$ of $K(v)$ in G, either $f(k_i(v))$ or $f(k_j(v))$ must belong to any identifying code. Repeating this argument for each pair in $K(v)$ shows that at least $\Delta - 1$ vertices from $A(v)$ are needed in the code. Since for any two cliques $K(u)$ and $K(v)$, the sets $A(u)$ and $A(v)$ are disjoint, at least $n_H(\Delta - 1)$ vertices are needed in an identifying code of G.

For the other direction, we build an identifying code of this size by choosing one vertex of each set $A(x)$, in such a way that for each vertex u originally in H, exactly one vertex of $K(u)$ is chosen. Then the set of non-chosen vertices will be an identifying code of G. To select this set of vertices, one can consider the bipartite incidence multigraph B of H: the vertex set of B is $V \cup V'$ (where V and V' are copies of $V(H)$) and there is an edge xx' in B if $x \in V$, $x' \in V'$ and $xx' \in E(H)$. The multigraph B is k-regular and bipartite, thus by P. Hall's marriage theorem (Theorem 2.1) it has a perfect matching M. For each vertex $x \in V$, let $\rho(x)$ be the vertex in V' such that $x\rho(x) \in M$. Let now v'^x_M be the vertex of G that belongs to both sets $A(x)$ and $K(\rho(x))$ (in G). We let $\mathcal{C} = V(G) \setminus \{v'^x_M\}_{x \in H}$. Exactly one element of each $A(x)$ is not in $\mathcal{C}$, and for each vertex x, exactly one vertex of $K(x)$ is not in $\mathcal{C}$. This implies that $\mathcal{C}$ is an identifying code and completes the proof. ☆

Proposition 4.8 shows that despite the fact that $\mathcal{G}_2(H)$ has no forced vertices, $\gamma^{\text{ID}}(\mathcal{G}_2(H)) = n - \frac{n}{\Delta}$.

We remark that Constructions 4.5 and 4.7 are close to Sierpiński graphs, which were defined in [134]. Recently, it has been shown that Sierpiński graphs are also extremal with respect to Conjecture 4.4 [95], i.e. for any Sierpiński graph G on n vertices with maximum degree Δ, $\gamma^{\text{ID}}(G) = n - \frac{n}{\Delta}$.

We make the following observation about Constructions 4.5 and 4.7:

Observation 4.9. *Given a Δ_H-regular multigraph H (without loops) on n_H vertices, the graphs $\mathcal{G}_1(H)$ and $\mathcal{G}_2(H)$ constructed using Constructions 4.5 and 4.7, respectively, are line graphs.*

Proof. It is easily checked that $\mathcal{G}_1(H)$ is the line graph of the graph H' obtained from H after subdividing each of its edges once. Similarly, the graph $\mathcal{G}_2(H)$ is the line graph of the graph obtained from H' by adding a degree 1-neighbour to each vertex of H' which also was a vertex in H. ☆

The following construction was introduced in the author's master thesis [Fo09] and yields regular bipartite graphs.

Construction 4.10. *Given an even number $2k$ and an integer $\Delta \ge 3$, we construct an identifiable Δ-regular bipartite graph $\mathcal{G}_3(2k, d)$ on $n = 2k\Delta$ vertices as follows.*

1. *Let $\{c_0, \dots, c_{2k-1}\}$ be a set of $2k$ vertices and add the edges of the perfect matching $\{c_i c_{(i+1) \bmod 2k} \mid i \text{ is odd}\}$.*

2. *For each even i $(0 \le i \le 2k-2)$, build a copy $K(i)$ of the complete bipartite graph $K_{\Delta-1,\Delta-1}$. Join vertex c_i to all vertices of one part of the bipartition of $K(i)$, and join vertex c_{i+1} to all other vertices of $K(i)$.*

An example of this construction is given in Figure 4.6, where $2k = 8$, $\Delta = 3$, and the black vertices form a minimum identifying code of $\mathcal{G}_3(8, 3)$.

Proposition 4.11. *Let $G = \mathcal{G}_3(2k, \Delta)$ be the graph on $n = 2k\Delta$ vertices and maximum degree Δ obtained by applying Construction 4.10. We have $\gamma^{ID}(G) = n - \frac{n}{2\Delta/3}$.*

Proof. Consider an identifying code of G. Note that in each copy $K(i)$ of $K_{\Delta-1,\Delta-1}$, at least $2\Delta - 4$ vertices have to belong to the code in order to separate the vertices being in the same part of the bipartition of $K(i)$. Now, if exactly $2\Delta - 4$ vertices of $K(i)$ belong to the code, in order to separate the two remaining vertices, either c_i or c_{i+1} belongs to the code. Hence for each odd i, at most three vertices from $\{c_i, c_{i+1}\} \cup V(K(i))$ do not belong to a code of G. On

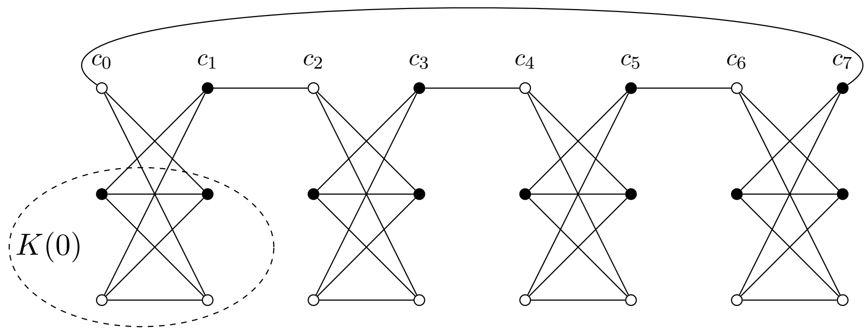


Figure 4.6: The graph $\mathcal{G}_3(8,3)$.

the other hand, taking all vertices c_i such that i is even together with $\Delta - 2$ vertices of each part of the bipartition of each copy of $K_{\Delta-1,\Delta-1}$ yields an identifying code of this size. Hence $\gamma^{\text{ID}}(\mathcal{G}_3(2k,\Delta)) = k + 2k(\Delta-2) = n - \frac{n}{2\Delta/3}$. ☆

4.2.3 On the number and structure of false twins and forced vertices in a graph

The aim of what follows is to study the structure of false twins and forced vertices in a graph. This study is related to Conjecture 4.4 since a graph having many false twins or forced vertices will have a large identifying code number and could provide a counterexample to this conjecture.

4.2.3.1 False twins

The results of this subsection are useful when dealing with identifying codes. Indeed, as pointed out earlier, for any set of mutually false twins in G, all but one of them belong to any identifying code of G. The next proposition gives an upper bound on the number of false twins in a graph.

Proposition 4.12. *Let G be a graph on n vertices having maximum degree Δ and no isolated vertices, then G has at most $\frac{n(\Delta-1)}{2}$ pairs of false twins.*

Proof. Let us build a graph H on $V(G)$, where two vertices u, v are adjacent in H if they are false twins in G. Note that since a vertex can have at most $\Delta - 1$ false twins, H has maximum degree $\Delta - 1$. Therefore it has at most $\frac{n(\Delta-1)}{2}$ edges and the claim follows. ☆

Note that the bound of Proposition 4.12 is tight since in a complete bipartite graph $K_{\Delta,\Delta}$, $n = 2\Delta$ and there are exactly $2\binom{\Delta}{2} = \frac{n(\Delta-1)}{2}$ pairs of false twins.

The next proposition (which appeared in [FKKR12]) shows how to build an identifying code of a graph G with relatively small size when G contains a large number of false twins. We let $\equiv$ denote the *false twin relation* over $V(G)$, where $u \equiv v$ if u, v are false twins. This relation is an equivalence relation. We call an equivalence class of $\equiv$ *nontrivial* if it has at least two elements.

Proposition 4.13. *Let G be a nontrivial connected identifiable triangle-free graph on n vertices and maximum degree Δ non isomorphic to C_4. Let $\mathcal{F} = \{F_1, \ldots, F_{|\mathcal{F}|}\}$ be the set of all nontrivial equivalence classes over $\equiv$ in G. Then G has an identifying code of size at most $n - |\mathcal{F}|$.*

Proof. First, we may suppose that G is not isomorphic to P_3 since in that case the lemma holds: P_3 has its minimum identifying code of size 2 and $|\mathcal{F}| = 1$.

For each $F_i \in \mathcal{F}$, $1 \leq i \leq |\mathcal{F}|$, let x_i be an arbitrary vertex of F_i, and let $X = \cup_{i=1}^{|\mathcal{F}|} x_i$. We claim that if G is not isomorphic to P_3 or C_4, $\mathcal{C} = V(G) \setminus X$ is an identifying code of G. First, observe that $\mathcal{C}$ is a dominating set of G. Now, consider two vertices x, y. We need to show that they are separated from each other.

If x, y are false twins, the one belonging to the code separates them. Otherwise, since G is identifiable, there is a vertex z which is able to separate them, say z belongs to $N[x]$, but not to $N[y]$. If z belongs to the code, we are done. Otherwise, $z \in X$.

If z is a neighbour of x, consider a false twin z' of z. If $z' \neq y$, z' belongs to the code and separates x, y, so we are done. Otherwise, since G is not isomorphic to P_3 and z, y are false twins, one of x or y has another neighbour, say t. If t belongs to the code we are done. Otherwise, if t is a neighbour of y, since G is not isomorphic to C_4, either x or y has another neighbour. We can repeat the argument but this time, either this neighbour or its false twin separates x, y. If t is a neighbour of x, t cannot be a false twin of y and therefore either t or its false twin separates x, y.

Finally, if $z = x$, x and y are not adjacent. But since they are not false twins, there is another vertex, say u, with $u \notin \{x, y\}$, such that u is adjacent to exactly one of x, y. Now, either u belongs to the code and we are done, or a false twin of u (which also is adjacent to exactly one of x, y), which completes the proof. ☆

4.2.3.2 Forced vertices

In this subsection, we give lower bounds on the ratio of *non-forced* vertices of G, defined as follows:

Definition 4.14. *Given a graph G on n vertices, we denote by $NF(G)$ the* proportion of non-forced vertices *of G, i.e. the ratio $\frac{x}{n}$, where x is the number of vertices of G that are not forced.*

We now define A_∞^+ as the infinite graph with vertex set $\mathbb{N} \times \{0, 1\}$ and with edge set $\{\{(i, b), (j, b)\} \mid i \neq j, b \in \{0, 1\}\} \cup \{\{(i, 0), (j, 1)\} \mid i < j\}$. Note that A_∞^+ is an induced subgraph of the graph A_∞ (which we studied in Chapter 3.5, Definition 3.29). The following lemma, which is a strengthening of Theorem 2.27, is from N. Bertrand's Master thesis [22]. We give an independent proof in Appendix A.1 (which appeared in [FGK+11]) as [22] is not accessible.

Lemma 4.15 ([22])**.** *If G is an identifiable graph (infinite or not) not containing A_∞^+ as an induced subgraph, then for every vertex x of G, there is a vertex $y \in N[x]$ such that $G - y$ is identifiable.*

The following proposition is a direct consequence of Lemma 4.15.

Proposition 4.16. *Let G be a graph on n vertices and of maximum degree Δ. Then $NF(G) \geq \frac{1}{\Delta+1}$.*

Proof. Observe that a vertex v of G is not forced only if $V(G) \setminus \{v\}$ is an identifying code of G. Hence, by Lemma 4.15, the set S of non-forced vertices is a dominating set of G, and thus $|S| \geq \frac{n}{\Delta+1}$. ☆

Note that Proposition 4.16 is tight. Indeed, consider the graph A_k on $2k$ vertices (introdueced with more details in Chapter 3.4) defined as follows: $V(A_k) = \{x_1, \ldots, x_{2k}\}$ and $E(A_k) = \{x_i x_j, |i - j| \leq k - 1\}$. A_k can be seen as the $(k-1)$-th power of the path P_{2k}. In the graph A_k with an additional universal vertex x (i.e. x is adjacent to all vertices of A_k), one can check that all vertices but x are forced. This graph has $n = 2k + 1$ vertices, maximum degree $2k$ and exactly $1 = \frac{n}{\Delta+1}$ non-forced vertex. Taking all forced vertices gives a minimum identifying code of this graph.

Observe that graph A_k contains two cliques of k vertices. In fact, we can improve the bound of Proposition 4.16 for graphs having no large cliques. Let us first introduce an auxiliary structure that will be needed in order to prove this result.

Let G be an identifiable graph. We define a partial order $\preceq$ over the set of vertices of G such that $u \preceq v$ if $N[u] \subseteq N[v]$. We construct an oriented graph $\mathcal{H}(G)$ on $V(G)$ as a subgraph of the Hasse diagram of poset $(V(G), \preceq)$.[1] The arc set of $\mathcal{H}(G)$ is the set of all arcs $\overrightarrow{uv}$ where there exists some vertex x such that $N[v] = N[u] \cup \{x\}$. Then x is uv-forced, and we note $x = f(\overrightarrow{uv})$. For a vertex v of $V(G)$, we define the set $F(v)$ as the union of v itself and the set of all predecessors and successors of v in $\mathcal{H}(G)$. Observe that $\mathcal{H}(G)$ has no directed cycle since it represents a partial order, and thus predecessors and successors are well-defined.

[1] The *Hasse diagram* of a poset $(V(G), \preceq)$ is an oriented graph on vertex set $V(G)$ in which there is an arc xy whenever $x \preceq y$ and there is no z such that $x \preceq z \preceq y$.

Lemma 4.17. *Let G be a graph having no k-clique. Then for each vertex u, $|F(u)| \leq \beta(k)$, where $\beta(k)$ is a function depending only on k.*

Proof. First of all, we prove that the maximum in-degree of $\mathcal{H}(G)$ is at most $2k-3$, and its out-degree is at most $k-2$.

Let u be a vertex of G. Suppose u has $2k-2$ in-neighbours in $\mathcal{H}(G)$. Since for each in-neighbour v of u, $|N[u]\Delta N[v]| = 1$ in G, each of them is non-adjacent in G to at most one of the other in-neighbours (in the worst case the in-neighbours of u induce in G a clique of $2k-2$ vertices minus the edges of a perfect matching). Hence they induce a clique of size at least $k-1$ in G. Together with vertex u, they form a k-clique in G, a contradiction.

Now suppose u has $k-1$ out-neighbours in $\mathcal{H}(G)$. Since for each out-neighbour v of u in $\mathcal{H}(G)$, $N[u] \subseteq N[v]$ in G, u and its out-neighbours form a k-clique in G, a contradiction.

Now, consider the subgraph of $\mathcal{H}(G)$ induced by $F(u)$. We claim that the longest directed chain in this subgraph has at most $k-1$ vertices. Indeed, all the vertices of such a chain are pairwise adjacent in G. Since G is assumed not to have any k-cliques, there are at most $k-1$ vertices in a directed chain.

Finally, we obtain that $F(u)$ has size at most $\beta(k) = \sum_{i=0}^{k-2}(2k-3)^i$ and the claim of the lemma follows. ☆

We now need to prove a few additional claims regarding the structure of $\mathcal{H}(G)$. In the following claims, we suppose that G is an identifiable graph.

Claim 4.18. *Let s be a forced vertex in G with $s = f(\overrightarrow{uv})$ for some vertices u and v. If t is an in-neighbour of s in $\mathcal{H}(G)$, then $v = f(\overrightarrow{ts})$. Moreover if v is forced with $v = f(\overrightarrow{xy})$, then necessarily $y = s$.*

Proof. For the first implication, suppose s has an in-neighbour t in $\mathcal{H}(G)$. An illustration is provided in Figure 4.7. Since $u \not\sim s$, then $u \not\sim t$. Moreover $v \not\sim t$ since $s = f(\overrightarrow{uv})$. Since $s \sim v$ the claim follows. For the other implication, suppose there exist two vertices x, y such that $v = f(\overrightarrow{xy})$. Hence $y \sim v$ but $x \not\sim v$. Therefore $u \not\sim x$ (otherwise v would be adjacent to x too) and hence $u \not\sim y$. Now the only vertex adjacent to v but not to u is s, so $y = s$. ☆

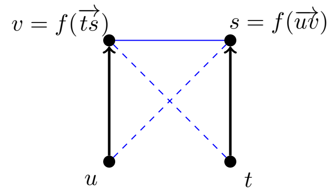


Figure 4.7: The situation of Claim 4.18. Arcs belong to $\mathcal{H}(G)$. Full thin edges belong to G only, dashed edges are non-edges in G.

Claim 4.19. *Let s be a forced vertex in G with $s = f(\overrightarrow{uv})$ for some vertices u and v. Then s has at most one in-neighbour in $\mathcal{H}(G)$.*

Proof. Suppose s has two distinct in-neighbours t and t' in $\mathcal{H}(G)$ (see Figure 4.8 for an illustration). By Claim 4.18, v is both ts-forced and $t's$-forced. But then $N[t] = N[s] \setminus \{v\} = N[t']$. Then t and t' are twins, a contradiction since G is identifiable. ☆

Claim 4.20. *Let s be a forced vertex in G with $s = f(\overrightarrow{uv})$, and let t be a forced in-neighbour of s in $\mathcal{H}(G)$ with $t = f(\overrightarrow{xy})$ for some vertices u, v, x, y. Then $x = v$.*

Proof. Since $t \sim y$, then $s \sim y$ too. But since $t = f(\overrightarrow{xy})$, $x \sim s$ and $x \not\sim t$. Now by Claim 4.18, $v = f(\overrightarrow{ts})$, that is, v is the unique vertex such that v is adjacent to s, but not to t. Therefore $x = v$. ☆

We now obtain the following lemma using the previous claims.

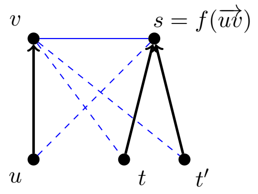


Figure 4.8: The situation of Claim 4.19. Arcs belong to $\mathcal{H}(G)$. Full thin edges belong to G only, dashed edges are non-edges in G.

Lemma 4.21. *Let s be a non-isolated sink in $\mathcal{H}(G)$ which is forced in G with $s = f(\overrightarrow{uv})$ for some vertices u and v. Then either s has a non-forced predecessor t in $\mathcal{H}(G)$ such that $F(s) \subseteq F(t)$, or there exists a non-forced vertex $w(s)$ such that $F(s) \subseteq N_G[w(s)]$. Moreover, if there are ℓ additional sinks $\{s_1, \dots, s_\ell\}$ which are all non-isolated in $\mathcal{H}(G)$ and such that $w(s) = w(s_1) = \dots = w(s_\ell)$, then there exists a set of $\ell + 1$ distinct vertices inducing a clique together with $w(s)$.*

Proof. First of all, recall that $\mathcal{H}(G)$ has no directed circuits. Suppose s has a non-forced predecessor in $\mathcal{H}(G)$ and let t be one such predecessor having the shortest distance to s in $\mathcal{H}(G)$. By Claim 4.19, predecessors of s are either successors or predecessors of t, and there is a directed path from t to s in $\mathcal{H}(G)$. Hence $F(s) \subseteq F(t)$, which proves the first part of the statement.

Now suppose all predecessors of $s = f(\overrightarrow{uv})$ are forced. By Claim 4.19, s and its predecessors form a directed path $\{t_0, \dots, t_m, s\}$ in $\mathcal{H}(G)$ (for an illustration, see Figure 4.9(a)). Note that by Claim 4.18, we have $v = f(\overrightarrow{t_m s})$. By our assumption we know that t_m is forced, say $t_m = f(\overrightarrow{xv_m})$ for some vertices x and v_m. But now by Claim 4.20, $x = v$ and $t_m = f(\overrightarrow{vv_m})$. Now, repeating these arguments for each other predecessor of s shows that there is a directed path $\{u, v, v_m, \dots, v_0\}$ with $t_m = f(\overrightarrow{vv_m})$ and for all i, $0 \le i \le m-1$, $t_i = f(\overrightarrow{v_{i+1}v_i})$. In particular, $t_0 = f(\overrightarrow{v_1 v_0})$. Observe also that for all $i \ge 1$, $v_i = f(\overrightarrow{t_{i-1}t_i})$. By applying Claim 4.20 on vertices v_1, v_0 and t_0, if v_0 is forced then t_0 has an in-neighbour in $\mathcal{H}(G)$, a contradiction — hence v_0 is non-forced. Moreover note that since $v_0 \sim t_0$, then v_0 is adjacent to all successors of t_0 in $\mathcal{H}(G)$, that is, to all elements of $F(s)$. Therefore, putting $w(s) = v_0$, we obtain the second part of the statement.

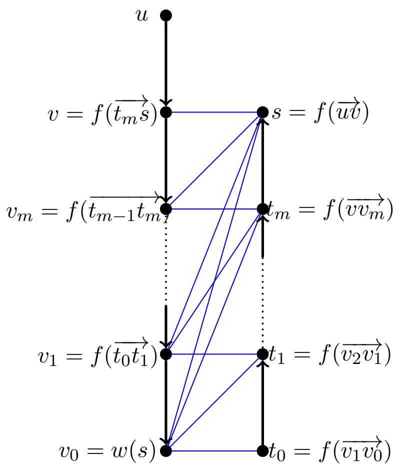
For the last part, suppose there exists a set of ℓ additional forced sinks $\{s_1, \dots, s_\ell\}$ which are non-isolated in $\mathcal{H}(G)$ and such that all their predecessors in $\mathcal{H}(G)$ are forced with $w(s_i) = v_0$ for $1 \le i \le \ell$ (for an illustration, see Figure 4.9(b)). For each such sink s_i, by the previous paragraph, the vertices of $F(s_i)$ induce a directed path $\{t_0^i, \dots, t_{m_i}^i, s_i\}$ in $\mathcal{H}(G)$. Moreover we know that there is a vertex x_i such that t_0^i is $x_i v_0$-forced. We claim that the set of vertices $X = \{x_1, \dots, x_\ell\}$ together with v_0 and v_1, form a clique in G of $\ell + 2$ vertices.

We first claim that for all i, j in $\{1, \dots, \ell\}$, $x_i \neq t_0^j$. If $i = j$, this is clear by our assumptions. Otherwise, suppose by contradiction, that $x_i = t_0^j$ for some $i \neq j$ in $\{1, \dots, \ell\}$. Then we claim that $x_j = t_0^i$. Indeed, by the previous part of the proof, we know that $f(\overrightarrow{x_j v_0}) = t_0^j = x_i$ — hence $x_j \not\sim x_i$. But since $\overrightarrow{x_i v_0}$ is an arc in $\mathcal{H}(G)$, we must have $f(\overrightarrow{x_i v_0}) = x_j$. Again, we know that $f(\overrightarrow{x_i v_0}) = t_0^i$, hence $x_j = t_0^i$. Let t_1^i denote the successor of t_0^i in the directed path from t_0^i to s_i in $\mathcal{H}(G)$. We know from the previous part of the proof that $f(\overrightarrow{t_0^i t_1^i}) = x_i = t_0^j$. However since $t_0^i = x_j$ we also know that $f(\overrightarrow{t_0^i v_0}) = x_i$. This implies that $N_G[v_0] = N_G[t_1^i]$, a contradiction since these two vertices are distinct and G is identifiable.

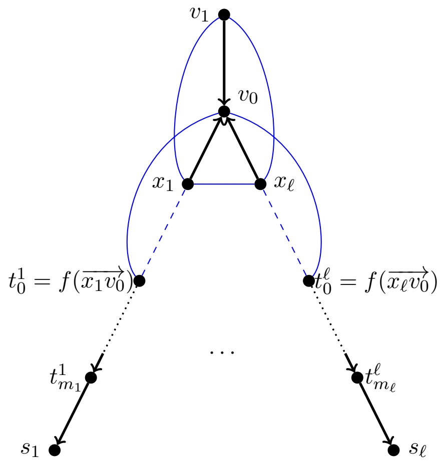
Now, observe that the vertices of X must all be pairwise adjacent. All vertices of X are adjacent to v_0, and for each x_i, $N[v_0] = N[x_i] \cup \{t_0^i\}$, hence x_i is adjacent to all neighbours of v_0 except t_0^i. But by the previous paragraph, we know that $t_0^i \neq x_j$ for all $j \in \{1, \dots, \ell\}$, hence x_i is adjacent to all $x_j \neq x_i$, $j \in \{1, \dots, \ell\}$. For the same reason, each x_i is adjacent to v_1. Hence, the vertices of X form a clique together with v_0 and v_1.

Finally, let us show that all the vertices of X are distinct: by contradiction, suppose that $x_i = x_j$ for some $i \neq j$, $1 \le i, j \le \ell$. Since t_0^i is $x_i v_0$-forced and t_0^j is $x_j v_0$-forced, we have $t_0^i = t_0^j$. Since s_i and s_j are distinct, this means that s_i and s_j have one predecessor in common. Hence their common predecessor which is nearest to s_i and s_j, say t, has two out-neighbours.

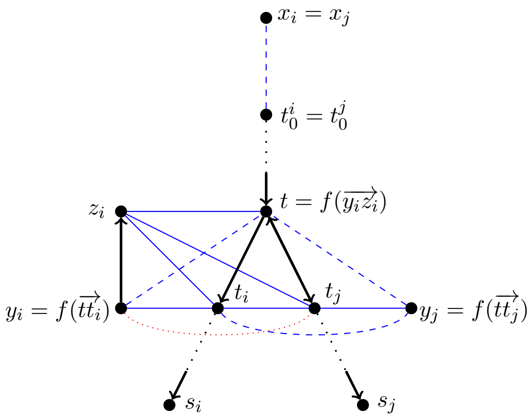
Let t_i (respectively t_j) be the out-neighbour of t which is a predecessor of s_i (respectively s_j) — see Figure 4.9(c) for an illustration. We know that there are two vertices y_i, y_j such that $y_i = f(\overrightarrow{tt_i})$ and $y_j = f(\overrightarrow{tt_j})$. First note that y_i and y_j are distinct: otherwise, we would have $N[t_i] = N[t] \cup \{y_i\} = N[t] \cup \{y_j\} = N[t_j]$ and then t_i, t_j would be twins in G. Observe that since $t \not\sim y_i$ and $y_i \neq f(\overrightarrow{tt_j})$, we have $t_j \not\sim y_i$. We know that t is forced, in fact by the first part of this proof, we also know that $t = f(\overrightarrow{y_iz_i})$ for some vertex z_i. Hence $z_i \sim t$, and since $N[t] \subseteq N[t_j]$, $z_i \sim t_j$. But since $t_j \neq f(\overrightarrow{y_iz_i})$, $t_j \sim y_i$, a contradiction. Hence x_i and x_j are distinct, which completes the proof. ☆



(a) Vertex s and all its predecessors in $\mathcal{H}(G)$ are forced



(b) Vertices $v_0, v_1, x_1, \dots, x_\ell$ induce a clique in G



(c) If $x_i = x_j$, the dotted edge y_it_j is both an edge and a non-edge of G

Figure 4.9: Three situations in the proof of Lemma 4.21. Arcs belong to $\mathcal{H}(G)$. Full thin edges belong to G only, dashed edges are non-edges in G.

We are now ready to prove a bound on $NF(G)$:

Proposition 4.22. *Let G be a graph having no k-clique. Then there exists a constant $\gamma(k)$ depending only on k, such that $NF(G) \geq \frac{1}{\gamma(k)}$.*

Proof. To prove the result, we use $\mathcal{H}(G)$ to construct a set $X = \{x_1, \dots, x_\ell\}$ of non-forced vertices such that $\bigcup_{i=1}^{\ell} A(x_i) = V(G)$, where $A(x_i)$ is a set of at most $\gamma(k)$ vertices. Then we have $\ell \geq \frac{n}{\gamma(k)}$ vertices in X and the claim of the proposition follows.

We now describe a procedure to build set X while considering each non-isolated sink of $\mathcal{H}(G)$. We denote by s the currently considered sink.

Case 1: Sink s is non-forced. Then we set $A(s)$ to be $F(s)$ together with all the vertices which are forced by a pair u, v of vertices of $F(s)$. Note that by Lemma 4.17, $|F(s)| \leq \beta(k)$, where $\beta(k)$ only depends on k. Hence, $|A(s)| \leq \beta(k) + \binom{\beta(k)}{2}$.

Case 2: Sink s is forced. By Lemma 4.21, either s has a non-forced predecessor t such that $F(s) \subseteq F(t)$, or there exists a non-forced vertex $w(s)$ such that $F(s) \subseteq N_G[w]$.

In the first case, we choose t as our non-forced vertex, and we set $A(t)$ to be $F(t)$ together with all the vertices which are forced by a pair u, v of vertices of $F(t)$. Again we have $|A(t)| \leq \beta(k) + \binom{\beta(k)}{2}$.

In the second case, we choose $w = w(s)$ as our non-forced vertex. Now, let $S = \{s, s_1, \ldots, s_\ell\}$ be the set of forced sinks having no non-forced predecessor and such that $w(s) = w(s_1) = \ldots w(s_\ell)$. By Lemma 4.21 we know that there are $\ell+1$ distinct vertices inducing a clique together with w, hence $\ell+2 < k$. We set $A(w)$ to be $F(w) \cup F(s) \cup F(s_1) \cup \ldots \cup F(s_\ell)$ together with all the vertices which are forced by a pair u, v of vertices of this set. We have $|A(w)| \leq k\beta(k) + \binom{k\beta(k)}{2}$.

We have now covered all the vertices which are not isolated in $\mathcal{H}(G)$, since for each non-isolated sink s of $\mathcal{H}(G)$, $F(s)$ is a subset of $A(x)$ for some $x \in X$. Moreover all isolated vertices of $\mathcal{H}(G)$ which are forced, have also been put into some set $A(x)$. Hence only non-forced isolated vertices of $\mathcal{H}(G)$ need to be covered. For each such vertex v, we add v to X and set $A(v) = \{v\}$.

Finally, all vertices belong to some set $A(x)$, $x \in X$, and the size of each set $A(x)$ is at most $\gamma(k) = k\beta(k) + \binom{k\beta(k)}{2}$, which completes the proof. ☆

4.3 Using complements of independent sets to approach Conjecture 4.4

In this section, we introduce the idea of building a (large) independent set S in an identifiable graph G; when set S fulfills a number of special conditions, we are able to show that the complement of S in G, $V(G) \setminus S$, is an identifying code. Depending on the size of S, we get an upper bound on the identifying code number of G.

4.3.1 First bounds

We now provide the following lemma, which will be used to give a first upper bound on $\gamma^{\text{ID}}(G)$ depending on the order and the maximum degree of G. This bound will be improved later using other techniques.

Lemma 4.23. *Let G be an identifiable graph, and I be a distance-4-independent set containing no forced vertex and no isolated vertex. Then $V(G) \setminus I$ is an identifying code of G.*

Proof. Clearly $\mathcal{C} = V(G) \setminus I$ is a dominating set of G. Let x, y be a pair of vertices of G. If they both belong to I, $\mathcal{C} \cap N[x] \neq \mathcal{C} \cap N[y]$ because of the distance between x and y. Otherwise, one of them, say x, is in $\mathcal{C}$. If they are not separated by $\mathcal{C}$, then they must be adjacent. Thus, together they could have only one neighbour in I, call it u. This is a contradiction because u is not forced. ☆

We note that the number 4 is the best possible in the previous lemma (when using a distance-4-independent set). For example, let $G \cong P_4$ and assume x and y are the two ends of G. It is easy to check that $V(G) \setminus \{x\}$ and $V(G) \setminus \{y\}$ are both identifying codes of G but $V(G) \setminus \{x, y\}$ is not.

Theorem 4.24. *Let G be an identifiable graph on n vertices without isolated vertices and with maximum degree Δ. Then $\gamma^{ID}(G) \leq n(1 - \frac{\Delta-2}{\Delta(\Delta-1)^5-2}) = n - \frac{n}{\Theta(\Delta^5)}$. If G has no forced vertex, $\gamma^{ID}(G) \leq n(1 - \frac{1}{1+\Delta-\Delta^2+\Delta^3}) = n - \frac{n}{\Theta(\Delta^3)}$.*

Proof. First of all, we note that if I is a maximal distance-6-independent set, then $|I| \geq \frac{n(\Delta-2)}{\Delta(\Delta-1)^5-2}$. This is true because $|N_5[x]| \leq \frac{\Delta(\Delta-1)^5-2}{\Delta-2}$ for every vertex x. Now, let I be a distance-6-independent set. For each vertex $x \in I$ let $f(x)$ be the vertex found using Lemma 4.15 and $f(I) = \{f(x) \mid x \in I\}$. Since I is a distance-6-independent set, $f(I)$ is a distance-4-independent set of G and $|f(I)| = |I|$. Now, by Lemma 4.23, we know that $\mathcal{C} = V(G) \setminus f(I)$ is an identifying code of G. The first bound is now obtained by taking any maximal distance-6-independent set I.

If G has no forced vertex, we can directly consider a distance-4-independent set; again, a maximal distance-4-independent set of size at least $\frac{n}{1+\Delta-\Delta^2+\Delta^3}$ can be found because $|N_3[x]| \leq \frac{\Delta(\Delta-1)^3-2}{\Delta-2} = 1 + \Delta - \Delta^2 + \Delta^3$. ☆

4.3.2 A refined general approach

Unfortunately, the requirement of having a distance-4-independent set in Lemma 4.23 gives upper bounds of the form $n - \frac{n}{\Theta(\Delta^k)}$ for some $k \geq 3$. We refine this approach by using classical (distance-1-) independent sets instead. In order to still obtain an identifying code, we need the independent set to fulfill more complex properties. We start with the following proposition:

Proposition 4.25. *Let G be an identifiable graph, and let I be an independent set of G such that:*

1. *I contains no isolated vertex of G,*
2. *I contains no pair of false twins of G, and*
3. *for each pair u, v of adjacent vertices of G, $N[u] \ominus N[v] \not\subseteq I$ (in particular I contains no forced vertex),*

then, $V(G) \setminus I$ is an identifying code of G.

Proof. Observe that since I is an independent set and it contains no isolated vertex, $V(G) \setminus I$ is clearly a dominating set. Hence, by Observation 1.8, we just need to show that $V(G) \setminus I$ separates all pairs of vertices at distance at most 2 from each other. Note that by property number 3 of I, this is the case for all pairs of adjacent vertices. Let u, v be a pair of vertices with $d(u, v) = 2$. If either u or v do not belong to I, we are done. In particular, if u, v are false twins, by property number 2 of I, this is the case. Otherwise, both u, v belong to I; but then, none of their neighbours belong to I, and since they are not false twins, they are separated by one of these neighbours. ☆

The idea we will use next in order to apply Proposition 4.25 is to consider the set N of vertices that are not forced in G, and to build an independent set $I \subseteq N$ of G containing no pair of false twins. Then, we have to deal with property number 3 of I as required in Proposition 4.25. In order to do so, we build an auxiliary graph G' with vertex set I, for which we build an independent set $I' \subseteq I$, making sure that property number 3 is fulfilled. This idea is applied in the following lemma. Recall that $NF(G)$ was defined in Definition 4.14 as the ratio of the number of non-forced vertices of a graph G with respect to the order of G.

Lemma 4.26. *Let G be an identifiable graph of order n without isolated vertices and with maximum degree Δ having $n \cdot NF(G)$ non-forced vertices. There exists an independent set I of G fulfilling the three properties of Proposition 4.25 and having size at least $\frac{n \cdot NF(G)}{2\Delta(\Delta^2-\Delta+1)}$.*

Proof. Let N be the set of non-forced vertices in G: $|N| = n \cdot NF(G)$. We build an independent set $I_0 \subseteq N$ of G in a greedy way as follows. First, let $I_0 = \emptyset$ and $X = N$ be the set of candidate vertices. Pick an arbitrary vertex x from X; add x to I and remove all vertices of $N[x]$ from I. Remove as well each vertex x' which is a false twin of x. This process is repeated while $X \neq \emptyset$.

The correctness of this greedy algorithm is clear: no two adjacent vertices u, v can belong to I_0 (say u is picked; then, v is removed from the candidate set X as it belongs to $N[u]$) and similarly no two false twins belong to I_0. Moreover we have $|I_0| \geq \frac{|N|}{2\Delta}$. Indeed, at each step, at most 2Δ vertices are removed from X: say u is picked in I then we remove at most $\Delta + 1$ vertices from X for the set $N[u]$, and at most $\Delta - 1$ false twins of u.

Note that I_0 is an independent set of G fulfilling the first two properties required in Proposition 4.25, but there might be some pairs u, v of adjacent vertices that are not separated by $V(G) \setminus I_0$: $N[u] \ominus N[v] \subseteq I_0$. Consider the set of all such pairs u, v (note that the same vertex may participate to several such pairs) and let $\mathcal{S}$ be the collection of sets $\{N[u] \ominus N[v] \mid N[u] \ominus N[v] \subseteq I_0\}$. We remark that each set of $\mathcal{S}$ has at least two elements since we have no forced vertex in I_0. We now want to pick a vertex from each of the sets of $\mathcal{S}$ and remove it from I_0, in order to also fulfill property number 3 (this also preserves the other two properties of I_0). To this end, we build the auxiliary graph G' with vertex set I_0 and where for each set S of $\mathcal{S}$, we create an arbitrary edge within two vertices of S (recall that $|S| \geq 2$). Now, observe that it is sufficient to build a vertex cover of G' and remove it from I_0 to get an independent set fulfilling all three required properties. Recalling that any vertex cover is the complement of an independent set, it is sufficient to build an independent set $I \subseteq I_0$ of G' (remark that isolated vertices of G' may stay in I).

First of all, we note that $\Delta(G') \leq \Delta(\Delta - 1)$. Indeed, each vertex x of G' has at most one incident edge for each pair u, v such that $x \in N[u] \ominus N[v]$. Assuming $x \sim u$ but $x \not\sim v$, we have at most Δ choices for u, and then, $\Delta - 1$ choices for v (since we can exclude x from the second choice). By the same technique as in our earlier discussions, we can build a maximal independent set I of G' that has at size at least $\frac{|I_0|}{\Delta(G')+1} \geq \frac{|I_0|}{\Delta(\Delta-1)+1}$.

To summarize, we have $|I| \geq \frac{n \cdot NF(G)}{2\Delta(\Delta^2-\Delta+1)}$ and I fulfills all required properties. ☆

The following improvement of Theorem 4.24 demonstrates the usefulness of Proposition 4.25 and Lemma 4.26.

Theorem 4.27. *Let G be an identifiable graph of order n without isolated vertices and with maximum degree Δ. Then $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^4+2\Delta}$. If G has no forced vertices, $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^3}$.*

Proof. Let I be an independent set as constructed in Lemma 4.26. By Proposition 4.16, we have $NF(G) \geq \frac{1}{\Delta+1}$. Hence, we get $|I| \geq \frac{n}{2\Delta(\Delta^2-\Delta+1)} \cdot \frac{1}{\Delta+1} \geq \frac{n}{2\Delta^4+2\Delta}$. If G has no forced vertices, $NF(G) = 1$ and $|I| \geq \frac{n}{2\Delta(\Delta^2-\Delta+1)} \geq \frac{n}{2\Delta^3}$. In both cases, Proposition 4.25 completes the proof. ☆

This approach can be successfully applied to the class of quasi-line graphs, improving the bound of Theorem 4.27 as follows:

Theorem 4.28. *Let G be an identifiable quasi-line graph of order n without isolated vertices and with maximum degree Δ. Then $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^3+5\Delta^2+\Delta-2}$. If G has no forced vertices, $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^2+3\Delta-2}$.*

The bound of Theorem 4.28 will be improved using probabilistic methods in Section 4.4, hence we do not present its proof in the main body of the thesis, but in Appendix A.2.

4.3.3 An application to triangle-free graphs

In this section, we improve our previous bounds for the case of triangle-free graphs. We use the same ideas as in the two previous sections, but with more complicated arguments that lead to much improved bounds. We note that these results appeared in [FKKR12]; however for this thesis, we have chosen to present them as a more general formulation which we apply to various subclasses of triangle-free graphs in Subsection 4.3.3.5.

4.3.3.1 Proof ideas

The following proposition is similar to Proposition 4.25, but it is adapted to the case of triangle-free graphs.

Proposition 4.29. *Let G be an identifiable (not necessarily connected) triangle-free graph, and I, an independent set of G. Then, if the following properties hold, $V(G) \setminus I$ is an identifying code of G.*

1. *I contains no isolated vertex of G.*
2. *For any pair u, v of vertices of I, $N(u) \neq N(v)$ (i.e. I does not contain any pair of false twins).*
3. *For each vertex v of degree 1 in G, some vertex at distance 2 from v does not belong to I.*
4. *The graph $G[V(G) \setminus I]$ has no isolated edges.*

Proof. Let $\mathcal{C} = V(G) \setminus I$. Since I is an independent set and does not contain any isolated vertex, $\mathcal{C}$ is a dominating set. Let us now check the separation condition. Let u, v be an arbitrary pair of vertices of $V(G)$. We distinguish several cases.

If u and v are adjacent and both have degree at least 2, since they cannot form an isolated edge in $G[\mathcal{C}]$, a neighbour of either one of u, v belongs to $\mathcal{C}$ and separates them.

If u, v are adjacent and one of them, say u, has degree 1, since G is identifiable, v has at least one neighbour. Then, by the third property of I, there is a vertex at distance 2 of u in $\mathcal{C}$, separating u and v.

If u and v are false twins, they do not both belong to I and hence they are separated by themselves.

Finally, if u and v are not adjacent and are not false twins, if either u or v belong to $\mathcal{C}$, they are separated. If both u and v belong to I, all their neighbours belong to $\mathcal{C}$, and since they have distinct sets of neighbours they are separated. ☆

In order to prove our main result, we show how to build (large enough) independent sets in triangle-free graphs such that the three first conditions of Proposition 4.29 hold (see Lemma 4.34). However, it seems difficult to also ensure that the last condition holds while keeping the size of I reasonably large. Therefore, after building I, we compute the set M of isolated edges of $G[V \setminus I]$ and partition $V(G)$ into the end-vertices of M (set R) together with their neighbours (set L) on the one hand, and the remaining vertices, $V \setminus (L \cup R)$, on the other hand. We then build a sufficiently small (L, R)-*quasi-identifying code* $\mathcal{C}_1$, a variation of an identifying code which will be defined later (see Definition 4.32). This construction is done in Lemmas 4.35 and 4.36. Setting $\mathcal{C}_2$ as the complement of I restricted to $V \setminus (L \cup R)$, our final code is $\mathcal{C}_1 \cup \mathcal{C}_2$. We also combine this method with another technique (Proposition 4.13) which is suitable for the special case where the graph has a large number of false twins. The whole procedure is sketched in Algorithm 4.10.

This process is detailed in Subsection 4.3.3.4 (Theorem 4.37). All auxiliary results needed for this proof are developed in the next subsections.

4.3.3.2 Preliminary considerations

In the proof of our main result, we first construct an independent set I having some given properties. Then, we consider the set $V(G) \setminus I$ as a potential code, and modify it in order to identify those vertices which form isolated edges in $G[V(G) \setminus I]$. The following definition introduces a notion which helps to formalize this situation.

Definition 4.30. *Given a graph G together with an induced matching M of G, we denote by $R(M)$, the set of end-vertices of the edges of M and by $L(M)$, the set of neighbours of the vertices of $R(M)$: $L(M) = N(R(M)) \setminus R(M)$. M is called a* strong induced matching *if the following holds:*

- *$L(M)$ is an independent set in G.*
- *Each vertex x of $R(M)$ has degree at least 2 in G (i.e. $N(x) \cap L(M) \neq \emptyset$).*

Algorithm 4.10 Construction of an identifying code

Input: a nontrivial connected identifiable triangle-free graph $G = (V, E)$

Compute the set X of vertices having at least one false twin
if X is "small" **then**
  Use Lemma 4.34 to compute an independent set I of G fulfilling the three first properties listed in Proposition 4.29.
  Compute the set $R \subseteq V$ of vertices such that for each $v \in R$, v has a neighbour u where both u and v are of degree at least 2, and all the vertices of $N(u) \cup N(v) \setminus \{u, v\}$ belong to I.
  $L \leftarrow N(R) \setminus R$
  Compute an (L, R)-quasi-identifying code $\mathcal{C}_1$ of G using the constructions of Lemmas 4.35 and 4.36.
  $\mathcal{C}_2 \leftarrow (V \setminus (L \cup R)) \setminus I$
  $\mathcal{C} \leftarrow \mathcal{C}_1 \cup \mathcal{C}_2$
else {i.e. X is "big"}
  $\mathcal{C} \leftarrow$ an identifying code of G computed using Proposition 4.13.
end if
return $\mathcal{C}$

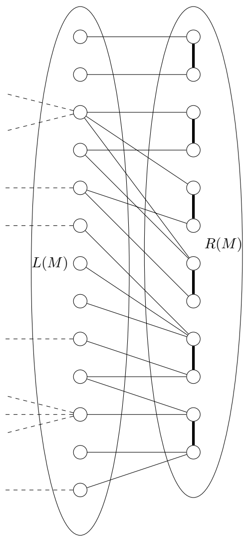


Figure 4.11: Example of a strong induced matching M (thick edges) in a triangle-free graph

An illustration of a strong induced matching is given in Figure 4.11. Note that in some graphs, one cannot necessarily find a strong induced matching. Indeed, if G is triangle-free, each edge of such a matching must belong to at least some induced path on four vertices.

Note that in any triangle-free graph G having a strong induced matching M, $G[L(M) \cup R(M)]$ has no isolated edge (i.e. two adjacent vertices of degree 1). Since in a triangle-free graph, a pair of twins necessarily forms an isolated edge, the following observation is immediate.

Observation 4.31. *Let G be a triangle-free graph having a strong induced matching M. Then $G[L(M) \cup R(M)]$ is identifiable.*

In order to construct small identifying codes of a triangle-free graph G having a strong induced matching M, we will construct special codes for the subgraph of G induced by set $L(M) \cup R(M)$. These codes are defined as follows.

Definition 4.32. *Let G be a triangle-free identifiable graph having a strong induced matching M with $L = L(M)$ and $R = R(M)$. Let $G' = G[L \cup R]$. We say that $\mathcal{C} \subseteq L \cup R$ is an (L, R)-*quasi-identifying code *of G if:*

1. *Each vertex of $L \cup R$ is dominated by some vertex of $\mathcal{C}$.*
2. *For each pair u, v of vertices in $L \cup R$, $\mathcal{C} \cap N_{G'}[u] \neq \mathcal{C} \cap N_{G'}[v]$, unless u and v both belong to L and $N_{G'}(u) = N_{G'}(v)$.*
3. *For each edge e of M, at least one of the vertices of e belongs to $\mathcal{C}$.*

Note that because of condition number 2 of Definition 4.32, an (L, R)-quasi-identifying code of G is not necessarily an $(L \cup R)$-identifying code of G. Conversely, because of condition number 3, an $(L \cup R)$-identifying code of G might not be an (L, R)-quasi-identifying code of G.

The following proposition shows that we can use an (L, R)-quasi-identifying code of G to construct a valid identifying code of G.

Proposition 4.33. *Let $G = (V, E)$ be an identifiable triangle-free graph having a strong induced matching M, with $L = L(M)$ and $R = R(M)$, and suppose that L does not contain any pair of false twins in G. Also suppose that there exists an (L, R)-quasi-identifying code $\mathcal{C}_1$ of G without $\mathcal{C}_1$-isolated vertices and a $(V \setminus (L \cup R))$-identifying code $\mathcal{C}_2$ of G where all the neighbours of vertices of L within $V \setminus (L \cup R)$ belong to $\mathcal{C}_2$.*[2] *Then, $\mathcal{C}_1 \cup \mathcal{C}_2$ is an identifying code of G.*

Proof. We show that each pair of vertices of G is separated. Since $\mathcal{C}_2$ is a $(V \setminus (L \cup R))$-identifying code, all pairs of vertices of $V \setminus (L \cup R)$ are separated. Since $\mathcal{C}_1$ is (L, R)-quasi-identifying and there are no $\mathcal{C}_1$-isolated vertices, each vertex x of $L \cup R$ is dominated by at least one vertex of $R \cap \mathcal{C}_1$ (see points number 1 and 3 of Definition 4.32), which we denote $f_{\mathcal{C}_1}(x)$. Moreover, by definition of sets L and R, no vertex of $V \setminus (L \cup R)$ is dominated by a vertex of R. Therefore, all pairs of vertices x, y with $x \in L \cup R$ and $y \in V \setminus (L \cup R)$ are separated by $f_{\mathcal{C}_1}(x)$. It remains to check the pairs of vertices of $L \cup R$. By contradiction, suppose there are two vertices u, v of $L \cup R$ which are not separated. By point number 2 of Definition 4.32, u and v belong to L and have the same neighbourhood within $L \cup R$. But since we assumed that they are not false twins and all their neighbours in $V \setminus (L \cup R)$ are in $\mathcal{C}_2$, u and v are separated by the neighbours they do not have in common, a contradiction. ☆

Next, we provide a lemma which shows how to build a large independent set in the subgraph of a triangle-free graph induced by the set of vertices that have no false twin and fulfilling property number 3 in Proposition 4.29.

Lemma 4.34. *Let G be an identifiable triangle-free graph on n vertices and maximum degree $\Delta \geq 3$, such that each subgraph H of G has an independent set of size at least $f(\Delta)|V(H)|$. Let Y be the set of all vertices of G having no false twin. Then $G[Y]$ has an independent set I with the following properties:*

[2] Note that if a $(V \setminus (L \cup R))$-identifying code $\mathcal{C}$ exists (i.e. $G[V \setminus (L \cup R)]$ is identifiable), then adding all neighbours of vertices of L to $\mathcal{C}$ yields an identifying code. In fact, any superset of an identifying code is still an identifying code.

1. *For each vertex u of degree 1 in G, there exists a vertex of G at distance 2 of u which does not belong to I.*

2. $|I| \geq \min\left\{\frac{1}{3}, f(\Delta)\right\}|Y|$

Proof. Let $I_1 \subseteq Y$ be the set of vertices of Y having degree 1 in G. Note that since G is identifiable, it has no isolated edges and therefore I_1 is an independent set in G (and $G[Y]$). Moreover since Y has no vertices having a false twin, all vertices of I_1 are at distance at least 3 from each other. Let T_1 be the set of vertices constructed as follows. All the vertices of I_1 belong to T_1. For each element s of I_1, its unique neighbour in G belongs to T_1, and some arbitrary neighbour at distance 2 of s belongs to T_1. Since all the vertices of I_1 are at distance at least 3 from each other, for each vertex s of I_1 there is a vertex at distance 2 of s belonging to $T_1 \setminus I_1$. We now set $Y_1 = T_1 \cap Y$. Note that we have $|I_1| \geq \frac{|T_1|}{3} \geq \frac{|Y_1|}{3}$ since for each vertex of I_1, at most three vertices of G have been inserted into T_1.

Now, let $Y_2 = Y \setminus Y_1$. By the previous construction, Y_2 neither contains a vertex of degree 1 in G, nor a neighbour of such a vertex. By our assumptions, $G[Y_2]$ has an independent set I_2 of size at least $f(\Delta)|Y_2|$.

Taking $I = I_1 \cup I_2$, we get an independent set of $G[Y]$ fulfilling the first property of the claim. Moreover, Y_1 and Y_2 form a partition of Y, $I_1 \subseteq Y_1$ and $I_2 \subseteq Y_2$. Hence, we have:

$$|I| \geq \frac{|Y_1|}{3} + f(\Delta)|Y_2| \geq \min\left\{\frac{1}{3}, f(\Delta)\right\}|Y|,$$

which completes the proof. ☆

4.3.3.3 Quasi-identifying the vertices around a strong induced matching

This subsection is devoted to the construction of small enough quasi-identifying codes.

Recall that in order to prove our main result, given a nontrivial identifiable connected triangle-free graph G, we will construct an independent set I and consider the (possibly empty) strong induced matching M such that $R(M)$ forms the set of isolated edges of $V(G) \setminus I$. In order to ensure that there are no isolated edges uv in $G[V(G) \setminus I]$, it would suffice to remove an arbitrary neighbour of either u or v from I. However, this could lead to a very large identifying code. Indeed, consider the example of a complete graph K_n where each edge is subdivided twice, K_n^*. The original vertices of K_n form a (maximal) independent set I and each original edge of K_n corresponds to an isolated edge in the subgraph of K_n^* induced by the complement of I, $K_n^*[V(K_n^*) \setminus I]$. Now, in K_n^*, getting rid of all isolated edges of $K_n^*[V(K_n^*) \setminus I]$ by removing vertices from I requires a vertex cover of K_n, that is, $n-1$ vertices. This would yield an identifying code of size $|V(K_n^*)| - 1$, which is not interesting.

Hence, in order to overcome this problem, we show in this subsection how to build an $(L(M), R(M))$-quasi-identifying code of bounded size. We first deal with the special case where all vertices of $R(M)$ have degree exactly 2 (note that by Definition 4.30 they must have degree at least 2).

Lemma 4.35. *Let G be an identifiable (not necessarily connected) triangle-free graph having a strong induced matching M where $L = L(M)$, $R = R(M)$, and all vertices of R have degree exactly 2. Then, there is an (L, R)-quasi-identifying code $\mathcal{C}$ of G having the following properties:*

1. $|\mathcal{C}| \leq |L| + \frac{|R|}{2}$.

2. *No vertex of R is $\mathcal{C}$-isolated.*

3. *At least half of the vertices of L belong to $\mathcal{C}$.*

Proof. In order to simplify its construction, let us first define the multigraph $G_{L,R} = (L, E)$ with vertex set L and in which there is an edge between two vertices l_1, l_2 of L if and only if there exist two vertices r_1, r_2 of R, such that l_1, r_1, r_2, l_2 is a 3-path in G. In other words, we contract every path of length 3 of $G[L \cup R]$ having both endpoints in L, into one edge. There can be multiple edges in $G_{L,R}$ (but no loops), since several disjoint 3-paths may join l_1 to l_2.

From $G_{L,R}$ we will build an oriented multigraph $\overrightarrow{G}_{L,R}$. Given an orientation of $\overrightarrow{G}_{L,R}$, we define the subset $I(\overrightarrow{G}_{L,R})$ of vertices of $L \cup R$ in the following way: all the vertices of L belong to $I(\overrightarrow{G}_{L,R})$, and for each arc $\overrightarrow{l_1 l_2}$ of $\overrightarrow{G}_{L,R}$ corresponding to the path $l_1 r_1 r_2 l_2$ in G, r_2 belongs to $I(\overrightarrow{G}_{L,R})$. Note that $|I(\overrightarrow{G}_{L,R})| = |L| + \frac{|R|}{2}$. An illustration is given in Figure 4.12, where the gray vertices belong to $I(\overrightarrow{G}_{L,R})$. Our aim is to construct an orientation of $\overrightarrow{G}_{L,R}$ for which $I(\overrightarrow{G}_{L,R})$ is the desired (L,R)-quasi-identifying code of G.

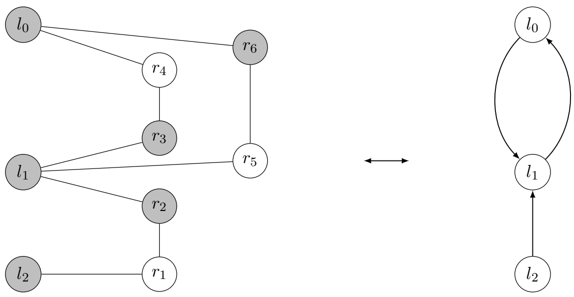


Figure 4.12: Correspondence between a special subset of $L \cup R$ and $\overrightarrow{G}_{L,R}$

We start by orienting the arcs of $\overrightarrow{G}_{L,R}$ in an arbitrary way. Note that $I(\overrightarrow{G}_{L,R})$ fulfills all three required properties of the statement of the lemma. Hence, if $I(\overrightarrow{G}_{L,R})$ is an (L,R)-quasi-identifying code of G, we are done. So, suppose this is not the case. Note that $I(\overrightarrow{G}_{L,R})$ fulfills conditions number 1 and 3 of Definition 4.32. Hence, there are pairs of vertices of $L \cup R$ which are not separated by $I(\overrightarrow{G}_{L,R})$. The only case where a pair l, r is not separated by $I(\overrightarrow{G}_{L,R})$, is when $l \in L$, $r \in R$, and both belong to $I(\overrightarrow{G}_{L,R})$, but they are only dominated by each other and themselves. This is equivalent to the case where l is of in-degree 1 in $\overrightarrow{G}_{L,R}$ (see Figure 4.13 for an illustration). In this case, in order to fix this problem, we modify the orientation of $\overrightarrow{G}_{L,R}$ as follows.

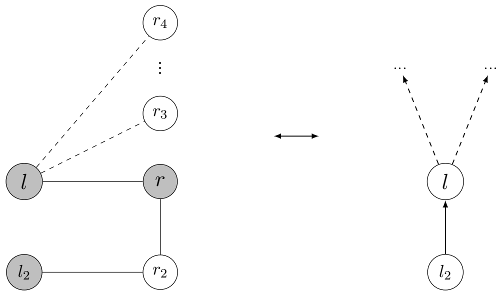


Figure 4.13: Vertices l and r are not separated

At first, consider a connected component $\overrightarrow{G}_1$ of $\overrightarrow{G}_{L,R}$, and construct an arbitrary spanning tree $\overrightarrow{T}_1$ of $\overrightarrow{G}_1$, rooted in some vertex l. Now, go through all vertices of $\overrightarrow{T}_1$, level by level in a bottom-up order from the leaves up to the root. Whenever the in-degree of the current vertex, v, is equal to 1, swap the orientation of the arc joining v to its parent in $\overrightarrow{T}_1$. Doing so, the in-degree of v in $\overrightarrow{G}_1$ becomes distinct from 1, and the in-degree of its parent is either incremented or decremented by 1. Note that except for the root l, all vertices of $\overrightarrow{G}_1$ have now an in-degree different from 1. This process is repeated for all connected components of $\overrightarrow{G}_{L,R}$.

Let $\mathcal{C} = I(\overrightarrow{G}_{L,R})$ be the new set corresponding to the new orientation. If $\mathcal{C}$ is an (L,R)-quasi-identifying code of G, we are done. Otherwise, as observed earlier, it means that some roots of the spanning trees we built, have in-degree 1 in $\overrightarrow{G}_{L,R}$. Let l be such a root with in-degree 1. Observe that l has a unique neighbour in $\mathcal{C} \cap R$, say r. Let r_2 be the neighbour of r

in R. It is sufficient to take out l from $\mathcal{C}$ and to replace it by r_2 in order to separate l from r in $G[L \cup R]$ (see Figure 4.14 for an illustration), without changing the cardinality of $\mathcal{C}$. Moreover, all neighbours of l are still separated from the other vertices because they are all in $R \setminus \mathcal{C}$ and therefore have a neighbour in $R \cap \mathcal{C}$, which itself has at least one neighbour in $L \cap \mathcal{C}$. Hence $\mathcal{C}$ is now an (L, R)-quasi-identifying code of G. Since the process did not change the cardinality of $\mathcal{C}$, we get property number 1 of the claim of the lemma.

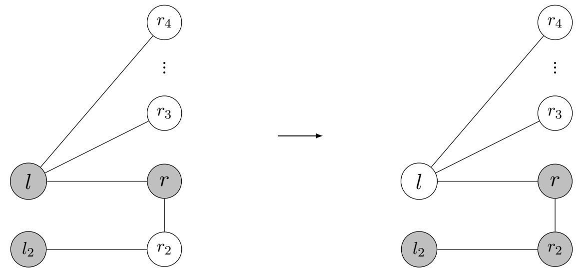


Figure 4.14: Local modification of the constructed code

Notice that there are at most $\frac{|L|}{2}$ connected components in $G[L \cup R]$ since each of them contains at least two vertices of L. Thus property number 3 of the claim of the lemma follows.

Property number 2 is fulfilled by the construction of $\mathcal{C}$ since in each pair of adjacent vertices of R, either it has a code vertex in L as a neighbour if there was no modification done, or in R if a switch of two elements of L and R was necessary. Moreover, for each such pair, at least one of its elements belongs to the code. This shows that $\mathcal{C}$ is an (L, R)-quasi-identifying code and completes the proof. ☆

We now deal with the general case, where the vertices of $R(M)$ have degree *at least* 2 as required in Definition 4.30.

Lemma 4.36. *Let G be an identifiable (not necessarily connected) triangle-free graph having a strong induced matching M, with $L = L(M)$ and $R = R(M)$. There exists a set L' of vertices of $L \cup R$ such that $|L'| \geq \frac{|L|}{3}$, and $\mathcal{C} = (L \cup R) \setminus L'$ is an (L, R)-quasi-identifying code of G having no $\mathcal{C}$-isolated vertices. If $\delta(G) \geq 3$, we have $|L'| \geq \frac{|L|}{2}$.*

Proof. Let us first divide sets L and R into the following subsets: let $R_1 \subseteq R$ be such that $r \in R_1$ if both r and its unique neighbour in R are of degree 2. Let $L_1 \subseteq L$ be the set of all neighbours of vertices of R_1, let $R_2 = R \setminus R_1$, and let $L_2 = L \setminus L_1$ (see Figure 4.15 for an illustration).

We can use Lemma 4.35 to construct an (L_1, R_1)-quasi-identifying code $\mathcal{C}_1$ of G such that the three properties described in the statement of Lemma 4.35 are fulfilled. Let $\mathcal{C}_1$ be such a code, in particular we have $|\mathcal{C}_1| \leq |L_1| + \frac{|R_1|}{2}$. Let us now describe the construction of two distinct (L, R)-quasi-identifying codes $\mathcal{C}_a$ and $\mathcal{C}_b$.

- **Construction of code $\mathcal{C}_a$.**
  We construct $\mathcal{C}_a$ such that $|\mathcal{C}_a| \leq |L_1| + \frac{|R_1|}{2} + |L_2| + \frac{|R_2|}{2} + \min\left\{\frac{|L_1|}{2}, \frac{|R_2|}{2}\right\}$, as follows.

  1. Put $\mathcal{C}_1$ into $\mathcal{C}_a$.
  2. Put L_2 into $\mathcal{C}_a$.
  3. For each pair r, r' of adjacent vertices of R_2, let r^* be one of them having at least two neighbours in L (by definition of R_2 either r or r' has this property). Put r^* into $\mathcal{C}_a$.
  4. For each pair r, r' of adjacent vertices of R_2, let r^* be the one which was put into $\mathcal{C}_a$ in the previous step. Check if r^* has less than two neighbours within $\mathcal{C}_a \cap L$ (this may happen if some of its neighbours are in L_1, and they do not belong to $\mathcal{C}_1$). If this is the case, pick an additional neighbour of r^* — which exists since r has at least two neighbours in L — and put it into $\mathcal{C}_a$. Note that this is done at most $\frac{|R_2|}{2}$ times.

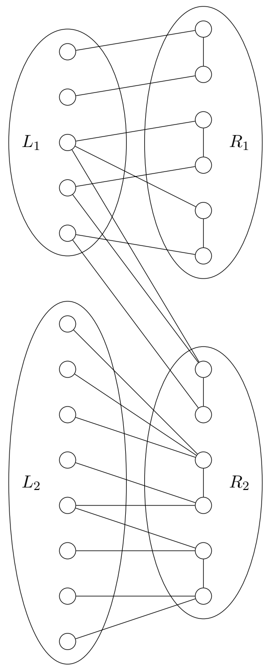


Figure 4.15: Illustration of sets L_1, L_2, R_1, and R_2

Moreover, at most $\frac{|L_1|}{2}$ new vertices from L_1 are put into $\mathcal{C}_a$ in such a way since by property number 3 of Lemma 4.35, there are at most $\frac{|L_1|}{2}$ vertices of L_1 not in $\mathcal{C}_1$.

5. Finally, consider each $\mathcal{C}_a$-isolated vertex l of L, take it out of $\mathcal{C}_a$ and put an arbitrary neighbour of l into $\mathcal{C}_a$ (this operation does not affect the size of $\mathcal{C}_a$).

- **Construction of code $\mathcal{C}_b$.**
  We construct $\mathcal{C}_b$ such that $|\mathcal{C}_b| \leq |L_1| + \frac{|R_1|}{2} + 3\frac{|R_2|}{2}$, as follows.

  1. Put $\mathcal{C}_1$ into $\mathcal{C}_b$.
  2. Put R_2 into $\mathcal{C}_b$.
  3. For each pair r, r' of adjacent vertices of R_2, one arbitrary neighbour in L of either r or r' is put into $\mathcal{C}_b$.
  4. Finally, in the same way as for the construction of $\mathcal{C}_a$, we get rid of each $\mathcal{C}_b$-isolated vertex l of L by taking l out of $\mathcal{C}_b$ and putting an arbitrary neighbour of l into $\mathcal{C}_b$ instead.

We omit the proof of the fact that $\mathcal{C}_a$ and $\mathcal{C}_b$ are (L, R)-quasi-identifying codes without $\mathcal{C}_a$-isolated or $\mathcal{C}_b$-isolated vertices. This proof can be found in Claim A.4 (Appendix A.3).

Let us now determine a lower bound on the cardinality of $(L \cup R) \setminus \mathcal{C}_x$, for $x \in \{a, b\}$. Taking into account that $|L_1| \leq |R_1|$, we obtain:

$$
\begin{aligned}
|(L \cup R) \setminus \mathcal{C}_a| &\geq |L_1| + |L_2| + |R_1| + |R_2| - |\mathcal{C}_a| \\
&\geq \frac{|R_1|}{2} + \frac{|R_2|}{2} - \min\left\{\frac{|L_1|}{2}, \frac{|R_2|}{2}\right\}
\end{aligned}
$$

Thus, both following equations hold:

$$
|(L \cup R) \setminus \mathcal{C}_a| \geq \frac{|R_1|}{2} + \frac{|R_2|}{2} - \frac{|L_1|}{2} \geq \frac{|R_2|}{2} \tag{4.4}
$$

$$
|(L \cup R) \setminus \mathcal{C}_a| \geq \frac{|R_1|}{2} + \frac{|R_2|}{2} - \frac{|R_2|}{2} = \frac{|R_1|}{2} \geq \frac{|L_1|}{2} \tag{4.5}
$$

Similarly,

$$
\begin{aligned}
|(L \cup R) \setminus \mathcal{C}_b| &\geq |L_1| + |L_2| + |R_1| + |R_2| - |\mathcal{C}_b| \\
&\geq |L_2| + \frac{|R_1|}{2} - \frac{|R_2|}{2} \\
&\geq |L_2| + \frac{|L_1|}{2} - \frac{|R_2|}{2} \\
&= |L| - \frac{|L_1|}{2} - \frac{|R_2|}{2}
\end{aligned} \tag{4.6}
$$

Hence intuitively, the previous equations show that our two codes fit to two different situations: $\mathcal{C}_a$ is useful when either $|L_1|$ or $|R_2|$ is large enough compared to $|L|$, whereas $\mathcal{C}_b$ is useful when $|L_1| + |R_2|$ is small enough compared to $|L|$. Let $\mathcal{C} \in \{\mathcal{C}_a, \mathcal{C}_b\}$ be the code having the minimum cardinality. We distinguish two cases.

Case a. $\delta(G) \leq 2$.
Using inequalities (4.4), (4.5) and (4.6) and denoting $b = \frac{\max\{|L_1|,|R_2|\}}{|L|}$ we get:

$$
\begin{aligned}
|(L \cup R) \setminus \mathcal{C}| &\geq \max\left\{\frac{|L_1|}{2}, \frac{|R_2|}{2}, |L| - \frac{|L_1|}{2} - \frac{|R_2|}{2}\right\} \\
&= \frac{|L|}{2} \cdot \max\left\{\frac{|L_1|}{|L|}, \frac{|R_2|}{|L|}, 2 - \frac{|L_1| + |R_2|}{|L|}\right\} \\
&\geq \frac{|L|}{2} \cdot \max\left\{\frac{\max\{|L_1|, |R_2|\}}{|L|}, 2 - \frac{2 \cdot \max\{|L_1|, |R_2|\}}{|L|}\right\} \\
&= \frac{|L|}{2} \cdot \max\{b, 2 - 2b\} \\
&\geq \frac{|L|}{2} \cdot \min_{b \geq 0}\{\max\{b, 2 - 2b\}\}.
\end{aligned}
$$

Note that $\min\limits_{b \geq 0}\{\max\{b, 2 - 2b\}\} = \frac{2}{3}$. Hence, we get:[3]

$$
|(L \cup R) \setminus \mathcal{C}| \geq \frac{|L|}{2} \cdot \frac{2}{3} = \frac{|L|}{3}.
$$

Case b. $\delta(G) \geq 3$.
In that case, we have $L_1 = R_1 = \emptyset$. Similar to Case a, setting $b = \frac{|R_2|}{2|L|}$, we get:

$$
\begin{aligned}
|(L \cup R) \setminus \mathcal{C}| &\geq \max\left\{\frac{|R_2|}{2}, |L| - \frac{|R_2|}{2}\right\} \\
&= |L| \cdot \max\left\{\frac{|R_2|}{2|L|}, 1 - \frac{|R_2|}{2|L|}\right\} \\
&= |L| \cdot \max\{b, 1 - b\} \\
&\geq |L| \cdot \min_{b \geq 0}\{\max\{b, 1 - b\}\}.
\end{aligned}
$$

Since $\min\limits_{b \geq 0}\{\max\{b, 1 - b\}\} = \frac{1}{2}$, we obtain:

$$
|(L \cup R) \setminus \mathcal{C}| \geq \frac{|L|}{2}.
$$

In both cases, putting $L' = (L \cup R) \setminus \mathcal{C}$, we are done. ☆

[3] We note that equality in the inequality is achieved when $|L_1| = |R_1| = |R_2| = 2|L_2|$.

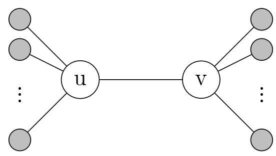


Figure 4.16: Vertices u, v with $(N(u) \cup N(v)) \setminus \{u, v\} \subseteq I$

4.3.3.4 The upper bound

We are now ready to prove the main theorem of this section. The proof has been sketched in Algorithm 4.10, we now provide all the details.

Theorem 4.37. *Let G be a connected identifiable triangle-free graph on n vertices with maximum degree $\Delta \geq 3$ such that each subgraph H of G has an independent set of size at least $f(\Delta)|V(H)|$. Let $f'(\Delta) = \min\left\{\frac{1}{3}, f(\Delta)\right\}$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + \frac{3}{f'(\Delta)}}$. If G has no false twins, then $\gamma^{ID}(G) \leq n - \frac{n \cdot f'(\Delta)}{3}$.*
If $\delta(G) \geq 2$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + \frac{3}{f(\Delta)}}$; if moreover, G has no false twins, then $\gamma^{ID}(G) \leq n - \frac{n \cdot f(\Delta)}{3}$.
If $\delta(G) \geq 3$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + \frac{2}{f(\Delta)}}$; if moreover, G has no false twins, then $\gamma^{ID}(G) \leq n - \frac{n \cdot f(\Delta)}{2}$.

Proof. For the proof, we assume that $\delta(G)$ is arbitrary. The bounds for $\delta(G) \geq 2$ and $\delta(G) \geq 3$ can be obtained similarly.[4]

Let $\mathcal{F} = \{F_1, \ldots, F_{|\mathcal{F}|}\}$ be the set of all nontrivial equivalence classes over the false twin relation $\equiv$ over $V(G)$. Let $X = \cup_{i=1}^{|\mathcal{F}|} F_i$ and $Y = V(G) \setminus X$. We distinguish two cases.

Case a. $|Y| \geq \frac{3n}{\Delta f'(\Delta) + 3}$.
In this case, let I be an independent set of $G[Y]$ given by Lemma 4.34. we have:

$$|I| \geq \min\left\{\frac{1}{3}, f(\Delta)\right\}|Y| = f'(\Delta)|Y|.$$

Consider all pairs u, v of vertices of G such that u and v are adjacent, both u and v have degree at least 2, and all the vertices of $N(u) \cup N(v) \setminus \{u, v\}$ belong to I (see Figure 4.16 for an illustration). Since all neighbours of u and v (except u and v themselves) are in I, these neighbours form an independent set. Let M be the (possibly empty) set of all edges uv such that u and v form such a pair. By the previous remark, M is a strong induced matching of G. Let us denote $L = L(M)$ and $R = R(M)$. Note that we have $L(M) \subseteq I$.

Let us now partition $V(G)$ into two subsets of vertices: $L \cup R$ on the one hand, and $V(G) \setminus (L \cup R)$ on the other hand. Such a partition is illustrated in Figure 4.17. Note that $G[L \cup R]$ is identifiable by Observation 4.31. Let us show that $G[V(G) \setminus (L \cup R)]$ is also identifiable. By contradiction, suppose it is not the case and let u, v be a pair of vertices such that $N_{G[V(G) \setminus (L \cup R)]}[u] = N_{G[V(G) \setminus (L \cup R)]}[v]$. Vertices u and v are therefore adjacent, and since G is triangle-free, neither u nor v has other neighbours within $G[V(G) \setminus (L \cup R)]$. Since G is identifiable, at least one of them has a neighbour in L. Suppose they both have a neighbour in L. Then by construction of I, u and v both do not belong to I. But then u and v should belong to R, a contradiction. Thus, one of them, say u, has degree 1 in G, and all neighbours of v belong to $L \subseteq I$. But by the first property of I in Lemma 4.34, at least one vertex at distance 2 of u does not belong to I, a contradiction.

We will now build two subsets $\mathcal{C}_1 \subseteq L \cup R$ and $\mathcal{C}_2 \subseteq V(G) \setminus (L \cup R)$ such that $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ is an identifying code of G.

[4] When $\delta(G) \geq 2$, since G has no degree 1-vertices, we do not need to use Lemma 4.34 and hence we can directly use $f(\Delta)$ instead of $f'(\Delta)$. When $\delta(G) \geq 3$, we can use the bound $|L'| \geq \frac{|L|}{2}$ instead of $|L'| \geq \frac{|L|}{3}$, as stated in Lemma 4.36; then, the threshold between Case a and Case b is replaced by $|Y| \geq \frac{2n}{\Delta f(\Delta) + 2}$.

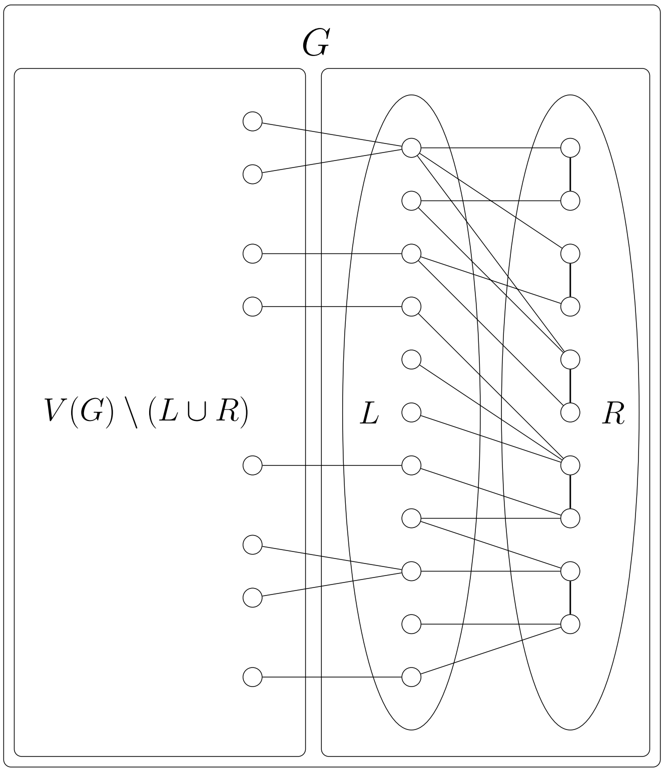


Figure 4.17: Partition of $V(G)$

- **Building $\mathcal{C}_1 \subseteq L \cup R$.**
  If $L \cup R = \emptyset$ we take $\mathcal{C}_1 = \emptyset$. Otherwise, we build $\mathcal{C}_1$ using Lemma 4.36: applying it to G and M, we know that there exists an (L, R)-quasi-identifying code $\mathcal{C}_1$ of G without $\mathcal{C}_1$-isolated vertices. From Lemma 4.36 we also know that $|L'| \geq \frac{|L|}{3}$, where $L' = (L \cup R) \setminus \mathcal{C}_1$.

- **Building $\mathcal{C}_2 \subseteq V(G) \setminus (L \cup R)$.**
  Again if $V(G) \setminus (L \cup R) = \emptyset$ we take $\mathcal{C}_2 = \emptyset$. Otherwise, we take $\mathcal{C}_2$ to be the complement of I in $V(G) \setminus (L \cup R)$: $\mathcal{C}_2 = (V(G) \setminus (L \cup R)) \setminus I$. Let us show that $\mathcal{C}_2$ is a $\big(V(G) \setminus (L \cup R)\big)$-identifying code of G.

  First, recall that $G' = G[V(G) \setminus (L \cup R)]$ is identifiable. Note that I does not contain any vertex v which is isolated in G'. Indeed, G does not contain any isolated vertex, hence if v is isolated in G', v has a neighbour in L. But $L \subseteq I$, a contradiction since I is an independent set. We also claim that for each vertex v of degree 1 in G', there is a vertex at distance 2 of v in G' not belonging to I. Let w be the unique neighbour of v in G'. If v is also of degree 1 in G, since G' has no pair of twins, by the first property of I in Lemma 4.34, w must have a neighbour x not in I. Vertex x cannot belong to L, hence it belongs to G' and we are done. Now, if v is not of degree 1 in G, all its neighbours in G other than w belong to L. But since G' is identifiable, w has at least one neighbour other than v, belonging to G' but not to I, since otherwise v and w would belong to set R. Finally, by construction of G', there are no isolated edges in $G[V(G') \setminus I]$.

  Under these conditions we can apply Proposition 4.29 on G' and on set I restricted to $V(G')$, which shows that $\mathcal{C}_2$ is a $\big(V(G) \setminus (L \cup R)\big)$-identifying code of G.

We now have an (L, R)-quasi-identifying code $\mathcal{C}_1$ of G without $\mathcal{C}_1$-isolated vertices, and showed that $\mathcal{C}_2$ is a $(V(G) \setminus (L \cup R))$-identifying code of G. Moreover, I does not contain any pair of false twins. Furthermore, since $\mathcal{C}_2$ is the complement of I in $G[V(G) \setminus (L \cup R)]$, all neighbours of L in $G[V(G) \setminus (L \cup R)]$ belong to $\mathcal{C}_2$. Therefore, we can apply Proposition 4.33 and $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$ is an identifying code of G.

Let us now upper-bound the size of $\mathcal{C}$. To this end, we lower-bound the size of its complement. From the construction of $\mathcal{C}_1$ and $\mathcal{C}_2$, we have $V(G) \setminus \mathcal{C} = (I \setminus L) \cup L'$.

Since $L \subseteq I$ and $|L'| \geq \frac{|L|}{3}$, we have $|(I \setminus L) \cup L'| \geq \frac{|I|}{3} \geq \frac{f'(\Delta)|Y|}{3}$.

We now distinguish two sub-cases. If $Y = V(G)$ (i.e. G has no false twins), we get:

$$|V(G) \setminus \mathcal{C}| \geq \frac{n \cdot f'(\Delta)}{3},$$

and hence,

$$\mathcal{C} \leq n - \frac{n \cdot f'(\Delta)}{3}.$$

Otherwise, by the assumption of the case distinction, we have $|Y| \geq \frac{3n}{\Delta f'(\Delta)+3}$. Hence:

$$\begin{aligned}|V(G) \setminus \mathcal{C}| &\geq \frac{f'(\Delta)|Y|}{3} \\ &\geq \frac{f'(\Delta)}{3} \frac{3n}{\Delta f'(\Delta) + 3} \\ &= \frac{n}{\Delta + \frac{3}{f'(\Delta)}}.\end{aligned}$$

Hence, $|\mathcal{C}| \leq n - \frac{n}{\Delta + \frac{3}{f'(\Delta)}}$.

Case b. $|Y| \leq \frac{3n}{\Delta f'(\Delta)+3}$.

Then, $|X| = |V(G) \setminus Y| \geq n - \frac{3n}{\Delta f'(\Delta)+3} = n\frac{\Delta f'(\Delta)}{\Delta f'(\Delta)+3}$. Since each set of $\mathcal{F}$ has size at most Δ, we have:

$$\begin{aligned}|\mathcal{F}| &\geq \frac{|X|}{\Delta} \\ &\geq \frac{f'(\Delta)}{\Delta f'(\Delta) + 3} n \\ &= \frac{n}{\Delta + \frac{3}{f'(\Delta)}}.\end{aligned}$$

Since $\Delta \geq 3$, G is not isomorphic to $\mathcal{C}_4$ and we can apply Proposition 4.13: G has an identifying code of size at most $n - |\mathcal{F}| \leq n - \frac{n}{\Delta + \frac{3}{f'(\Delta)}}$. ☆

4.3.3.5 Applying Theorem 4.37

A general bound

In order to use Theorem 4.37, we need to build (large enough) independent sets in triangle-free graphs. We first recall the following result of J. Shearer [178].

Theorem 4.38 ([178])**.** *Let G be a triangle-free graph on n vertices and average degree $\overline{d}$. Then G has an independent set of size at least $\frac{\overline{d}(\ln \overline{d} - 1)+1}{(\overline{d}-1)^2} n$.*

The following corollary of Theorem 4.38 is an approximate bound which is easier to deal with and which is tight enough for our purposes. It follows from the facts that $\overline{d}(G) \leq \Delta(G)$ and that when $x > 1$, the function $\frac{x(\ln x - 1)+1}{(x-1)^2}$ is decreasing. Moreover in that case, $\frac{x(\ln x-1)+1}{(x-1)^2} \geq \frac{\ln x - 1}{x}$ and for $x \geq 3$, $\frac{\ln x - 1}{x} > 0$.

Corollary 4.39. *Let G be a triangle-free graph on n vertices and maximum degree $\Delta \geq 3$. Then G has an independent set of size at least $\frac{\ln \Delta - 1}{\Delta} n$.*

We get the following corollaries of Theorem 4.37:

Corollary 4.40. *Let G be a connected identifiable triangle-free graph on n vertices and maximum degree $\Delta \geq 3$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + \frac{3\Delta}{\ln \Delta - 1}} = n - \frac{n}{\Delta(1+o_\Delta(1))}$. If G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\frac{3\Delta}{\ln \Delta - 1}} = n - \frac{n}{o(\Delta)}$.*

Corollary 4.41. *Let G be a connected identifiable triangle-free graph on n vertices, and maximum degree $\Delta \geq 3$, and minimum degree at least 3. Then $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + \frac{2\Delta}{\ln \Delta - 1}}$. If G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\frac{2\Delta}{\ln \Delta - 1}}$.*

We remark that due to the bound $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{o(\Delta)}$ when G has no false twins, any class of connected triangle-free graphs of maximum degree Δ having its minimum identifying code of size at least $n - \frac{n}{\Theta(\Delta)}$ should contain false twins. Note that this is the case of the complete $(\Delta-1)$-ary tree of height h, $T^h_{\Delta-1}$, as already mentioned in Chapter 2 (Theorem 2.33), whose leaves all are false twins, and also of the graphs of Construction 4.10.

Graphs of bounded chromatic number

It is an easy observation that any k-colourable graph has an independent set of size at least $\frac{n}{k}$, and any subgraph of a k-colourable graph is k-colourable. For example, bipartite graphs are 2-colourable, k-degenerate graphs are $(k+1)$-colourable [28, Exercise 14.1.5], graphs having no K_ℓ-minor are $O(\ell\sqrt{\ln(\ell)})$-colourable [135] (it is famously conjectured by H. Hadwiger that they are $(\ell-1)$-colourable [101]), and graphs of genus at most g are $\left(\left\lceil \frac{7+\sqrt{1+48g}}{2} \right\rceil\right)$-colourable [111].[5] In particular, planar triangle-free graphs are 3-colourable by H. Grötzsch's theorem [99].

We get the following corollary:

Corollary 4.42. *Let G be a nontrivial connected identifiable triangle-free graph on n vertices with maximum degree Δ and chromatic number $\chi(G)$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + 3\max\{3,\chi(G)\}}$, and if G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{3\max\{3,\chi(G)\}}$. In particular:*

- *If G is bipartite or planar, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+9}$; if moreover, G has no false twins, $\gamma^{ID}(G) \leq \frac{8n}{9}$.*
- *If G is k-degenerate, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+3(k+1)}$.*
- *If G has no K_ℓ-minor, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + O(\ell\sqrt{\ln(\ell)})}$.*
- *If G has genus at most g, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta + 3\left\lceil \frac{7+\sqrt{1+48g}}{2} \right\rceil}$.*

We can also strengthen the bounds when $\delta(G) \geq 2$ and $\delta(G) \geq 3$ using the corresponding cases in Theorem 4.37:

Corollary 4.43. *Let G be a nontrivial connected identifiable triangle-free graph on n vertices with maximum degree Δ and chromatic number $\chi(G)$. If $\delta(G) \geq 2$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+3\chi(G)}$, and if moreover, G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{3\chi(G)}$. If $\delta(G) \geq 3$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+2\chi(G)}$, and if moreover, G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{2\chi(G)}$. In particular:*

- *If G is bipartite and $\delta(G) \geq 2$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+6}$; if moreover, G has no false twins, $\gamma^{ID}(G) \leq \frac{5n}{6}$.*
- *If G is bipartite and $\delta(G) \geq 3$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+4}$; if moreover, G has no false twins, $\gamma^{ID}(G) \leq \frac{3n}{4}$.*
- *If G is planar and $\delta(G) \geq 3$, $\gamma^{ID}(G) \leq n - \frac{n}{\Delta+6}$; if moreover, G has no false twins, $\gamma^{ID}(G) \leq \frac{5n}{6}$.*

[5] We note that the latter three classes can in fact be summarized by the class of K_ℓ-minor-free graphs (for some ℓ) since k-degenerate graphs are $K_{O(k\sqrt{\ln(k)})}$-minor-free [188] and graphs of genus at most g are $K_{f(g)}$-minor-free for some function f [173].

Graphs of small maximum degree

We can improve our bounds for graphs having small maximum degree.

Theorem 4.44. *Let G be a nontrivial connected identifiable triangle-free graph on n vertices with maximum degree $\Delta \geq 3$ such that each subgraph H of G has an independent set of size at least $f(\Delta)|V(H)|$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, 9, \frac{3}{f(\Delta)}\right\}}$.*
If G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, 9, \frac{3}{f(\Delta)}\right\}}$.
If G has minimum degree at least 3, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, \frac{2}{f(\Delta)}\right\}}$; if moreover G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, \frac{2}{f(\Delta)}\right\}}$.

The proof of this theorem can be found in Appendix A.4. It uses a similar technique than the of the proof of Theorem 4.37, except that there is no case distinction on the number of vertices having a false twin, and that an independent set of G is computed using a specific greedy algorithm.

We can apply Theorem 4.44 using the lower bound $f(\Delta) \geq \frac{\ln \Delta - 1}{\Delta}$ of Corollary 4.39. It is also known (from Brook's theorem [32]) that when $\Delta(G) \geq 3$, any connected graph G which is not isomorphic to a complete graph is Δ-colourable, implying that $f(\Delta) \geq \frac{1}{\Delta}$ (this holds, of course, for triangle-free graphs). This bound is in fact stronger than the previous one for $\Delta < e^2 < 8$.

Corollary 4.45. *Let G be a nontrivial connected identifiable triangle-free graph on n vertices with maximum degree $\Delta \geq 3$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, 9, \min\{3\Delta, \frac{3\Delta}{\ln(\Delta)-1}\}\right\}}$.*
If G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, 9, \min\{3\Delta, \frac{3\Delta}{\ln(\Delta)-1}\}\right\}}$.
If G has minimum degree at least 3, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, \min\{2\Delta, \frac{2\Delta}{\ln(\Delta)-1}\}\right\}}$; if moreover G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, \min\{2\Delta, \frac{2\Delta}{\ln(\Delta)-1}\}\right\}}$.

We point out that the bounds of Corollary 4.45 are indeed stronger than the ones of Corollary 4.41 for relatively small values of the maximum degree. For example, for $\Delta < e^4 < 55$, $n - \frac{n}{\max\left\{2\Delta, 9, \min\{3\Delta, \frac{3\Delta}{\ln(\Delta)-1}\}\right\}}$ is smaller than $n - \frac{n}{\Delta + \frac{3\Delta}{\ln(\Delta)-1}}$.

We note that subcubic graphs seem already interesting due to the work in [112], where identifying *open* codes in subcubic graphs are studied (as mentioned earlier, the authors prove that an identifying open code in a not necessarily connected subcubic graph has size at most $\frac{3n}{4}$). Recall that in this specific case, the bound conjectured in Conjecture 4.4 is $\gamma^{\mathrm{ID}}(G) \leq \frac{2n}{3} + c$ for some constant c. The bounds of Corollary 4.45 for this case are as follows (we do not get any improvement when the graph has no false twins):

Corollary 4.46. *Let G be a nontrivial connected identifiable subcubic triangle-free graph on n vertices. Then $\gamma^{ID}(G) \leq \frac{8n}{9}$. If G is cubic, $\gamma^{ID}(G) \leq \frac{5n}{6}$.*

4.4 Using the probabilistic method to tackle Conjecture 4.4

In this section, we use the probabilistic method to show improved upper bounds on the identifying code number depending on the order and the maximum degree of the graph. We first introduce the probabilistic tools that we will need.

4.4.1 Probabilistic tools

We first recall a well-known probabilistic tool: the Lovász Local Lemma, introduced in [77]. We use its weighted version, a particularization of the general version where each event has an assigned weight. The proof can be found in [152].

Lemma 4.47 (Weighted Local Lemma [152]). *Let $\mathcal{E} = \{E_1, \dots, E_M\}$ be a set of (typically "bad") events such that each E_i is mutually independent of $\mathcal{E} \setminus (\mathcal{D}_i \cup \{E_i\})$ where $\mathcal{D}_i \subseteq \mathcal{E}$. Suppose that there exist some integer weights $t_1, \dots, t_M \geq 1$ and a real $p \leq \frac{1}{4}$ such that for each $1 \leq i \leq M$:*

- $\Pr(E_i) \leq p^{t_i}$, *and*
- $\sum_{E_j \in \mathcal{D}_i} (2p)^{t_j} \leq \frac{t_i}{2}$

Then $\Pr(\bigcap_{i=1}^{M} \overline{E_i}) \geq \prod_{i=1}^{M} (1 - (2p)^{t_i}) > 0$.

Note that in Lemma 4.47, since $p \leq \frac{1}{4}$ and $(1 - x) \geq e^{-(2\ln 2)x}$ in $x \in [0, 1/2]$, we have:

$$\Pr(\bigcap_{i=1}^{M} \overline{E_i}) \geq \exp\left\{-(2\ln 2) \sum_{i=1}^{M} (2p)^{t_i}\right\}. \tag{4.7}$$

We also use the following version of the well-known Chernoff bound, which is a reformulation of Theorem A.1.13 in [2].

Theorem 4.48 (Chernoff bound [2]). *Let X be a random variable of n independent trials of probability p, and let $a > 0$ be a real number. Then* $\Pr(X - np \leq -a) \leq e^{-\frac{a^2}{2np}}$.

4.4.2 The upper bound

In this section, we improve the upper bounds on the identifying code number by using the Weighted Local Lemma, stated in Lemma 4.47.

Theorem 4.49. *Let G be an identifiable graph G on n vertices having maximum degree $\Delta \geq 3$. Then* $\gamma^{ID}(G) \leq n - \frac{nNF(G)^2}{103\Delta}$.

Proof. Let F be the set of forced vertices of G, and $V' = V(G) \setminus F$. Note that $|V'| = nNF(G)$. By the definition of a forced vertex, any identifying code must contain all vertices of F.

In this proof, we first build a set S in a random manner by choosing vertices from V'. Then we exhibit some "bad" configurations — if none of those occurs, the set $\mathcal{C} = F \cup (V' \setminus S)$ is an identifying code of G. Using the Weighted Local Lemma, we compute a lower bound on the (non-zero) probability that none of these bad events occurs. Finally, we use the Chernoff bound to show that with non-zero probability, the size of S is also large enough for our purposes. This shows that such a "good" large set S exists, and it can be used to build an identifying code that has a sufficiently small size.

Let $p = p(\Delta)$ be a probability which will be determined later. We build the set $S \subseteq V'$ such that each vertex of V' independently belongs to S with probability p. Therefore the random variable $|S|$ follows a binomial distribution Bin($nNF(G)$,p) and has expected value $\mathbb{E}(|S|) = pnNF(G)$.

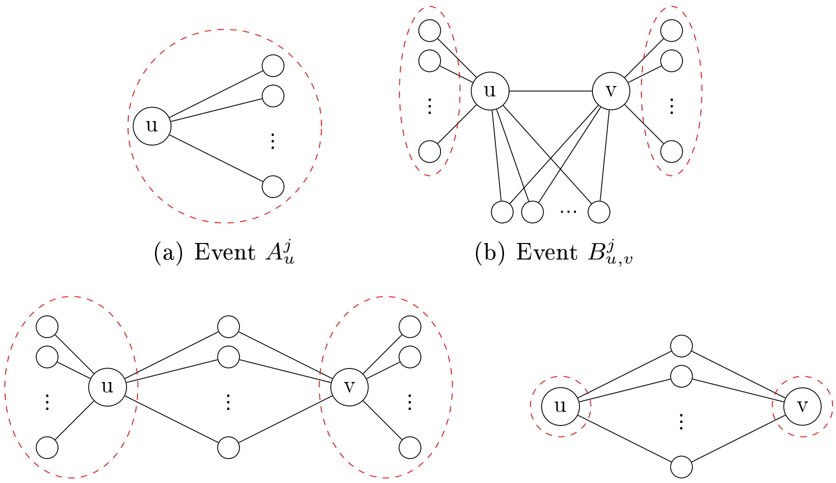
Let us now define the set $\mathcal{E}$ of "bad" events. These are of four types. An illustration of these events is given in Figure 4.18.

- **Type A^j ($2 \leq j \leq \Delta + 1$):** for each vertex $u \in V'$, let A_u^j be the event that $|N[u] = j|$ and $N[u] \subseteq S$.
- **Type B^j ($2 \leq j \leq 2\Delta - 2$):** for each pair $\{u, v\}$ of adjacent vertices, let $B_{u,v}^j$ be the event that $|(N[u] \ominus N[v])| = j$ and $(N[u] \ominus N[v]) \subseteq S$.
- **Type C^j ($3 \leq j \leq 2\Delta$):** for each pair $\{u, v\}$ of vertices in V' at distance two from each other, let $C_{u,v}^j$ be the event that $|(N[u] \ominus N[v])| = j$ and $(N[u] \ominus N[v]) \subseteq S$.
- **Type D:** for each pair $\{u, v\}$ of false twins in V', let $D_{u,v}$ be the event that $(N[u] \ominus N[v]) = \{u, v\} \subseteq S$.

For the sake of simplicity, we refer to the events of type A^j, B^j and C^j as events of type A, B and C respectively whenever the size of the symmetric difference is not relevant.

Events of type $B^1_{u,v}$ are not defined since then $|N[u] \ominus N[v]| = 1$ and F belongs to the code, so they never happen. Observe that the events $C^j_{u,v}$ and $D_{u,v}$ are just defined over the pairs of vertices in V' because if either u or v belongs to F, the event does not happen.

If no event of type A occurs, $V(G) \setminus S$ is a dominating set of G. If no event of type B occurs, all pairs of adjacent vertices are separated by $V(G) \setminus S$. If no event of type C or D occurs, all pairs of vertices at distance 2 from each other are separated. Thus by Observation 1.8, if no event of type A, B, C or D occurs, then $V(G) \setminus S$ is also a separating set of G, and therefore it is an identifying code of G.



(a) Event A^j_u (b) Event $B^j_{u,v}$

(c) Event $C^j_{u,v}$ (d) Event $D_{u,v}$

Figure 4.18: The "bad" events. The vertices in dashed circles belong to set S.

Let $V(E_i)$ denote the set of vertices that must belong to set S so that E_i holds (see Figure 4.18, where the sets $V(E_i)$ are the ones inside the dashed circles). We will say that a vertex $v \in V(G)$ participates to E_i, if $v \in V(E_i)$. We define the weight t_i of each event $E_i \in \mathcal{E}$ as $|V(E_i)|$. For $j \geq 2$ and for $T \in \{A^j, B^j, C^j, D\}$, let t_T be the weight of an event of type T (for an event $E_i \in \mathcal{E}$ of type T, $t_i = t_T$). We have:

$$t_{A^j} = j \qquad t_{B^j} = j \qquad t_{C^j} = j \qquad t_D = 2.$$

Some vertex x can participate to at most $\Delta + 1$ events of type A since if it participates to some event A^j_u, then $u \in N[x]$. Vertex x can participate to at most $\Delta(\Delta-1)$ events of type B: supposing $x \in V(B^j_{u,v})$ and u is adjacent to x, there are at most Δ ways to choose u, and at most $\Delta - 1$ ways to choose v among $N(u) \setminus \{x\}$. Observe that if $x = u$ or $x = v$, then $x \notin V(B_{u,v})$ (see Figure 4.18(b)). Similarly x can participate to at most $\Delta^2(\Delta-1)$ events of type C: for some event $C^j_{u,v}$, there are at most $\Delta(\Delta-1)$ possibilities if $x = u$ or $x = v$ and at most $\Delta(\Delta-1)^2$ if u or v is a neighbour of x. Finally, x can participate to at most $\Delta - 1$ events $\Delta_{u,v}$ since x can have at most $\Delta - 1$ false twins. For each type T of events ($T \in \{A^j, B^j, C^j, D\}$) and any vertex $v \in V(G)$, let us define $\mathrm{g}(v, T)$ to be the number of events E_i of type T such that $v \in V(E_i)$. Hence:

$$\begin{aligned} &\sum_{j=2}^{\Delta+1} \mathrm{g}(v, A^j) \leq d+1 \qquad && \sum_{j=2}^{2\Delta-2} \mathrm{g}(v, B^j) \leq \Delta(\Delta-1) \\ &\sum_{j=3}^{2\Delta} \mathrm{g}(v, C^j) \leq \Delta^2(\Delta-1) && \mathrm{g}(v, D) \leq \Delta - 1 \end{aligned} \tag{4.8}$$

Let us call E_{IC} the event that no event of $\mathcal{E}$ occurs. Using the Weighted Local Lemma, we want to show that $\Pr(E_{\mathrm{IC}}) > 0$. Given two events E_i and E_j of $\mathcal{E}$, we note $i \sim j$ if $V(E_i) \cap V(E_j) = \emptyset$. Observe that for any event E_i and any set $T \subseteq \{j : i \not\sim j\}$, we have $\Pr(E_i \mid \cap_{j \in T} \overline{E_j}) = \Pr(E_i)$, since the vertices are included in S with independent probabilities. This means that E_i is mutually independent from the set of all events E_j for which $V(E_i) \cap V(E_j) = \emptyset$.

In order to apply the Weighted Local Lemma (Lemma 4.47), the following conditions must hold for each event $E_i \in \mathcal{E}$:

$$\sum_{i \sim j} (2p)^{t_j} \leq \frac{t_i}{2}$$

The latter conditions are implied by the following ones (for each event $E_i \in \mathcal{E}$):

$$\sum_{j=2}^{\Delta+1} \sum_{v \in V(E_i)} \mathrm{g}(v, A^j)(2p)^{t_{A^j}} + \sum_{j=2}^{2\Delta-2} \sum_{v \in V(E_i)} \mathrm{g}(v, B^j)(2p)^{t_{B^j}} + \\ \sum_{j=3}^{2\Delta} \sum_{v \in V(E_i)} \mathrm{g}(v, C^j)(2p)^{t_{C^j}} + \sum_{v \in V(E_i)} \mathrm{g}(v, D)(2p)^{t_D} \leq \frac{t_i}{2}$$

Which are implied by:

$$t_i \cdot \max_{v \in V(E_i)} \left\{ \sum_{j=2}^{\Delta+1} \mathrm{g}(v, A^j)(2p)^{t_{A^j}} \right\} + t_i \cdot \max_{v \in V(E_i)} \left\{ \sum_{j=2}^{2\Delta-2} \mathrm{g}(v, B^j)(2p)^{t_{B^j}} \right\} + \\ t_i \cdot \max_{v \in V(E_i)} \left\{ \sum_{j=3}^{2\Delta} \mathrm{g}(v, C^j)(2p)^{t_{C^j}} \right\} + t_i \cdot \max_{v \in V(E_i)} \left\{ \mathrm{g}(v, D)(2p)^{t_D} \right\} \leq \frac{t_i}{2}$$

Using the bounds of Inequalities (4.8) and noting that for $p \leq 1/4$ and any j, $(2p)^{t_{A^j}} \leq (2p)^2$, $(2p)^{t_{B^j}} \leq (2p)^2$ and $(2p)^{t_{C^j}} \leq (2p)^3$, for any event E_i this equation is implied by:

$$(\Delta+1)(2p)^2 + \Delta(\Delta-1)(2p)^2 + \Delta^2(\Delta-1)(2p)^3 + (\Delta-1)(2p)^2 = 4\Delta^2 p^2 + 8\Delta^3 p^3 + 4\Delta p^2 - 8\Delta^2 p^3 \leq \frac{1}{2} \tag{4.9}$$

Hence, we fix $p = \frac{1}{k\Delta}$ where k is a constant to be determined later. Equation (4.9) holds for $k \geq 3.68$ for all $\Delta \geq 3$. In fact, in the following steps of the proof, we will assume that $k \geq 30$, and so Equation (4.9) will be satisfied for any $\Delta \geq 3$. Since $p \leq \frac{1}{4}$ and $\Pr(E_i) \leq p^{t_i}$ by the definition of t_i and the choice of S, the Weighted Local Lemma can be applied.

Let M_T be the number of events of type T, where $T \in \{A^j, B^j, C^j, D\}$. By Lemma 4.47 we have:

$$\Pr(E_{\mathrm{IC}}) \geq \prod_{j=2}^{\Delta+1} \prod_{i=1}^{M_{A^j}} (1 - (2p)^{t_{A^j}}) \prod_{j=2}^{2\Delta-2} \prod_{i=1}^{M_{B^j}} (1 - (2p)^{t_{B^j}}) \prod_{j=3}^{2\Delta} \prod_{i=1}^{M_{C^j}} (1 - (2p)^{t_{C^j}}) \prod_{i=1}^{M_D} (1 - (2p)^{t_D}).$$

Note that $\sum_{j=2}^{\Delta+1} M_{A^j} = nNF(G)$ since by definition there exists exactly one event A_u^j for each vertex of $u \in V'$. Moreover, $\sum_{j=2}^{2\Delta-2} M_{B^j} \leq \frac{n\Delta}{2}$ since there is exactly one event type $B_{u,v}^j$ for each edge $uv \in E(G)$ and at most $\frac{n\Delta}{2}$ edges in G. We also have that $\sum_{j=3}^{2\Delta} M_{C^j}$ is at most the number of pairs of vertices in V' at distance 2 from each other. This is also at most the number of paths of length 2 with both endpoints in V', which is upper-bounded by $\frac{nNF(G)\Delta(\Delta-1)}{2}$. Finally, M_D is the number of pairs of false twins in V', which is at most $nNF(G)\frac{\Delta-1}{2}$ by Proposition 4.12. Hence, we have:

$$\Pr(E_{\mathrm{IC}}) \geq (1-(2p)^2)^{nNF(G)}(1-(2p)^2)^{\frac{n\Delta}{2}}(1-(2p)^3)^{\frac{nNF(G)\Delta(\Delta-1)}{2}}(1-(2p)^2)^{\frac{nNF(G)(\Delta-1)}{2}}.$$

Using Lemma 4.47 (more precisely, we use Equation (4.7)) and the fact that $p = \frac{1}{k\Delta}$, we obtain:

$$\begin{aligned}\Pr(E_{\mathrm{IC}}) &\geq \exp\left\{-(2\ln 2)(2p)^2\left(NF(G)+\frac{\Delta}{2}+\frac{NF(G)\Delta(\Delta-1)2p}{2}+\frac{NF(G)(\Delta-1)}{2}\right)n\right\}\\ &\geq \exp\left\{-\frac{4\ln 2}{k^2\Delta}\left(\frac{2NF(G)}{\Delta}+1+\frac{2NF(G)}{k}+NF(G)\right)n\right\}\end{aligned}$$

Since $NF(G) \leq 1$ and it is assumed that $k \geq 30$, one can check that for any $\Delta \geq 3$:[6]

$$\Pr(E_{\mathrm{IC}}) \geq \exp\left\{-\frac{164\ln 2}{15k^2\Delta}n\right\}.$$

The Weighted Local Lemma shows that S has the desired properties with probability $\Pr(E_{\mathrm{IC}}) > 0$, implying that such a set exists. Note that we have no guarantee on the size of S. In fact, if $S = \emptyset$ then $V(G) \setminus S = V(G)$ is always an identifying code. Therefore we need to estimate the probability that $|S|$ is far below its expected size. In order to do this, we use the Chernoff bound of Theorem 4.48 by putting $a = \frac{nNF(G)}{c\Delta}$ where c is a constant to be determined. Let E_{BIG} be the event that $|S| - np > -\frac{nNF(G)}{c\Delta}$. We obtain:

$$\begin{aligned}\Pr(\overline{E_{\mathrm{BIG}}}) &\leq \exp\left\{-\frac{\left(\frac{nNF(G)}{c\Delta}\right)^2}{2pnNF(G)}\right\}\\ &= \exp\left\{-\frac{kNF(G)}{2c^2\Delta}n\right\}\end{aligned}$$

Now we have:

$$\begin{aligned}\Pr(E_{\mathrm{IC}} \text{ and } E_{\mathrm{BIG}}) &= 1-\Pr(\overline{E_{\mathrm{IC}}} \text{ or } \overline{E_{\mathrm{BIG}}})\\ &\geq 1-\Pr(\overline{E_{\mathrm{IC}}})-\Pr(\overline{E_{\mathrm{BIG}}})\\ &= 1-(1-\Pr(E_{\mathrm{IC}}))-\Pr(\overline{E_{\mathrm{BIG}}})\\ &= \Pr(E_{\mathrm{IC}})-\Pr(\overline{E_{\mathrm{BIG}}})\\ &\geq \exp\left\{-\frac{164\ln 2}{15k^2\Delta}n\right\}-\exp\left\{-\frac{kNF(G)}{2c^2\Delta}n\right\}\end{aligned}$$

Thus, $\Pr(E_{\mathrm{IC}} \text{ and } E_{\mathrm{BIG}}) > 0$ if $c < \frac{k^{3/2}NF(G)^{1/2}}{\sqrt{\frac{328\ln 2}{15}}}$. We (arbitrarily) set $c = \frac{k^{3/2}NF(G)^{1/2}}{\sqrt{22\ln 2}}$ in order to fulfill this condition.

Now we have to check that E_{BIG} implies that S is still large enough.

$$\begin{aligned}|S| &\geq \mathbb{E}(|S|)-\frac{nNF(G)}{c\Delta}\\ &= \frac{nNF(G)}{k\Delta}-\frac{nNF(G)}{c\Delta}\\ &= \left(\frac{1}{k}-\frac{\sqrt{22\ln 2}}{k^{3/2}NF(G)^{1/2}}\right)\frac{nNF(G)}{\Delta}\end{aligned} \tag{4.10}$$

[6] Note that this bound could be strengthened by assuming Δ to be large enough. Indeed, here the term $\frac{2NF(G)}{\Delta}$ can be as high as $\frac{2}{3}$ when $\Delta = 3$ and $NF(G) = 1$, but can be chosen to be as low as desired by assuming Δ to be larger. However we aim at giving a bound for any $\Delta \geq 3$, hence we use the weaker bound presented here.

Since $|S|$ must be positive, from Equation (4.10) we need $k^{3/2}NF(G)^{1/2} > \sqrt{22\ln 2}\,k$, which leads to $k = \frac{a_0}{NF(G)}$ for $a_0 > 22\ln 2$. Using all our previous assumptions, by derivating the expression of $|S|$, one can check that $|S|$ is maximized when $a_0 = \frac{99\ln 2}{2}$. Hence we set $k = \frac{99\ln 2}{2NF(G)}$.

Remark that under this condition and since $NF(G) \leq 1$, we have $k \geq 34$ and our assumption following Equation (4.9) that $k \geq 30$, is fulfilled.

Now, with $a_0 = \frac{99\ln 2}{2}$, we can see that:

$$|S| \geq \left(\frac{1}{k} - \frac{1}{c}\right)\frac{nNF(G)}{\Delta} = \frac{a_0^{1/2} - \sqrt{22\ln 2}}{a_0^{3/2}}\frac{NF(G)^2}{\Delta}n = \frac{2}{297\ln 2}\frac{NF(G)^2}{\Delta}n \geq \frac{NF(G)^2}{103\Delta}n.$$

Hence finally the identifying code $\mathcal{C} = V \setminus S$ has size

$$|\mathcal{C}| \leq n - \frac{nNF(G)^2}{103\Delta}.$$

☆

4.4.3 Corollaries of the bound

Note that for regular graphs, $NF(G) = 1$ because a forced vertex implies the existence of two vertices with distinct degrees. We obtain the following result:

Corollary 4.50 (Graphs with constant proportion of non-forced vertices)**.** *Let G be an identifiable graph on n vertices having maximum degree Δ and $NF(G) = \frac{1}{\alpha}$ for some constant $\alpha \geq 1$. Then $\gamma^{ID}(G) \leq n - \frac{n}{103\alpha^2\Delta}$. In particular if G is Δ-regular, $\gamma^{ID}(G) \leq n - \frac{n}{103\Delta}$.*

Using the bound $NF(G) \geq \frac{1}{\Delta+1}$ of Propositions 4.16 from Chapter 4.2.3, we obtain the following corollary:

Corollary 4.51 (General case)**.** *Let G be an identifiable graph on n vertices having maximum degree Δ. Then $\gamma^{ID}(G) \leq n - \frac{n}{103\Delta(\Delta+1)^2} = n - \frac{n}{\Theta(\Delta^3)}$.*

Using the bound $NF(G) \geq \frac{1}{\gamma(k)}$ for each graph G with clique number at most k from Proposition 4.22 (Chapter 4.2.3), we get the following extension of Corollary 4.50, where $c(k) \leq 103\gamma(k)^2$:

Corollary 4.52 (Graphs with bounded clique number)**.** *Let G be an identifiable graph on n vertices having maximum degree Δ and clique number smaller than k. Then $\gamma^{ID}(G) \leq n - \frac{n}{c(k)\Delta}$ for some constant $c(k)$ depending only on k. In particular this applies to triangle-free graphs, planar graphs, or more generally, graphs of bounded genus.*

We remark here that the previous corollaries support Conjecture 4.4. They also lead us to think that the difficulty of the problem lies in forced vertices.

4.5 Conclusion

In this chaper, we have shown that the maximum degree has a strong influence on the identifying code number, both for lower bounds and for upper bounds on this parameter.

We have given the first characterization of all graphs G reaching the lower bound $\gamma^{\mathrm{ID}}(G) \geq \frac{2|V(G)|}{\Delta(G)+2}$ of Theorem 2.29. However, we wonder what are the computational properties of the recognition problem for this class:

Question 4.53. *What is the computational complexity of deciding whether for a given graph G, the bound $\gamma^{ID}(G) = \frac{2|V(G)|}{\Delta(G)+2}$ holds? Equivalently, what is the complexity of deciding whether G can be obtained from Construction 4.1?*

We have conjectured in Conjecture 4.4 that the bound $\gamma^{\mathrm{ID}}(G) \leq n - \frac{n}{\Delta} + c$ holds and have given several bounds that approximate this conjecture for different graph classes. The best result known so far that is valid for all graphs is the bound $\gamma^{\text{ID}}(G) \le n - \frac{n}{\Theta(\Delta^3)}$ from Corollary 4.51, that we proved using the probabilistic method. We point out that the bound provided by our technique is tight whenever the graph has only few (or no) forced vertices (this includes the case of regular graphs). Hence, we do not expect an improvement in the probabilistic parts of the proof, but maybe one direction of research is to better understand the structure of forced vertices. This seems to be the difficult case in this approach.

The question raised through Conjecture 4.4 remains widely open. A big step forward would be to prove the following relaxation of Conjecture 4.4:

Question 4.54. *Is it true that for any identifiable graph G on n vertices, $\gamma^{ID}(G) \le n - \frac{n}{\Theta(\Delta(G))}$?*

For now, this question is answered in the affirmative for graphs with few forced vertices (Corollary 4.51), which includes regular graphs and graphs with bounded clique number, see Corollaries 4.50 and 4.52. Even stronger bounds are given for all triangle-free graphs (Corollary 4.40), and specifically for triangle-free graphs that have bounded chromatic number or no false twins (which includes bipartite graphs and planar triangle-free graphs), see Corollaries 4.40 and 4.42.

It would also be of interest to investigate whether the following special cases of Conjecture 4.4 hold. These cases might be easier than the general case because of their strong structural properties:

Question 4.55. *Does Conjecture 4.4 (or Question 4.54) hold for subcubic graphs? Does it hold for e.g. trees, planar graphs, line graphs?*

It would also be interesting to exhibit further constructions of graphs reaching the bound of Conjecture 4.4 than the ones we gave in Subsection 4.2.2.

Question 4.56. *Can the graphs G with maximum degree Δ such that $\gamma^{ID}(G) = n - \frac{n}{\Delta} + c$ (for some small constant c) be fully described?*

We have given the bound $NF(G) \ge \frac{1}{\Delta(G)+1}$ for the ratio of non-forced vertices in a graph G in Proposition 4.16. We showed that, given a value of $\Delta(G)$, it is tight for one special graph. However, we do not know whether there exist arbitrarily large *connected* graphs for which this bound is tight. If so, Conjecture 4.4 would be wrong as such graphs G would have their identifying code number of size at least $|V(G)| - \frac{|V(G)|}{\Delta(G)+1}$. A similar fact would hold for any bound smaller than $\frac{1}{\Delta(G)} - \frac{c}{|V(G)|}$ for every constant c.

Question 4.57. *Do there exist an infinite family of connected graphs such that the bound $NF(G) < \frac{1}{\Delta(G)} - \frac{c}{|V(G)|}$ holds for every constant c and every graph G of this family?*

Since we gave bounds for triangle-free graphs (that is, graphs of girth at least 4) in Subsection 4.3.3, it is natural to ask whether (much) stronger bounds on parameter γ^{ID} hold for graphs of larger girth. However, the answer to this question is negative because of the complete $(\Delta-1)$-ary tree of height h and on n vertices $T^h_{\Delta-1}$, which was already mentioned in Theorem 2.33. This graph has infinite girth and $\gamma^{\text{ID}}(T^h_{\Delta-1}) = \lceil n - \frac{n}{\Delta-1+1/\Delta} \rceil$ [24]. However, with an additional condition on the minimum degree of the graph, the question can be answered positively; we study this question in Section 5.1.

We finally make a remark regarding Theorem 2.28 from [96], which states that in any non-trivial infinite identifiable graph G whose vertices are all of finite degree, there exists a vertex x such that $V(G) \setminus \{x\}$ is an identifying code of G. Using Lemma 4.15 and similar to the proof of Theorem 4.24, we can strengthen their result as follows:

Corollary 4.58. *Let G be an infinite identifiable graph whose vertices all have finite non-zero degree. Then there exists an infinite set of vertices $I \subseteq V(G)$, such that $V(G) \setminus I$ is an identifying code of G.*

Chapter 5
Identifying codes in specific graph classes

THis chapter is devoted to the properties of identifying codes of graphs belonging to certain specific graph classes.

In Section 5.1, we study graphs with given minimum degree (at least 2) and girth 5. We show that such graphs admit relatively small identifying codes (in contrast to graphs with minimum degree 1 or girth less than 5). In particular, we use a constructive technique based on a DFS-tree in Subsection 5.1.1 to show that when their minimum degree is at least 2, graphs on n vertices and with girth 5 have an identifying code of size at most $\frac{7n}{8}$ (Theorem 5.1). In Subsection 5.1.2, we show that one can use a 2-dominating set (a notion that will be defined at that point) to construct an identifying code of a graph of girth 5 (Lemma 5.2). We then prove using the probabilistic method (more precisely, the Alteration Method) that these graphs, when having given minimum degree δ and n vertices, admit a 2-dominating set with its size of the order $\frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n$. This implies that for any such graph G, $\gamma^{\text{ID}}(G) \leq (1+o_\delta(1))\frac{3\ln\delta}{2\delta}n$ and $\gamma^{\text{ID}}(G) \leq \frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n$ when the average degree is not too large compared to the minimum degree of G (Theorem 5.3). We then use the latter result to determine the identifying code number of a random regular graph (with high probability). Indeed, we show that the distribution of the value of this parameter is asymptotically concentrated around one precise value (Theorems 5.4 and 5.5), leading to the bounds of Corollary 5.7: $\frac{\ln d-2\ln\ln d}{d}n \leq \gamma^{\text{ID}}(G) \leq \frac{\ln d+\ln\ln d+O_d(1)}{d}n$ holds w.h.p. for a random d-regular graph G.

In Section 5.2, we investigate interval graphs. We show that for any identifiable interval graph, the bound $\gamma^{\text{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ holds (Theorem 5.8 and Corollary 5.9). We then prove a few interesting propositions for identifying codes in identifiable *unit* interval graphs, leading to the bound $\gamma^{\text{ID}}(G) \geq \frac{|V(G)|+1}{2}$ when G belongs to this class (Theorem 5.11). Some of these propositions are of independent interest and will be used in Section 8.1.

Finally, in Section 5.3, we turn our attention to the class of line graphs. To this end, we use the notion of an edge-identifying code (it is much easier to deal with this concept than with the equivalent one of identifying codes in line graphs). This notion has not been introduced previously. We begin with giving useful preliminary lemmas about edge-identifying codes in Subsection 5.3.1 (Lemmas 5.14, 5.15 and 5.16), where we also determine the edge-identifying code number of complete graphs (Theorem 5.17). In Subsection 5.3.2, we prove the lower bound $\gamma^{\text{EID}}(G) \geq \frac{|V(G)|}{2}$ on the edge-identifying code number of any graph G, which implies a bound of the form $\gamma^{\text{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ for any identifiable line graph G (Theorem 5.19 and Corollary 5.20). We use this lower bound to determine the edge-identifying code number of hypercubes (Theorem 5.22), before refining this bound in Theorems 5.23 and 5.24, where we give an upper bound of roughly $\binom{\frac{4}{3}k}{2}$ on the size of a graph having an edge-identifying code of size k. We then provide general upper bounds on the edge-identifying code number of a graph in Subsection 5.3.3. Our main result there is to show that the edges of a minimal edge-identifying code induce a 2-degenerate graph (Theorem 5.27). This leads to the corollary that except for the graphs K_4 and K_4^-, $\gamma^{\text{EID}}(G) \leq 2|V(G)| - 5$ (Corollary 5.28). We conclude by another corollary showing that when a graph has average degree at least 5, Conjecture 4.4 holds for its line graph (Corollary 5.29).

The result of Subsection 5.1.1 is a revised version of a result that appeared in the author's master thesis [Fo09]. The results from Subsection 5.1.2 are joint work with G. Perarnau and appeared in [FP12]. The results from Section 5.2 are from [FKM+12] (joint work with A. Kosowski, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov) and those from Section 5.3 appeared in [FGN+12] (joint work with S. Gravier, R. Naserasr, A. Parreau and P. Valicov).

5.1 Graphs of given minimum degree and girth at least 5

We now investigate the identifying code number of graphs having no small cycles (i.e. no triangles and no 4-cycles), and also have minimum degree 2 or more. We show that these graphs have small identifying codes (compared to those graphs where at least one of these two conditions does not hold).

5.1.1 Minimum degree 2 and girth at least 5

We now consider identifiable graphs of girth at least 5 and of minimum degree at least 2. Let us first recall the notion of a DFS tree, that will be needed in this section.

A *DFS tree* of a connected graph G is a rooted spanning tree T of G constructed using *Depth-First Search*, i.e. using the following recursive procedure. Vertices of G can be marked or unmarked. Given a marked vertex v of G and the partially built tree T, the procedure goes as follows. If v has only marked neighbours in G, start the procedure at the parent of v (if v is the root of T, stop the whole process). Otherwise, select an arbitrary unmarked neighbour w of v, add it to T as a child of v, mark it, and recursively start the procedure at w. The tree T is built by selecting a vertex r as the root, marking it, and launching the procedure at r.

We are now ready to show the following upper bound for the minimum cardinality of an identifying code of such a graph:

Theorem 5.1. *Let G be an identifiable graph on n vertices, with minimum degree at least 2 and girth at least 5. Then $\gamma^{ID}(G) \leq \frac{7n}{8}$.*

Proof. We will assume that G is connected, but since the identifying code number of a disconnected graph is the sum of the identifying code numbers of the subgraphs induced by its connected components, this is enough to prove the bound.

Let T be a DFS tree of G rooted at some arbitrary vertex r. We denote the level of a vertex v (i.e., its distance to r) by $level(v)$. As any DFS-tree of G, T has the property that for any edge uv of G, either uv belongs to T, u is an ancestor of v in T, or v is an ancestor of u in T. Since G has girth at least 5, this implies that a vertex v with level i is not adjacent to any vertex with level $i-3, i-2, i, i+2$ or $i+3$.

Also note that since G has girth at least 5 and minimum degree 2, the level of a leaf v of T is at least 4: indeed, v has necessarily a neighbour other than its parent but can only be adjacent to its ancestors at levels at most $level(v) - 4$.

Let us now partition the vertices of G into four sets V_0, V_1, V_2 and V_3 defined as follows: for $i \in \{0,1,2,3\}$, $V_i = \{v \in V(G) \mid level(v) = i \bmod 4 \text{ in } T\}$. Let V_m ($m \in \{0,1,2,3\}$) be such a set with $|V_m| \geq \frac{n}{4}$ (by the pigeonhole principle, such a set necessarily exists, since otherwise there would be fewer than n vertices in G). Let $\mathcal{C}_0 = V(G) \setminus V_m$. An illustration is given in Figure 5.1 with $m = 1$. Black vertices belong to $\mathcal{C}_0$.

We now construct a set $\mathcal{C}$ from $\mathcal{C}_0$ as follows:

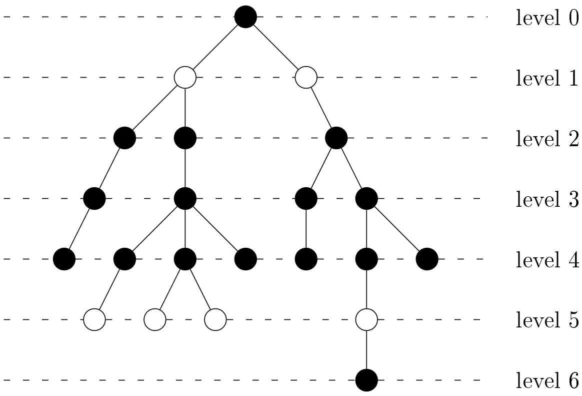


Figure 5.1: Example of a spanning tree of G and the set $\mathcal{C}_0 = V(G) \setminus V_1$ (black vertices).

1. For any set $L \subseteq V_m$ of leaves of T having the same parent p in T and with $N[l] \cap \mathcal{C}_0 = \{p\}$, we add an arbitrary ancestor[1] (other than p) of $|L| - 1$ of the leaves of L, to $\mathcal{C}_0$. See Figure 5.2 for an illustration.
2. For any leaf u of T having v as its parent and $N[u] \cap \mathcal{C} = N[v] \cap \mathcal{C} = \{u, v\}$, remove u from $\mathcal{C}$ and add the parent of v in T to $\mathcal{C}$. See Figure 5.3 for an illustration.
3. If the root r has a child u in T with $N[r] \cap \mathcal{C} = N[u] \cap \mathcal{C}$ (this implies $m = 2$), add an arbitrary child of u in T to $\mathcal{C}$.
4. If there is some leaf u of T such that $N[r] \cap \mathcal{C} = N[u] \cap \mathcal{C}$ (this implies $m = 1$), add an arbitrary child of r to $\mathcal{C}$.

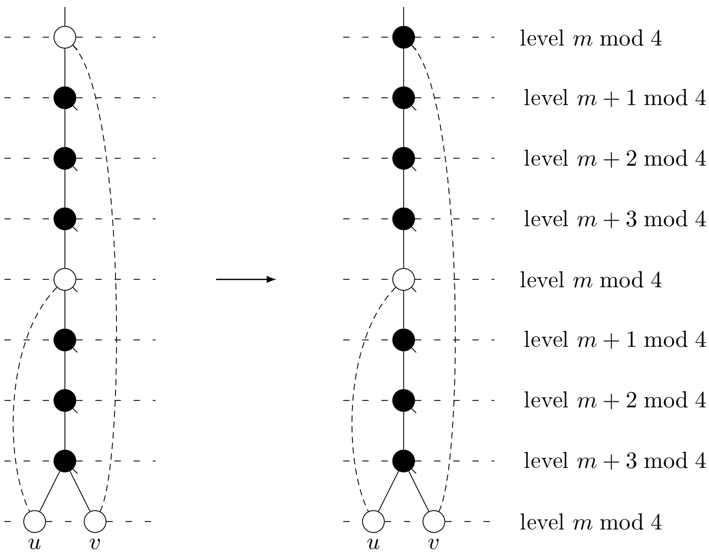


Figure 5.2: Step 1: two leaves u and v which are not separated by $\mathcal{C}_0$.

We claim that the resulting set $\mathcal{C}$ is an identifying code of G. First of all, observe that by the construction, for any two neighbours in T that are at consecutive levels, at least one of them belongs to $\mathcal{C}$. Hence, each vertex which is not in $\mathcal{C}$ has a neighbour in $\mathcal{C}$ from a different level in T, and $\mathcal{C}$ is a dominating set of G.

Note that $\mathcal{C}$ has the property that any vertex at level i which is neither the root of T, nor a leaf of T, is dominated by at least two vertices of $\mathcal{C}$ from two different levels among levels

[1]We note that such an ancestor necessarily exists since each vertex of G has degree at least 2 and a leaf of T can only be adjacent to some of its ancestors.

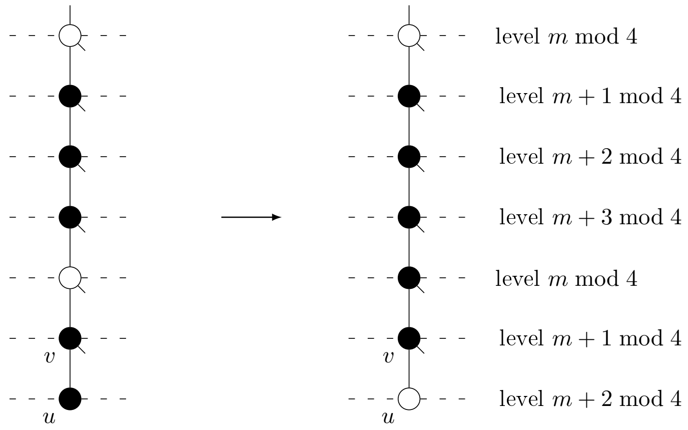


Figure 5.3: Step 2: a leaf u and its parent v are not separated by $\mathcal{C}_0$.

$i-1, i, i+1$. As a consequence, since G has girth at least 5, two such vertices that are not adjacent have at most one common neighbour, and hence they are separated by $\mathcal{C}$. Furthermore, two adjacent vertices u, v (with $level(u) < level(v)$) that are both neither the root nor a leaf of T, are also separated by some vertex of $\mathcal{C}$. Indeed, as noted just above, both u, v are dominated by at least two vertices in $\mathcal{C}$; moreover, since G has girth at least 5, they do not share any neighbour. Finally, it is easily observed that by the construction, if they both belong to $\mathcal{C}$ and are both dominated by $\{u, v\}$, either the parent of u or a child of v belongs to $\mathcal{C}$ and separates them.

It remains to check the separation condition for the root and the leaves of T. Assume first that the root r is not separated from some other vertex u. If u is a child of r, by step number 3 of the construction, u and r are separated, and we are done. If u is a leaf, again by step number 4, we are done. Finally, if u is some other vertex at level i, then, as mentioned earlier, u is dominated by two vertices from two different levels among levels $i-1, i, i+1$. But r cannot be adjacant to both of them (otherwise we would have a triangle in G), a contradiction.

Now, assume that a leaf l is not separated from some other vertex u. The case where u is the root has been considered in the last paragraph. Assume that u is another leaf of T. Then, both u, l belong to $V(G) \setminus \mathcal{C}$ as otherwise, they would be separated by the one belonging to the code (two leaves of T cannot be adjacent in G). But then, by construcion, both their respective parents belong to $\mathcal{C}$. Hence, u, l have the same parent p. But then, in step number 1 of the construction, we have added an ancestor a of at least one of them (say u) to $\mathcal{C}$. But this ancestor cannot be adjacent to l as otherwise, we have a 4-cycle a, l, p, u. Hence a separates u, l and we are done. Finally, assume that u is a vertex that is neither a root or a leaf. As earlier, if u is not the parent of l, since u is dominated by two vertices from two different levels among levels $i-1, i, i+1$, vertices u, l are separated by one of these vertices, and we are done. If u is the parent of l, by step number 2 of the construction, they are separated, and we are done.

It now remains to prove the bound. Note that $|\mathcal{C}_0| \leq \frac{3n}{4}$. When modifying $\mathcal{C}_0$, we added some vertices in step 1 of the construction. In that step, for each set L of sibling leaves from V_m, we added $|L|-1$ vertices to the code. The total number of vertices added to the code is at most the number of such leaves minus one. Together, the number of such leaves and the added vertices account for at most $\frac{n}{4}$ vertices (since they all belong to V_m). Hence, we have added at most $\frac{n}{8}-1$ vertices to the code in that step. In step 2, we have not increased the size of the code. Finally, in steps 3 and 4, we have added at most one vertex to the code; however, both steps are disjoint since they imply $m=2$ and $m=1$, respectively. So, in summary, $|\mathcal{C}| \leq \frac{7n}{8}$. ☆

5.1.2 Larger minimum degree and girth at least 5

This section is devoted to the study of graphs that have girth at least 5; the proofs use the probabilistic method (in particular, the classic "Alteration Method"). We will apply these results to random regular graphs in Section 5.1.3.

We start by defining an auxiliary notion that will be used in this section. A subset $\mathcal{D} \subseteq V(G)$ is called a 2-*dominating set* if for each vertex v of $V(G) \setminus \mathcal{D}$, $|N(v) \cap \mathcal{D}| \geq 2$ [84]. The next lemma shows that we can use a 2-dominating set to construct an identifying code.

Lemma 5.2. *Let G be an identifiable graph on n vertices having girth at least 5. Let $\mathcal{D}$ be a 2-dominating set of G. If the subgraph induced by $\mathcal{D}$, $G[\mathcal{D}]$, has no isolated edge, then $\mathcal{D}$ is an identifying code of G.*

Proof. First observe that $\mathcal{D}$ is dominating since it is 2-dominating. Let us check that $\mathcal{D}$ is also separating.

Note that all the vertices that do not belong to $\mathcal{D}$ are separated because they are dominated at least twice each and $g(G) > 4$.

Similarly, a vertex $x \in \mathcal{D}$ and a vertex $y \in V(G) \setminus \mathcal{D}$ are separated since y has two vertices which dominate it, but they cannot both dominate x (otherwise there would be a triangle or a 4-cycle in G).

Finally, consider two vertices of $\mathcal{D}$. If they are not adjacent they are separated by themselves. Otherwise, by the assumption that $G[\mathcal{D}]$ has no isolated edge and that G has no triangles, we know that at least one of them has a neighbour in $\mathcal{D}$, which separates them since it is not a neighbour of the other. ☆

The following theorem makes use of Lemma 5.2. The idea of the proof is inspired by a classic proof of a result on dominating sets which can be found in [2, Theorem 1.2.2].

Theorem 5.3. *Let G be a graph on n vertices with minimum degree δ and girth at least 5. Then $\gamma^{ID}(G) \leq \frac{3\left(\ln\delta+\ln\ln\delta+1+\frac{\ln\ln\delta}{\ln\delta}+\frac{1}{\ln\delta}\right)}{2\delta} = (1+o_\delta(1))\frac{3\ln\delta}{2\delta}n$. Moreover if G has average degree $\overline{d} = O_\delta(\delta(\ln\delta)^2)$ (in particular, when G is regular) then $\gamma^{ID}(G) \leq \frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n$.*

Proof. Let $S \subseteq V(G)$ be a random subset of vertices, where each vertex $v \in V(G)$ is added to S uniformly at random with probability p (where p will be determined later). For every vertex $v \in V(G)$, we define the random variable X_v as follows:

$$X_v = \begin{cases} 0 & \text{if } |N[v] \cap S| \geq 2 \\ 1 & \text{otherwise} \end{cases}$$

Let $T = \{v \mid X_v = 1\}$. This set contains, in particular, the subset of vertices which are not 2-dominated by S. Note that $|T| = \sum X_v$. Let us estimate the size of T. Observing that $|N[v] \cap S| \sim \text{Bin}(\deg(v)+1, p)$ and $\deg(v) \geq \delta$, we obtain:

$$\begin{aligned}
\mathbb{E}(|T|) &= \sum_{v\in V(G)} \mathbb{E}(X_v) \\
&\leq n\left((1-p)^{\delta+1} + (\delta+1)p(1-p)^{\delta}\right) \\
&= n(1-p)^{\delta}((1-p)+(\delta+1)p) \\
&\leq n(1+\delta p)e^{-\delta p}.
\end{aligned}$$

where we have used the fact that $1-x \leq e^{-x}$. Now, note that the set $\mathcal{D} = S \cup T$ is a 2-dominating set of G. We have $|\mathcal{D}| \leq |S| + |T|$. Hence

$$\begin{aligned}
\mathbb{E}(|\mathcal{D}|) &\leq \mathbb{E}(|S|) + \mathbb{E}(|T|) \\
&\leq np + n(1+\delta p)e^{-\delta p} \qquad (5.1)
\end{aligned}$$

Let us set $p = \frac{\ln\delta+\ln\ln\delta}{\delta}$. Plugging this into Equation (5.1), we obtain:

$$\begin{aligned}\mathbb{E}(|\mathcal{D}|) \leq & \frac{\ln\delta+\ln\ln\delta}{\delta}n+\frac{1+\ln\delta+\ln\ln\delta}{\delta\ln\delta}n\\ = & \frac{\ln\delta+\ln\ln\delta+1+\frac{\ln\ln\delta}{\ln\delta}+\frac{1}{\ln\delta}}{\delta}\\ = & \frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n.\end{aligned}$$

This shows that there exists at least one 2-dominating set $\mathcal{D}$ having this size.

Case 1: (general case) Note that we can use Lemma 5.2 by considering all pairs u, v of vertices of $\mathcal{D}$ forming an isolated edge in $G[\mathcal{D}]$, and add an arbitrary neighbour of either one of them to $\mathcal{D}$. Observe that such a vertex exists, otherwise u and v would be twins in G. Since there are at most $\frac{|\mathcal{D}|}{2}$ such pairs, we obtain a 2-dominating set of size at most $|\mathcal{D}|+\frac{|\mathcal{D}|}{2}=(1+o_\delta(1))\frac{3\ln\delta}{2\delta}n$ having the desired property. Now applying Lemma 5.2 completes Case 1.

Case 2: (sparse case) Whenever $\overline{d}=O_\delta(\delta(\ln\delta)^2)$, we can get a better bound by estimating the number of isolated edges of $G[\mathcal{D}]$. For convenience, we define the random variables Y_{uv} for each edge uv of G, as follows:

$$Y_{uv}=\begin{cases}1 & \text{if } N[u]\Delta N[v]\subseteq V(G)\setminus S, \text{ and}\\ 0 & \text{otherwise.}\end{cases}$$

An isolated edge in $G[\mathcal{D}]$ might have been created in several ways. First, at the initial construction step of S: if both u, v belong to S, but none of their other neighbours do which happens with probability at most $p^2(1-p)^{2\delta-2}$. A second possibility is in the step where we add the vertices of T to our solution. This could happen if both u, v were not dominated at all by S, which occurs with probability at most $(1-p)^{2\delta}$, or if exactly one of u, v was part of S and none of their neighbours were, which has probability at most $2p(1-p)^{2\delta-1}$. Thus, the total probability of having an isolated edge in $G[\mathcal{D}]$ is bounded from above as follows.

$$\Pr(Y_{uv}=1)\leq p^2(1-p)^{2\delta-2}+(1-p)^{2\delta}+2p(1-p)^{2\delta-1}=(1-p)^{2\delta-2}.$$

Using the previous observation together with the facts that $p=\frac{\ln\delta+\ln\ln\delta}{\delta}$ and $1-x\leq e^{-x}$, let us calculate the expected value of $Y=\sum_{uv\in E(G)}Y_{uv}$.

$$\mathbb{E}(Y)=\sum_{uv\in E(G)}\mathbb{E}(Y_{uv})\leq\frac{n\overline{d}}{2}(1-p)^{2\delta-2}\leq\frac{n\overline{d}}{2}e^{-(2\delta-2)p}=\frac{n\overline{d}e^{-2(\ln\delta+\ln\ln\delta)}}{2}=\frac{n\overline{d}}{2\delta^2(\ln\delta)^2}.$$

We construct U by picking an arbitrary neighbour of either u or v for each edge uv such that $Y_{uv}=1$. We have $|U|\leq Y$. The final set $\mathcal{C}=S\cup T\cup U$ is an identifying code. Now we have:

$$\mathbb{E}(|\mathcal{C}|)\leq\mathbb{E}(|S|)+\mathbb{E}(|T|)+\mathbb{E}(|U|)\leq\frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n+\frac{\overline{d}}{2\delta^2(\ln\delta)^2}n.$$

Using that $\overline{d}=O_\delta(\delta(\ln\delta)^2)$,

$$\mathbb{E}(|\mathcal{C}|)\leq\frac{\ln\delta+\ln\ln\delta+O_\delta(1)}{\delta}n \tag{5.2}$$

Then there exists some choice of S such that $|\mathcal{C}|$ has the desired size, and completes the proof. ☆

In fact, it is shown in the next section (Corollary 5.7) that Theorem 5.3 is asymptotically tight.

Moreover, note that Theorem 5.3 cannot be extended much in the sense that if we drop the condition on girth 5, we know arbitrarily large Δ-regular triangle-free graphs having large minimum identifying codes. For instance, Construction 4.10 of Chapter 4.2.2 provides a graph G on n vertices which satisfies $\gamma^{\mathrm{ID}}(G) = n - \frac{n}{2\Delta/3}$. Similarly, we cannot drop the minimum degree condition. Indeed, we saw in Theorem 2.33 that the $(\Delta-1)$-ary complete tree $T^h_{\Delta-1}$ of height h, which has maximum degree Δ, minimum degree 1 and infinite girth, also has a large identifying code number (i.e. $\gamma^{\mathrm{ID}}(T^h_{\Delta-1}) = n - \frac{n}{\Delta-1+1/\Delta}$, see Theorem 2.33).

5.1.3 An application to identifying codes of random regular graphs

From the study of regular graphs arises the question of the value of the identifying code number for most regular graphs. We know some lower and upper bounds for this parameter, but is it concentrated around some value? A good way to study this question is to look at random regular graphs. We refer to Chapter 2.2.3 for its definition.

Our bounds hold with high probability; in fact, they also include asymptotic terms in d, which means they are meaningful for sufficiently large d.

Theorem 5.4. *If $G \in \mathcal{G}(n,d)$ for some $d \geq 3$, $\gamma^{ID}(G) \leq \frac{\ln d + \ln\ln d + O_d(1)}{d} n$ w.h.p.*

Proof. First of all we have to show that almost all random regular graphs are identifiable.

Observe that the number of perfect matchings of K_{2m} is $(2m-1)!! = (2m-1)(2m-3)(2m-5)\ldots 1$. Fix a vertex u of G and let $N(u) = \{v_1, \ldots, v_d\}$. We bound from above the probability that u and v_1 are twins, i.e. $N[u] = N[v_1]$. The number of perfect matchings of K_{nd} such that in the resulting graph G of $\mathcal{G}(n,d)$, v_1 and v_2 are adjacent, is at most $(d-1)(d-1)(nd-2d-3)!!$. Indeed, there must be an edge between v_1 and v_2, which gives $(d-1)(d-1)$ possibilities. Since u has d neighbours, the number of possibilities for the remaining graph is the number of perfect matchings of $K_{nd-2d-2}$.

Analogously the number of perfect matchings with $v_2, v_3 \in N(v_1)$ is at most $(d-1)(d-1)(d-2)(d-1)(nd-2d-5)!!$. Thus we have:

$$\begin{aligned}
\Pr(N[u] = N[v_1]) &\leq \Pr(N[u] \subseteq N[v_1]) \\
&= \frac{(d-1)(d-1)(d-2)(d-1)\ldots 2(d-1)1(d-1)(nd-4d+1)!!}{(nd-2d-1)!!} \\
&\leq \frac{d^{d-1}(d-1)!}{(nd-2d-1)\ldots(nd-4d+3)} \\
&\leq \left(\frac{d}{n}\right)^{d-1} \qquad \text{for } n \text{ large enough.}
\end{aligned}$$

As we have at most $\frac{nd}{2}$ possible pairs of twins (one for each edge), by the union bound and since $d \geq 3$, for sufficiently large n we obtain:

$$\Pr(G \text{ has twins}) \leq \frac{nd}{2}\left(\frac{d}{n}\right)^{d-1}.$$

which tends to 0 as n tends to infinity.

Therefore, random regular graphs are identifiable w.h.p.

By (5.2), for any $G \in \mathcal{G}(n,d)$, we have a set $\mathcal{C}$ with $|\mathcal{C}| \leq \frac{\ln d + \ln\ln d + O_d(1)}{d} n$ that separates any pair of vertices except from the ones where both vertices belong to a triangle or a 4-cycle. We have to add some vertices to $\mathcal{C}$ in order to separate the vertices of these small cycles.

Classical results on random regular graphs (independently, [26, Corollary 2.19] and [198]) state that the random variables that count the number of cycles of length k, X_k, tend in distribution to independent Poisson variables with parameter $\lambda_k = \frac{1}{2k}(d-1)^k$.

Observe that:

$$\mathbb{E}(X_3) = \frac{(d-1)^3}{6} \quad \mathbb{E}(X_4) = \frac{(d-1)^4}{8},$$

i.e. a constant number of triangles and 4-cycles are expected.

Using Markov's inequality we can bound the probability of having too many small cycles:

$$\Pr(X_3 > t) \leq \frac{(d-1)^3}{6t} \qquad \Pr(X_4 > t) \leq \frac{(d-1)^4}{8t}.$$

Setting $t = \vartheta(n)$, where $\vartheta(n) \to \infty$, the previous probabilities are $o(1)$. Then w.h.p., we have at most $\vartheta(n)$ cycles of length 3 and $\vartheta(n)$ cycles of length 4.

Let $T = \{u_1, u_2, u_3\}$ be a triangle in G. As $d \geq 3$ there exists at least one vertex v_i outside the triangle (moreover, we showed that the graph has no twins w.h.p.). Since our graph is identifiable, for each ordered pair (u_i, u_j) there exists some vertex v_{ij}, such that $v_{ij} \in N(u_i) \backslash N(u_j)$. Observe that we can add v_{12}, v_{23} and v_{31} to $\mathcal{C}$ and then any pair of vertices from T will be separated.

If $T = \{u_1, u_2, u_3, u_4\}$ induces a K_4, each pair of vertices of T is contained in some triangle and is separated by the last step. If T induces a 4-cycle, adding T to $\mathcal{C}$ separates all the elements in T. Otherwise, T induces two triangles and adding T to $\mathcal{C}$ separates the two vertices which have not been separated in the last step.

After these two steps, we have added at most $7\vartheta(n)$ vertices to $\mathcal{C}$. Hence, for any $G \in \mathcal{G}(n, d)$ w.h.p. we obtain:

$$\gamma^{\text{ID}}(G) \leq \frac{\ln d + \ln\ln d + O_d(1)}{d} n + 7\vartheta(n) = \frac{\ln d + \ln\ln d + O_d(1)}{d} n.$$

Observe that the $\frac{O_d(1)}{d} n$ term contains the $7\vartheta(n)$ term. ☆

Theorem 5.4 shows that despite the fact that for any d, we know infinitely many d-regular graphs having a very large identifying code number (e.g. $n - \frac{n}{d}$ for the graphs of Construction 4.7 of Section 4.2.2), almost all d-regular graphs have a very small identifying code.

Moreover, $\gamma^{\text{ID}}(G)$ is concentrated, as the following theorem and its corollary show. In fact the following result might be already known, since a similar result is stated for independent dominating sets in [105]. However, we could not find it in the literature and decided to give a proof in Appendix A.5 for the sake of completeness.

Theorem 5.5. *Let $G \in \mathcal{G}(n, d)$, then w.h.p. all the dominating sets of G have size at least $\frac{\ln d - 2\ln\ln d}{d} n$.*

Since any identifying code is also a dominating set, we obtain the following immediate corollary.

Corollary 5.6. *Let $G \in \mathcal{G}(n, d)$, then w.h.p. $\gamma^{ID}(G) \geq \frac{\ln d - 2\ln\ln d}{d} n$.*

Plugging together Theorems 5.4 and 5.5, we obtain the following result.

Corollary 5.7. *Let $G \in \mathcal{G}(n, d)$, then w.h.p.*

$$\frac{\ln d - 2\ln\ln d}{d} n \leq \gamma^{ID}(G) \leq \frac{\ln d + \ln\ln d + O_d(1)}{d} n.$$

5.2 Interval graphs

We saw in Section 3.4, that there are infinitely many unit interval graphs G with identifying code number $|V(G)| - 1$ (i.e. the graphs A_k of our classification, see Corollary 3.28). However, it can be shown that the lower bound $\gamma^{\text{ID}}(G) \geq \lceil \log_2(|V(G)| + 1) \rceil$ is far from being tight for the class of interval graphs.

Theorem 5.8. *Let G be an identifiable interval graph having an identifying code of size k. Then $|V(G)| \leq \frac{k(k+1)}{2}$.*

Proof. Let $\mathcal{C} = \{c_1, \ldots, c_k\}$ be an identifying code of G with size k, where the intervals $c_1, \ldots, c_k$ are ordered increasingly by their right endpoint (let us denote by r_i, the endpoint of code interval c_i). Using this order, we define a partition $\mathcal{E}_1, \ldots, \mathcal{E}_k$ of $V(G)$ as follows. Let $\mathcal{E}_1$ be the set of intervals that start strictly before r_1. For any i with $2 \leq i \leq k-1$, let $\mathcal{E}_i$ be the set of intervals whose left endpoint lies within $[r_{i-1}, r_i[$, and LET $\mathcal{E}_k$ be the set of intervals whose left endpoint is at least r_{k-1}. Now, observe any interval I of $\mathcal{E}_i$ with $1 \leq i \leq k$ can only intersect one of the $k-i+1$ sets of consecutive code intervals $c_i, \dots, c_k$. Hence, $\mathcal{E}_i$ can contain at most $k-i+1$ intervals. Hence, in total G has at most $\sum_{i=1}^{k}(k-i+1) \leq \frac{k(k+1)}{2}$. ☆

Moreover, the bound of Theorem 5.8 is tight: consider the interval graph formed by the intersection of the following family of intervals: $\{]i,j[\mid 1 \leq i < j \leq k+1, i,j \in \mathbb{N}\}$, where the subfamily $\{]i,i+1[\mid 1 \leq i \leq k\}$ forms an identifying code of size k. We get the following corollary:

Corollary 5.9. *Let G be an identifiable interval graph on n vertices. We have $\gamma^{ID}(G) \geq \sqrt{2n+\frac{1}{4}} - \frac{1}{2}$, and this bound is tight.*

Proof. The inequality is easy to obtain from the bound of Theorem 5.8, using the fact that $k(k+1) = (k+\frac{1}{2})^2 - \frac{1}{4}$. ☆

When considering *unit* interval graphs, we can much improve the bound of Corollary 5.9. For this we first give the following propositions, which are interesting for their own sake. We recall that in a unit interval graph, since all intervals have unit length, there is a natural ordering of the vertices (according to the starting points of their corresponding intervals). We say that two vertices are *consecutive* if they are consecutive in this ordering. For a unit interval graph on n vertices, we denote its vertex set $\{v_1, \dots, v_n\}$, where the index of a vertex denotes its rank in the ordering.

In the next proposition, we concentrate on the separation of pairs of consecutive vertices.

Proposition 5.10. *Let G be an identifiable unit interval graph. Let $\mathcal{C}$ be an identifying code of G. Then each vertex $v_k \in \mathcal{C}$ separates at most two pairs of consecutive vertices, one on the left of v_k, and one on the right of v_k.*

Proof. Let $v_k \in \mathcal{C}$. Assume that v_k separates two pairs on the right of v_k: v_i, v_{i+1} and v_j, v_{j+1}, with $k < i < j$. Then we must have $v_k \sim v_i$ and $v_k \sim v_j$, but $v_k \not\sim v_i$ and $v_k \not\sim v_j$. But then we have $k < i < j < i+1$, a contradiction since v_i, v_{i+1} are consecutive. A symmetric argument holds for the left of v_k. ☆

In an identifying code, each of the $|V(G)|-1$ pairs of consecutive vertices must be separated. This together with Proposition 5.10 leads to the following lower bound:

Theorem 5.11. *Let G be an identifiable unit interval graph on n vertices. We have $\gamma^{ID}(G) \geq \frac{n+1}{2}$.*

Proof. Let $\mathcal{C}$ be an identifying code of G. By Proposition 5.10, we know that each vertex of $\mathcal{C}$ can only separate at most two of the pairs of consecutive vertices in G. Moreover, note that v_1 and v_n must also be dominated. However, any vertex dominating v_1 (v_n, respectively) can only separate one pair of consecutive vertices, on its right (on its left, respectively). Hence, we consider $n-1$ pairs of consecutive vertices to be separated, and two vertices (v_1 and v_n) to be dominated. If G has no universal vertex, each vertex of G can separate or dominate at most two of these $n+1$ entities, hence $|\mathcal{C}| \geq \frac{n+1}{2}$. If G has a universal vertex, each vertex is adjacent to either v_1 or v_n; hence, each vertex can separate at most one pair of consecutive vertices, and $|\mathcal{C}| \geq n-1 \geq \frac{n+1}{2}$. ☆

Note that the bound of Theorem 5.11 is tight for paths (see Theorem 2.30), which are unit interval graphs.

In the proofs of Proposition 5.10 and Theorem 5.11, we used the fact that an identifying code has to separate all pairs of consecutive vertices. The next proposition shows that, together with the domination condition, this is in fact sufficient. This result will be used in Section 8.1 when discussing the computational complexity of IDENTIFYING CODE restricted to unit interval graphs.

Proposition 5.12. *Let G be an identifiable unit interval graph. Then, a subset $\mathcal{C}$ of $V(G)$ is an identifying code of G if and only if it is a dominating set that separates all pairs of consecutive vertices.*

Proof. The necessary side is trivial since an identifying code separates all pairs, and hence it also separates pairs of consecutive vertices. For the sufficient side, assume that $\mathcal{C}$ is a dominating set that separates all pairs of consecutive vertices that are at distance at most 2 from each other. By Observation 1.8, we just need to check that also pairs of non-consecutive vertices are separated. Let v_i, v_j $(i < j)$ be such a pair. We know that v_i, v_{i+1} are separated by some vertex $v_k \in \mathcal{C}$. If $k \le i$, $v_k \sim v_i$ but $v_k \not\sim v_{i+1}$, hence $v_k \not\sim v_j$ and v_i, v_j are separated. Otherwise, $v_k \ge v_{i+1}$. If v_k dominates v_j, we are done. Otherwise, v_j is at distance at least 3 of v_i, so we are done. ☆

5.3 Line graphs

In this section, we study identifying codes in line graphs. To do so, we will use the equivalent notion of an *edge-identifying code*, where we wish to identify edges of a graph using other edges. This approach is similar to the one of studying e.g. the notion of *edge-colourings*, which are equivalent to vertex-colourings of line graphs. As for edge-colourings, edge-identifying codes are much more natural to handle than the similar concept for vertices in line graphs, and as demonstrated in this chapter, this fact helps us proving nice results.

Given a graph G and an edge e of G, we define $N[e]$ to be the set of edges adjacent to e together with e itself. An *edge-identifying code* of a graph G is a subset $\mathcal{C}_E$ of edges such that for each edge e the set $N[e] \cap \mathcal{C}_E$ is nonempty and uniquely determines e. More formally:

Definition 5.13. *Given a graph G, a subset $\mathcal{C}_E$ of $E(G)$ is an edge-identifying code of G if $\mathcal{C}_E$ is both:*

- *an* edge-dominating set *of G, i.e. for each edge $e \in E(G)$, $N[e] \cap \mathcal{C}_E \neq \emptyset$, and*
- *an* edge-separating code *of G, i.e. for each pair $e, f \in E(G)$ $(e \neq f)$, $N[e] \cap \mathcal{C}_E \neq N[f] \cap \mathcal{C}_E$.*

We will say that an edge e *separates* edges f and g if either e belongs to $N[f]$ but not to $N[g]$, or vice-versa. When considering edge-identifying codes we will assume the edge set of the graph is nonempty. The line graph $\mathcal{L}(G)$ of a graph G is the graph with vertex set $E(G)$, where two vertices of $\mathcal{L}(G)$ are adjacent if the corresponding edges are adjacent in G. It is easily observed that the notion of edge-identifying code of G is equivalent to the notion of (vertex-)identifying code of the *line graph* of G. Thus a graph G admits an edge-identifying code if and only if $\mathcal{L}(G)$ is identifiable. A pair of twins in $\mathcal{L}(G)$ can correspond in G to a pair of:
1. parallel edges;
2. adjacent edges whose non-common ends are of degree 1;
3. adjacent edges whose non common ends are of degree 2 but which are adjacent.

As in almost the whole of this thesis, we will consider simple graphs only, therefore avoiding the first case. A pair of edges as in the second or third case is called *pendant* (see Figure 5.4) and thus a graph is *edge-identifiable* if and only if it is *pendant-free*. The smallest size of an edge-identifying code of an edge-identifiable graph G is denoted by $\gamma^{\text{EID}}(G)$ and is called *edge-identifying code number* of G.

We notice that five edges of a perfect matching of the Petersen graph P_{10}, form an edge-identifying code of this graph (see Figure 5.5). The lower bound of Theorem 2.24 proves that $\gamma^{\text{EID}}(P_{10}) \geq 4$. Later, by improving this bound for line graphs, we will see that in fact $\gamma^{\text{EID}}(P_{10}) = 5$ (see Theorem 5.19 and Theorem 5.24).

5.3.1 First results

In this section we first give some easy tools which help for finding minimum-size edge-identifying codes of graphs. We then apply these tools to determine the exact values of γ^{EID} for some basic families of graphs.

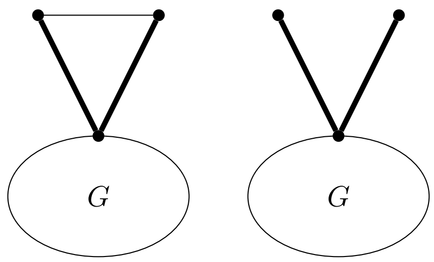


Figure 5.4: Two possibilities for a pair of pendant edges (thick edges) in G.

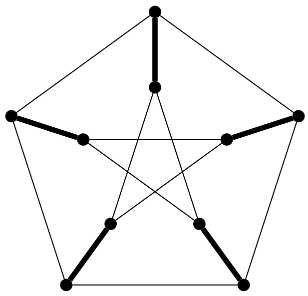


Figure 5.5: An edge-identifying code of the Petersen graph (thick edges).

Lemma 5.14. *Let G be a simple graph with girth at least 5. Let $\mathcal{C}_E$ be an edge cover of G such that the graph $(V(G), \mathcal{C}_E)$ is edge-identifiable. Then $\mathcal{C}_E$ is an edge-identifying code of G. In particular, if G has a perfect matching M, M is an edge-identifying code of G.*

Proof. The code $\mathcal{C}_E$ is an edge-dominating set of G because it covers all the vertices of G. To complete the proof, we need to prove that $\mathcal{C}_E$ is also an edge-separating code. Let e_1, e_2 be two edges of G. If $e_1, e_2 \in \mathcal{C}_E$, then $\mathcal{C}_E \cap N[e_1] \neq \mathcal{C}_E \cap N[e_2]$ because $(V(G), \mathcal{C}_E)$ is edge-identifiable. Otherwise, we can assume that $e_2 \notin \mathcal{C}_E$. If $e_1 \in \mathcal{C}_E$ and $\mathcal{C}_E \cap N[e_1] = \mathcal{C}_E \cap N[e_2]$, then e_2 must be adjacent to e_1. Let u be their common vertex and $e_2 = uv$. Since $\mathcal{C}_E$ is an edge cover, there is an edge $e_3 \in \mathcal{C}_E$ which is incident to v. However, e_3 cannot be adjacent to e_1 because G is triangle-free. Therefore e_3 separates e_1 and e_2. Finally, we assume neither of e_1 and e_2 is in $\mathcal{C}_E$. Then there are two edges of $\mathcal{C}_E$, say e_3 and e_4, adjacent to the two ends of e_1. But since G has neither C_3 nor C_4 as a subgraph, e_3 and e_4 cannot both be adjacent to e_2 and, therefore, e_1 and e_2 are separated. ☆

We note that in the previous proof the absence of C_4 is only used when the endpoints of e_1, e_2, e_3, e_4 could induce a C_4 which would not be adjacent to any other edge of $\mathcal{C}_E$. Thus, we have the following stronger statement:

Lemma 5.15. *Let G be a triangle-free graph. Let $\mathcal{C}_E$ be a subset of edges of G that covers vertices of G, such that $\mathcal{C}_E$ is edge-identifiable. If for no pair xy, uv of isolated edges in $\mathcal{C}_E$, the set $\{x, y, u, v\}$ induces a C_4 in G, then $\mathcal{C}_E$ is an edge-identifying code of G.*

We will also need the following lemma about edge-identifiable trees.

Lemma 5.16. *If T is an edge-identifiable tree on more than two vertices, then T has two vertices of degree 1, each adjacent to a vertex of degree 2.*

Proof. Take a longest path in T, then it is easy to verify that both ends of this path satisfy the condition of the lemma. ☆

We are now ready to determine the value of γ^{EID} of some families of graphs.

Theorem 5.17. *We have* $\gamma^{EID}(K_n) = \begin{cases} 5, & \text{if } n = 4 \text{ or } 5 \\ n-1, & \text{if } n \geq 6 \end{cases}$. *Furthermore, let $\mathcal{C}_E$ be an edge-identifying code of K_n of size $n-1$ ($n \geq 6$) and let $G_1, G_2, \ldots, G_k$ be the connected components of $(V(K_n), \mathcal{C}_E)$. Then exactly one component, say G_i, is isomorphic to K_1 and every other component G_j ($j \neq i$) is isomorphic to a cycle of length at least 5.*

The proof of Theorem 5.17 can be found in Appendix A.6.

We remark that the line graph of the complete bipartite graph $K_{n,n}$ is isomorphic to the cartesian product $K_n \square K_n$ (i.e. the graph with vertex set $K_n \times K_n$ and where $(u_1, u_2) \sim (v_1, v_2)$ if $u_1 = v_1$ or $u_2 = v_2$), whose identifying code number was determined in [98].[2]

Theorem 5.18 ([98])**.** $\gamma^{EID}(K_{n,n}) = \left\lfloor \frac{3n}{2} \right\rfloor$ *for* $n \geq 3$.

5.3.2 Lower Bounds

In this section, we give lower bounds on parameter γ^{EID} (and consequently on parameter γ^{ID} for line graphs) that improve the classic lower bound.

5.3.2.1 A first lower bound

Theorem 5.19. *Let G be a connected edge-identifiable graph. We have* $\gamma^{EID}(G) \geq \frac{|V(G)|}{2}$.

Proof. Let $\mathcal{C}_E$ be an edge-identifying code of G. Let G' be the subgraph induced by $\mathcal{C}_E$ and let $G_1, \ldots, G_s$ be the connected components of G'. Let n_i be the order of G_i and k_i be its size (thus $\sum_{i=1}^{s} k_i = |\mathcal{C}_E|$). Let $X = V(G) \setminus V(G')$ and n_i' be the number of vertices in X that are joined to a vertex of G_i in G. We show that $n_i' + n_i \leq 2k_i$. If $k_i = 1$, then clearly $n_i' = 0$ and $n_i' + n_i = 2 = 2k_i$. If G_i is a tree, then $n_i = k_i + 1$ and, by Lemma 5.16, G_i must have two vertices of degree 2 each having a vertex of degree 1 as a neighbour. Then no vertex of X can be adjacent to one of these two vertices in G. Moreover, each other vertex of G_i can be adjacent to at most one vertex in X. So $n_i' \leq k_i - 1$, and finally $n_i + n_i' \leq 2k_i$. If G_i is not at tree, we have $n_i \leq k_i$ and $n_i' \leq n_i$ and, therefore, $n_i' + n_i \leq 2k_i$. Finally, since G is connected, each vertex in X is connected to at least one G_i. Hence by counting the number vertices of G we have:

$$|V(G)| \leq \textstyle\sum_{i=1}^{s} (n_i + n_i') \leq 2 \sum_{i=1}^{s} k_i \leq 2|\mathcal{C}_E|. \qquad \star$$

Recall from Theorem 2.24 that $\gamma^{\text{ID}}(G)$ is bounded below by $\lceil \log_2(|V(G)|+1) \rceil$. As mentioned before, this bound is tight, and one of the main constructions achieving the bound is done as follows. Let $\mathcal{C}$ be a set of c isolated vertices. We build a graph G of order $2^c - 1$ such that $\mathcal{C}$ is an identifying code of G: for every subset X of $\mathcal{C}$ with $|X| \geq 2$, we associate a new vertex which is joined to all vertices in X and only to those vertices. It is easily seen that $\mathcal{C}$ is an identifying code of this graph. However, the graph built in this way is far from being a line graph as it contains $K_{1,c}$ as an induced subgraph (recall from Theorem 2.2 that line graphs are induced claw-free).

In fact, this lower bound turns out to be far from being tight for the class of line graphs, since we get as a corollary of Theorem 5.19, a lower bound of the order $\Omega\left(\sqrt{n}\right)$:

Corollary 5.20. *Let G be an identifiable line graph on n vertices. Then* $\gamma^{ID}(G) > \frac{\sqrt{2n}}{2}$.

Proof. Let $G = \mathcal{L}(H)$ be an identifiable line graph on n vertices. We have $n = |E(H)| \leq \binom{|V(H)|}{2} = \frac{|V(H)|^2}{2} - \frac{|V(H)|}{2} < \frac{|V(H)|^2}{2}$, hence $|V(H)| > \sqrt{2n}$. Applying the bound of Theorem 5.19 to H, we get:

$$\gamma^{\text{ID}}(G) = \gamma^{\text{EID}}(H) \geq \tfrac{|V(H)|}{2} > \tfrac{\sqrt{2n}}{2}. \qquad \star$$

We note that the bound of Corollary 5.20 is not tight (but the order of magnitude is); in fact, the constant $\frac{\sqrt{2}}{2}$ in the bound will be improved later (see Corollary 5.25).

[2] We note that this fact was overseen when we wrote [FGN+12], in which an independent proof of this result can be found.

5.3.2.2 Applying the lower bound to hypercubes

In this subsection, we apply Theorem 5.19 to the class of hypercubes. But first, Theorem 5.19 together with Lemma 5.15 leads to the following result:

Corollary 5.21. *Let G be a triangle-free edge-identifiable graph. Suppose G has a perfect matching M with the property that for any pair xy, uv of edges in M, the set $\{x, y, u, v\}$ does not induce a C_4. Then M is an optimal edge-identifying code and $\gamma^{EID}(G) = \frac{|V(G)|}{2}$.*

We note that in particular, if the girth of a graph G is at least 5 and G admits a perfect matching M, then M is a minimum-size identifying code of G. For example, the edge-identifying code of the Petersen graph given in Figure 5.5 is optimal.

As mentioned in the introduction, the problem of determining the identifying code number of hypercubes has proved to be a challenging one from both theoretical and computational points of view. In contrast, we show here that finding the edge-identifying code number of a hypercube is not so difficult.

Theorem 5.22. *For $d \geq 4$, we have $\gamma^{EID}(\mathcal{H}_d) = 2^{d-1}$.*

Proof. By Theorem 5.19, we have $\gamma^{\text{EID}}(\mathcal{H}_d) \geq 2^{d-1}$. We will construct by induction a perfect matching M_d of $\mathcal{H}_d$ such that no pair of edges induces a C_4, for $d \geq 4$. By Lemma 5.15, M_d will be an edge-identifying code of $\mathcal{H}_d$, proving the result. Two such matchings of $\mathcal{H}_4$, which are also disjoint, are presented in Figure 5.6. The matching M_5 can now be built using each of these two matchings of $\mathcal{H}_4$ — one matching per copy of $\mathcal{H}_4$ in $\mathcal{H}_5$. It is easily verified that M_5 has the required property. Furthermore, M_5 has the extra property that for each edge uv of M_5, u and v do not differ on the first coordinate (we build $\mathcal{H}_5$ from $\mathcal{H}_4$ by adding a new coordinate on the left, hence the first coordinate is the new one). We now build the matching M_d of $\mathcal{H}_d$ ($d \geq 6$) from M_{d-1} in such a way that no two edges of M_d belong to a 4-cycle in $\mathcal{H}_d$ and that for each edge uv of M_d, u and v do not differ on the first coordinate. To do this, let $\mathcal{H}'_1$ be the copy of $\mathcal{H}_{d-1}$ in $\mathcal{H}_d$ induced by the set of vertices whose first coordinate is 0. Similarly, let $\mathcal{H}'_2$ be the copy of $\mathcal{H}_{d-1}$ in $\mathcal{H}_d$ induced by the other vertices. Let $\mathcal{M}'_1$ be a copy of M_{d-1} in $\mathcal{H}'_1$ and let $\mathcal{M}'_2$ be a matching in $\mathcal{H}'_2$ obtained from $\mathcal{M}'_1$ by the following transformation: for $e = uv \in \mathcal{M}'_1$, define $\psi(e) = \sigma(u)\sigma(v)$ where $\sigma(x) = x + (1, 0, 0, \ldots, 0)$. It is now easy to check that the new matching $M_d = \mathcal{M}'_1 \cup \mathcal{M}'_2$ has both properties we need. ☆

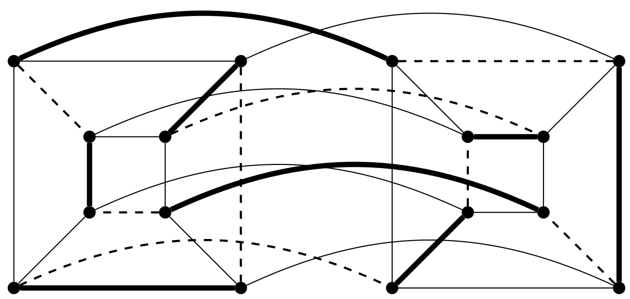


Figure 5.6: Two disjoint edge-identifying codes of $\mathcal{H}_4$.

We note that the formula of Proposition 5.22 does not hold for $d = 2$ and $d = 3$. For $d = 2$ the hypercube $\mathcal{H}_2$ is isomorphic to C_4 and thus $\gamma^{\text{EID}}(\mathcal{H}_2) = 3$. For $d = 3$, we note that an identifying code of size 4, if it exists, must be a matching with no pair of edges belonging to a 4-cycle. But this is not possible. An identifying code of size 5 is shown in Figure 5.7, therefore $\gamma^{\text{EID}}(\mathcal{H}_3) = 5$.

5.3.2.3 Refining the lower bound

In this section, we improve the lower bound of Corollary 5.20 by showing that the constant $\frac{\sqrt{2}}{2}$ is not tight (see Corollary 5.25).

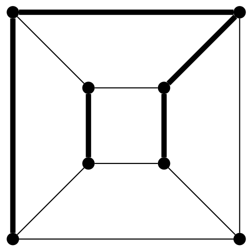


Figure 5.7: An optimal edge-identifying code of $\mathcal{H}_3$.

To do so, we upper-bound the number of edges in a graph G having an edge-identifying code $\mathcal{C}_E$ of size k. To avoid trivialities such as having isolated vertices we may assume G is connected. We note that this does not mean that the subgraph induced by $\mathcal{C}_E$ is also connected, in fact we observe almost the contrary, i.e. in most cases, an edge-identifying code of a minimum size will induce a disconnected subgraph of G. We first prove a lower bound for the case when an edge-identifying code induces a connected subgraph.

Theorem 5.23. *If an edge-identifying code $\mathcal{C}_E$ of a nontrivial graph G induces a connected subgraph of G which is not isomorphic to K_2, then G has at most $\binom{|\mathcal{C}_E|+2}{2}-4$ edges. Furthermore, equality can only hold if $\mathcal{C}_E$ induces a path.*

Proof. Let G' be the subgraph induced by $\mathcal{C}_E$. Since we assumed G' is connected, and since G' is edge-identifiable, it cannot have three vertices. Since we assumed $G' \not\cong K_2$, we conclude that G' has at least four vertices. For each vertex x of G', let $\mathcal{C}_E^x$ be the set of all edges incident to x in G'. Let $e = uv$ be an edge of G, then one or both of u and v must be in $V(G')$. Therefore, depending on which of these vertices belong to $\mathcal{C}_E$, e is uniquely determined by either $\mathcal{C}_E^u$ (if $u \in V(G')$ and $v \notin V(G')$), or $\mathcal{C}_E^v$ (if $u \notin V(G')$ and $v \in V(G')$), or $\mathcal{C}_E^u \cup \mathcal{C}_E^v$ (if both $u, v \in V(G')$). The total number of sets of this form can be at most $|V(G')| + \binom{|V(G')|}{2} = \binom{|V(G')|+1}{2}$, thus if $|V(G')| \leq |\mathcal{C}_E|$ we are done. Otherwise, since G' is connected, $|V(G')| = |\mathcal{C}_E| + 1$ and G' is an edge-identifiable tree on at least 4 vertices. If v is a vertex of degree 1 adjacent to u, then we have $\mathcal{C}_E^v = \{uv\}$ but $uv \in \mathcal{C}_E^u$ and, therefore, $\mathcal{C}_E^u = \mathcal{C}_E^u \cup \mathcal{C}_E^v$. On the other hand, by Lemma 5.16, there are two vertices of degree 2 that have neighbours of degree 1. Let u be such a vertex, let v be its neighbour of degree 1 and x be its other neighbour. Then $\mathcal{C}_E^v = \{uv\}$ and $\mathcal{C}_E^u = \{uv, ux\}$ and, therefore, $\mathcal{C}_E^u \cup \mathcal{C}_E^x = \mathcal{C}_E^v \cup \mathcal{C}_E^x$. Thus the total number of distinct sets of the form $\mathcal{C}_E^y$ or $\mathcal{C}_E^y \cup \mathcal{C}_E^z$ is at most $\binom{|\mathcal{C}_E|+2}{2} - 4$. But if equality holds there can only be two vertices of degree 1 in G' and hence $\mathcal{C}_E$ is a path. ☆

We note that if this bound is tight, then G' is a path. Furthermore, for each path P_{k+1} one can build many graphs which have P_{k+1} as an edge-identifying code and have $\binom{k+2}{2} - 4$ edges. The set of all these graphs will be denoted by $\mathcal{J}_k$. An example of such a graph is obtained from K_{k+2} by removing a certain set of four edges as shown in Figure 5.8. Note that every other member of $\mathcal{J}_k$ is obtained from the previous example by splitting the vertex that does not belong to P_{k+1} (but without adding any new edge).

Next, we consider the case when the subgraph induced by $\mathcal{C}_E$ is not necessarily connected.

Theorem 5.24. *Let G be an edge-identifiable graph and let $\mathcal{C}_E$ be an edge-identifying code of G with $|\mathcal{C}_E| = k$. Then we have:*

$$|E(G)| \leq \begin{cases} \binom{\frac{4}{3}k}{2}, & \text{if } k \equiv 0 \bmod 3, \\ \binom{\frac{4}{3}(k-1)+1}{2} + 1, & \text{if } k \equiv 1 \bmod 3, \\ \binom{\frac{4}{3}(k-2)+2}{2} + 2, & \text{if } k \equiv 2 \bmod 3. \end{cases}$$

Proof. Let G be a graph with maximum number of edges among all graphs with $\gamma^{\text{EID}}(G) = k$. It can be easily checked that for $k = 1, 2$ or 3, the maximum number of edges of G is 1, 3 or 6 respectively. For $k \geq 4$, we prove a slightly stronger statement: given an edge-identifying code

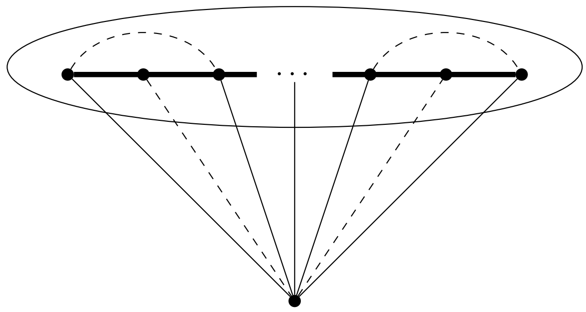


Figure 5.8: An extremal graph of $\mathcal{J}_k$ with its connected edge-identifying code

$\mathcal{C}_E$ of G of size k, all but at most two of the connected components of the subgraph induced by $\mathcal{C}_E$ must be isomorphic to P_4. When there is only one component not isomorphic to P_4, it must be isomorphic to a P_2, a P_5 or a P_6. If there are two such components, then they can be two copies of P_2, a P_2 with a P_5, or just two copies of P_5. This depends on the value of k mod 3.

To prove our claim let G be a graph as defined above, let $\mathcal{C}_E$ be an edge-identifying code of size k of G and let G' be the subgraph induced by $\mathcal{C}_E$. For each vertex $u \in V(G) \setminus V(G')$, we can assume that u has degree 1: if u has degree $d > 1$, with neighbours $v_1, \ldots, v_d$ necessarily in $V(G')$, then replace u by d vertices of degree 1: $u_1, \ldots, u_d$, connecting u_i to v_i. Then the number of edges does not change, and the code $\mathcal{C}_E$ remains an edge-identifying code of size k, thus it suffices to prove our claim for this new graph. Let $G'_1, G'_2, \ldots, G'_r$ be the connected components of G' with $|V(G'_i)| = n'_i$. For each $i \in \{1, \ldots, r\}$, let G_i be the graph induced by the vertices of G'_i and the vertices connected to G'_i only. To each vertex x of G' we assign the set $\mathcal{C}_E^x$ of edges in G' incident to x.

We first note that no G'_i can be of order 3, because there is no connected edge-identifiable graph on three vertices. If u and v are vertices from two disjoint components of G' with each component being of order at least 4, then the pair u, v is uniquely determined by $\mathcal{C}_E^u \cup \mathcal{C}_E^v$, thus by maximality of G, uv is an edge of G. If a component of G' is isomorphic to K_2, assuming u and u' are vertices of this component, then for any other vertex v of G' exactly one of uv or $u'v$ is an edge of G.

We now claim that each G'_i with $n'_i \geq 4$ is a path. By contradiction, if a G'_i is not a path, we replace G_i by a member $J_{n'_i-1}$ of $\mathcal{J}_{n'_i-1}$ with $P_{n'_i}$ being its edge-identifying code. Then we join each vertex of $P_{n'_i}$ to each vertex of each G'_j (with $j \neq i$ and $n'_j \geq 4$) and to exactly one vertex of each G_j with $n'_j = 2$. We note that the new graph still admits an edge-identifying code of size k. However, it has more edges than G. Indeed, while the number of edges connecting G'_i and the G'_j's ($j \neq i$) is not decreased, the number of edges in G_i is increased when we replace G_i by $J_{n'_i-1}$. This can be seen by applying Theorem 5.23 on G_i.

We now show that none of the G'_i's can have more than six vertices. By contradiction, suppose G'_1 is a component with $n'_1 \geq 7$ vertices (thus $n'_1 - 1$ edges). We build a new graph G^*_1 from G as follows. We take disjoint copies of $J_3 \in \mathcal{J}_3$ and $J_{n'_1-4} \in \mathcal{J}_{n'_1-4}$ with P_4 and $P_{n'_1-3}$ being, respectively, their edge-identifying codes. We now let $V(G^*_1) = V(J_3) \cup V(J_{n'_1-4}) \cup (V(G) \setminus V(G_1))$. The edges of J_3, $J_{n'_1-4}$ and $G - G_1$ are also edges of G^*_1. We then add edges between these three parts as follows. We join every vertex of P_4 to each vertex of $P_{n'_1-3}$. For $i = 2, 3, \ldots, r$ if $n'_i \geq 4$, join every vertex of G'_i to each vertex of $P_4 \cup P_{n'_1-3}$. If $n'_i = 2$, we choose exactly one vertex of G'_i and join it to each vertex of $P_4 \cup P_{n'_1-3}$. The construction of G^*_1 ensures that it still admits an edge-identifying code of size k, but it has more edges than G. In fact, the number of edges is increased in two ways. First, because $P_4 \cup P_{n'_1-3}$ has one more vertex than G'_1, the number of edges connecting $P_4 \cup P_{n'_1-3}$ to $G - G_1$ has increased (unless $r = 1$). More importantly, the number of edges induced by $J_3 \cup J_{n'_1-4}$ is $6 + \binom{n'_1-2}{2} - 4 + 4 \times (n'_1 - 3) = \frac{n_1'^2}{2} + \frac{3n'_1}{2} - 7$ which is strictly more than $|E(G'_1)| = \frac{n_1'^2}{2} + \frac{n'_1}{2} - 4$ for $n'_1 \geq 3$. Since $n'_1 \geq 7$, this contradicts the maximality of G.

With a similar method, the following transformations strictly increase the number of edges while the new graph still admits an edge-identifying code of size k:

1. Two components of G' each on six vertices transform into two graphs of $\mathcal{J}_3$ and a graph of $\mathcal{J}_4$.
2. One component of G' on six vertices and another component on five vertices transform into three graphs of $\mathcal{J}_3$.
3. One component of G' on six vertices and one on two vertices transform into two graphs of $\mathcal{J}_3$.
4. Three components of G' each on five vertices transform into four graphs of $\mathcal{J}_3$.
5. Two components of G' on five vertices and one on two vertices transform into three graphs of $\mathcal{J}_3$.
6. A component of G' on five vertices and two on two vertices transform into two graphs of $\mathcal{J}_4$.
7. Three components of G' each isomorphic to P_2 transform into a graph of $\mathcal{J}_3$.

For the proof of case 7, we observe that the number of edges identified by the three P_2's would be the same as the number of edges identified by the P_4. However, since $k \geq 4$, there must be some other component in G'. Moreover, the number of vertices of the three P_2's, which are joined to the vertices of the other components of G', is three, whereas the number of these vertices of the P_4, is four. Hence the maximality of G is contradicted.

We note that cases 1, 2 and 3 imply that if a component of G' is isomorphic to P_6, every other component is isomorphic to P_4. Then cases 4, 5 and 6 imply that if a component is isomorphic to P_5, then at most one other component is not isomorphic to P_4 and such component is necessarily either a P_2 or a P_5. Finally, case 7 shows that there can be at most two components both isomorphic to P_2.

We conclude that each of the components of G' is isomorphic to P_4 except for possibly two of them. These exceptions are dependent on the value of k mod 3 as we described. The formulas of the theorem can be derived using these structural properties of G. For instance, in the case $k \equiv 0 \bmod 3$, each component of G' is isomorphic to P_4. There are $\frac{k}{3}$ such components. For each component G'_i, there are six edges in the graph G_i. That gives $2k$ edges. The other edges of G are edges between two components of G'. By maximality of G, between two components of G', there are exactly 16 edges. There are $\binom{\frac{k}{3}}{2}$ pairs of components of G'. Hence, the number of edges in G is:

$$2k + 16\binom{\frac{k}{3}}{2} = \binom{\frac{4}{3}k}{2}.$$

The other cases can be proved with the same method. ☆

We note that this bound is tight and the examples were in fact built inside the proof. More precisely, for $k \equiv 0 \bmod 3$ we take $\frac{k}{3}$ disjoint copies of elements of $\mathcal{J}_3$ each having a P_4 as an edge-identifying code. We then add an edge between each pair of vertices coming from two distinct such P_4's. We note that the union of these P_4's is a minimum edge-identifying code of the graph. If $k \not\equiv 0 \bmod 3$, then we build a similar construction. This time we use elements from $\mathcal{J}_3$ with at most two exceptions that are elements of $\mathcal{J}_4$ or $\mathcal{J}_5$.

The above theorem can be restated in the language of line graphs as follows.

Corollary 5.25. *Let G be an identifiable line graph on $n \geq 4$ vertices. Then we have $\gamma^{ID}(G) \geq \frac{3\sqrt{2}}{4}\sqrt{n}$, and this bound is tight.*

Proof. Suppose G is the line graph of an edge-identifiable graph H ($\mathcal{L}(H) = G$). Let $k = \gamma^{\text{ID}}(G) = \gamma^{\text{EID}}(H)$, and let n be the number of vertices of G ($n = |E(H)|$). Then, after solving the quadratic inequalities of Theorem 5.24 for k, we have:

$$\begin{aligned}
k \geq& \frac{3}{8} + \frac{3\sqrt{8n+1}}{8}, \text{ for } k \equiv 0 \bmod 3 \\
k \geq& \frac{5}{8} + \frac{3\sqrt{8n-7}}{8}, \text{ for } k \equiv 1 \bmod 3 \\
k \geq& \frac{3}{8} + \frac{3\sqrt{8n-15}}{8}, \text{ for } k \equiv 2 \bmod 3
\end{aligned}$$

It is then easy to check that the right-hand side of each of the three inequalities is at least as $\frac{3\sqrt{2}}{4}\sqrt{n}$ for $n \geq 3$. The tightness follows from the discussion of the previous paragraph. ☆

5.3.3 Upper bounds

In this section, we present upper bounds for parameter γ^{ID} in line graphs.

5.3.3.1 Line graphs with identifying code number their order minus one

We saw that a general bound, only in terms of the order of a graph, is provided by Theorem 2.27. Furthermore, the class of all graphs with $\gamma^{\text{ID}}(G) = |V(G)| - 1$ is classified in Chapter 3, and this class is infinite. It is easy to check that only six of these graphs are line graphs. Thus we have the following corollary:

Corollary 5.26. *If G is an identifiable line graph with $G \notin \{P_3, P_4, C_4, P_4 \bowtie K_1, C_4 \bowtie K_1, \mathcal{L}(K_4)\}$, then we have $\gamma^{ID}(G) \leq |V(G)| - 2$.*

Since $\gamma^{\text{EID}}(K_{2,n}) = 2n - 2$, $\gamma^{\text{ID}}(\mathcal{L}(K_{2,n})) = |V(\mathcal{L}(K_{2,n}))| - 2$, the bound of Corollary 5.26 is tight for an infinite family of graphs.

5.3.3.2 A minimal edge-identifying code induces a 2-degenerate graph

We recall that a graph on n vertices is *2-degenerate* if its vertices can be ordered $v_1, v_2, \dots, v_n$ such that each vertex v_i is joined to at most two vertices in $\{v_1, v_2, \dots, v_{i-1}\}$. Our main idea for proving upper bounds is to show that given an edge-identifiable graph G, any (inclusionwise) minimal edge-identifying code $\mathcal{C}_E$ induces a 2-degenerate subgraph of G and hence $|\mathcal{C}_E| \leq 2|V(G)| - 3$. Our proofs are constructive and one could build such small edge-identifying codes.

Theorem 5.27. *Let G be an edge-identifiable graph and let $\mathcal{C}_E$ be a minimal edge-identifying code of G. Then G', the subgraph induced by $\mathcal{C}_E$, is 2-degenerate, and hence $|\mathcal{C}_E| \leq 2|V(G)| - 3$.*

Proof. Let uv be an edge of G' with $d_{G'}(u), d_{G'}(v) \geq 3$. By minimality of $\mathcal{C}_E$ the subset $\mathcal{C}' = \mathcal{C}_E - uv$ of $E(G)$ is not an edge-identifying code of G. By the choice of uv, $\mathcal{C}'$ is still an edge-dominating set, thus there must be two edges, e_1 and e_2, that are not separated by $\mathcal{C}'$. Hence one of them, say e_1, is incident either to u or to v (possibly to both) and the other one (e_2) is incident to neither one.

We consider two cases: either $e_1 = uv$ or e_1 is incident to only one of u and v. In the first case, e_2 is adjacent to every edge of $\mathcal{C}'$ which uv is adjacent to. Since for each vertex of uv there are at least two edges in $\mathcal{C}'$ incident to this vertex, the subgraph induced by u, v and the vertices of e_2 must be isomorphic to K_4 and there should be no other edge of $\mathcal{C}'$ incident to any vertex of this K_4 (see Figure 5.9(a)).

In the other case, suppose e_1 is adjacent to uv at u. Let x and y be two neighbours of u in G' other than v. Then it follows that $e_2 = xy$ and, therefore, $d_{G'}(u) = 3$. Let z be the other end of e_1. We consider two sub-cases: either $z \notin \{x, y\}$, or, without loss of generality, $z = x$. Suppose $z \notin \{x, y\}$. Recall that uv is the only edge separating e_1 and e_2, but e_1 must be separated from ux. Thus $zy \in \mathcal{C}_E$. Similarly, e_1 must be separated from uy, so $zx \in \mathcal{C}_E$. Furthermore, $d_{G'}(x) = d_{G'}(y) = d_{G'}(z) = 2$ and $\{x, y, z, u\}$ induces a C_4 in G' (see Figure 5.9(b)). Now suppose $e_1 = ux$, since uv is the only edge separating e_1 and e_2, then uy and possibly xy are the only edges in G' incident to y, so $d_{G'}(y) \leq 2$ and $d_{G'}(u) = 3$ (see Figures 5.9(c) and 5.9(d)).

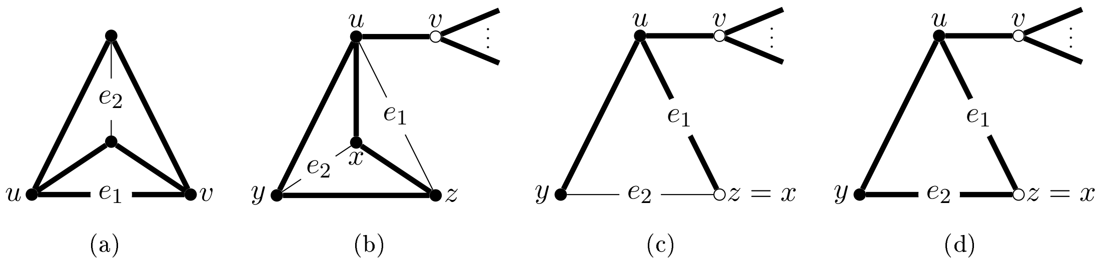


Figure 5.9: Case distinctions in the proof of Theorem 5.27. Black vertices have fixed degree in G'. Thick edges belong to $\mathcal{C}_E$.

To summarize, we proved that given an edge uv, in a minimal edge-identifying code $\mathcal{C}_E$, we have one of the following cases.

- One of u or v is of degree at most 2 in G'.
- Edge uv is an edge of a connected component of G' isomorphic to K_4^- (that is K_4 with an edge removed), see Figure 5.9(a).
- $d_{G'}(u) = 3$ (considering the symmetry between u and v) in which case either u is incident to a C_4 whose other vertices are of degree 2 in G' (Figure 5.9(b)), or to a vertex of degree 1 in G' (Figure 5.9(c)) or to a triangle with one vertex y of degree 2 in G' and y is not adjacent to v (Figure 5.9(d)).

In either case, there exists a vertex x of degree at most 2 in G' such that when x is removed, at least one of the vertices u, v has degree at most 2 in the remaining subgraph of G'. In this way we can define an order of elimination of the vertices of G' showing that G' is 2-degenerate. ☆

By further analysis of our proof we can in fact prove the following corollary, whose proof is given in Appendix A.7.

Corollary 5.28. *If G is an edge-identifiable graph on n vertices not isomorphic to K_4 or K_4^-, then $\gamma^{EID}(G) \leq 2n - 5$.*

We note that $\gamma^{\text{EID}}(K_{2,n}) = 2n - 2 = 2|V(K_{2,n})| - 6$ thus this bound cannot be improved much.

5.3.3.3 An application to Conjecture 4.4 in line graphs

Theorem 5.27 states that an edge-identifying code induces a sparse graph. This implies that if an edge-identifiable graph is dense enough, Conjecture 4.4 holds for its line graph:

Corollary 5.29. *If G is an edge-identifiable graph on n vertices and with average degree $\bar{d}(G) \geq 5$, then we have $\gamma^{ID}(\mathcal{L}(G)) \leq n - \frac{n}{\Delta(\mathcal{L}(G))}$.*

Proof. Let u be a vertex of degree $d(u) \geq \bar{d}(G) \geq 5$. Since G is edge-identifiable there is at least one neighbour v of u that is of degree at least 2. Thus there is an edge uv in G with $d(u) + d(v) \geq \bar{d}(G) + 2$ and, therefore, $\Delta(\mathcal{L}(G)) \geq \bar{d}(G)$. Hence, considering Corollary 5.28, it is enough to show that $2|V(G)| - 5 \leq |E(G)| - \frac{|E(G)|}{\bar{d}(G)}$.

To this end, since $\bar{d}(G) \geq 5$, we have $4|V(G)| \leq (\bar{d}(G) - 1)|V(G)|$, therefore,

$$4|V(G)| - 10 \leq (\bar{d}(G) - 1)|V(G)|.$$

Multiplying both sides by $\frac{\bar{d}(G)}{2}$, we have:

$$(2|V(G)|-5)\bar{d}(G) \leq (\bar{d}(G)-1)\frac{\bar{d}(G)}{2}|V(G)| = (\bar{d}(G)-1)|E(G)|. \qquad \star$$

5.4 Conclusion

In this chapter, we have turned our attention to graph classes for which no previous result regarding identifying codes were known. We have shown that interesting and nontrivial lower and upper bounds on the identifying code number of members of these classes hold.

Let us make a few observations regarding the results about graphs of given minimum degree and girth at least 5. It is known that a bound similar to the one of Theorem 5.3 holds for the domination number of a graph (of arbitrary girth) with given minimum degree:

Theorem 5.30 ([2, Theorem 1.2.2]). *Let G be a graph on n vertices with minimum degree δ. Then $\gamma(G) \leq \frac{n(1+\ln(\delta+1))}{\delta+1}$.*

The bounds of Theorem 5.3 are not relevant for small values of the minimum degree δ. Indeed, one bound includes an asymptotic term in δ (hence it is meaningful only for large values of δ), whereas for the other bound, $\gamma^{\mathrm{ID}}(G) \leq \frac{3\left(\ln\delta+\ln\ln\delta+1+\frac{\ln\ln\delta}{\ln\delta}+\frac{1}{\ln\delta}\right)}{2\delta}$, its value is strictly less than n only when $\delta \geq 7$. However, we recall that for large values of δ, the bounds are tight as discussed in the end of Subsection 5.1.3.

This raises some questions for small values of the minimum degree. We note the existence of the following related upper bounds from the literature on the domination number:

Theorem 5.31 ([108, Theorem 2.1]). *Let G be a graph on n vertices with minimum degree at least 1. Then $\gamma(G) \leq \frac{n}{2}$. This bound is tight for infinitely many graphs.*

Theorem 5.32 ([146]). *Let G be a connected graph on n vertices with minimum degree 2. If G does not belong to a set of seven exceptional graphs, then $\gamma(G) \leq \frac{2n}{5}$. This bound is tight for infinitely many graphs.*

Theorem 5.33 ([170]). *Let G be a graph on n vertices with minimum degree 3. Then $\gamma(G) \leq \frac{3n}{8}$. This bound is tight for infinitely many graphs.*

As discussed in Chapter 4.2, we cannot hope for similar bounds on parameter γ^{ID} without restrictions on the girth; however, they are to be related to our bound from Theorem 5.1, $\gamma^{\mathrm{ID}}(G) \leq \frac{7n}{8}$, which holds when G has minimum degree 2 and girth at least 5. We do not think that this bound is tight. The examples of graphs of girth 5 and minimum degree 2 having (up to our knowledge) largest identifying code number are the cycles C_n (which have identifying code roughly $\frac{n}{2}$, see Theorem 2.31). This leads to the following question:

Question 5.34. *What are tight bounds on $\gamma^{ID}(G)$ for graphs G of given (small) minimum degree $\delta \geq 2$ and girth at least 5?*

Moreover, we have not investigated *lower bounds* on parameter γ^{ID} for graphs of girth at least 5 and given minimum degree. This question deserves attention.

Question 5.35. *What are (tight) lower bounds on $\gamma^{ID}(G)$ for graphs G of given minimum degree and girth at least 5?*

We showed that a lower bound of the form $\gamma^{\mathrm{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ holds when G belongs either to the class of interval graphs (see Theorem 5.8) or to the one of line graphs (Corollary 5.20). Can these results be unified or generalized?

Question 5.36. *Are there superclasses of line graphs and/or interval graphs for which any of its members G satisfies $\gamma^{ID}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$?*

In particular, for the case of line graphs, L. W. Beineke characterized line graphs in terms of a list $\mathcal{B}$ of nine forbidden induced subgraphs [17] (see Theorem 2.2). It is natural to ask for which minimal list of forbidden induced subgraphs a similar bound would still hold:

Question 5.37. *For which minimal subsets $S_{\mathcal{B}}$ of $\mathcal{B}$ does a bound of the form $\gamma^{ID}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ hold for any $S_{\mathcal{B}}$-free graph G?*

However, we remark that a bound of the form $\gamma^{\text{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ does not hold for the class of co-bipartite graphs (i.e. complements of bipartite graphs). Examples can be built as follows: let A be a set of size k and let B be the set of nonempty subsets of A. Let G be the graph built on $A \cup B$, where A and B each induce a clique, and a vertex a of A is joined to a vertex b of B if $a \in b$. This graph is claw-free and it is easy to find an identifying code of size at most $2k = \Theta\left(\ln(|V(G)|)\right)$ in G (take the set A together with all vertices of B corresponding to a singleton of A).

This construction is interesting since the class of co-bipartite graphs forms a subclass of quasi-line graphs, therefore an upper bound of the form $\Omega\left(\sqrt{|V(G)|}\right)$ cannot hold for quasi-line graphs:

Proposition 5.38. *There is an infinite family of identifiable co-bipartite graphs (and therefore quasi-line graph and induced claw-free graphs) such that each graph G of this family has $\gamma^{ID}(G) = \Theta\left(\ln|V(G)|\right)$.*

To be even more precise, one can check that five graphs in Beineke's list are not co-bipartite: $B_1 = K_{1,3}$, B_4, B_5, B_7, and $B_9 = W_5$ from Figure 2.15. Therefore, since co-bipartiteness is closed by taking induced subgraphs, any subset of these five graphs cannot be an answer to Question 5.37:

Proposition 5.39. *Let $S_{\mathcal{B}}$ be a subset of $\mathcal{B}$. If a bound of the form $\gamma^{ID}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ for each $S_{\mathcal{B}}$-free graph G holds, then $S_{\mathcal{B}}$ contains at least one of $B_2, B_3 = K_5^-, B_6, B_8 = P_6^2$.*

Furthermore, we note that similar questions can be studied for other classes of graphs, such as the ones indicated in Tables 1.4 and 1.5:

Question 5.40. *What are tight lower bounds on parameetr γ^{ID} for given graph classes such as permutation graphs, (un)directed path graphs, (outer)planar, series-parallel graphs, respectively?*

Finally, let us discuss Corollary 5.29, which shows that Conjecture 4.4 holds for line graphs of graphs with average degree at least 5 (i.e. line graphs of "dense enough" graphs). We note that Conjecture 4.4 is particularly interesting for the class of line graphs, as most known families of graphs that reach the bound proposed by the conjecture are indeed line graphs (see Observation 4.9). This result lets us raise the following question that can be investigated:

Question 5.41. *Does Conjecture 4.4 hold for line graphs of sparser graphs? For example, does it hold for line graphs of trees?*

Part II

Algorithmic aspects

Chapter 6

Graph classes for which Min Id Code is log-APX-complete

We now turn our attention to algorithmic properties of the identifying code problem. In this chapter, we prove that Min Id Code is log-APX-hard, even for several restricted classes of graphs. It was already known that Min Id Code is log-APX-hard (see Theorem 2.50 from [20, 140, 185] independently), but no particular restriction on the graph class was given. Previously known reductions were from Min Set Cover [20, 140] and from Min Dom Set [185], but these reductions are quite intricate. Our reductions all use similar ideas (except for the one for DSP graphs), having Min Discrim Code as a starting point. The proximity between Min Discrim Code and Min Id Code allows us to design easier reductions, and to further restrict the class of graphs for which Min Id Code is log-APX-hard.

We first define two generic constructions in Section 6.1 that will be used in several reductions from this chapter.

In Section 6.2, we give an AP-reduction from Min Discrim Code to Min Id Code for bipartite graphs, thereby proving that Min Id Code is log-APX-hard in this class (Corollary 6.5).

In Section 6.3, we give another AP-reduction from Min Discrim Code to Min Id Code for split graphs, which implies that this problem is log-APX-hard for this class as well (Corollary 6.8).

In Section 6.4, we give an easy AP-reduction from Min Id Code to Min Id Code itself in DSP graphs (recall that a graph is DSP if it has a shortest path whose vertices form a dominating set). This implies that Min Id Code is log-APX-hard for DSP graphs (Corollary 6.11). This is in contrast with the complexity of Min Dom Set, which (unlike for bipartite or split graphs) is polynomial-time solvable for DSP graphs [136].

Finally, we extend the result for DSP graphs to the class of co-bipartite graphs (which form a subclass of DSP graphs) by providing an AP-reduction from Min Discrim Code to Min Id Code restricted to this class in Section 6.5. This implies that Min Id Code is log-APX-hard for co-bipartite graphs as well (Corollary 6.14).

The results of this chapter are solely my own work; they are new and have not appeared anywhere else than in this thesis.

6.1 Some useful constructions

We now describe two constructions, which will be helpful in many reductions of Chapter 6 in order to make sure that the vertices of some set $\mathcal{A}$ are correctly identified using the vertices of another set $\mathcal{L}$.

Construction 6.1 (bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A},\mathcal{L})$). *Given two sets of vertices $\mathcal{A}$ and $\mathcal{L}$ with $|\mathcal{L}| \geq \lceil \log_2(|\mathcal{A}|+1) \rceil$, the bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A},\mathcal{L})$, denoted $\mathcal{LOG}(\mathcal{A},\mathcal{L})$, is the graph of vertex set $\mathcal{A} \cup \mathcal{L}$ and where each vertex of $\mathcal{A}$ has a distinct nonempty subset of $\mathcal{L}$ as its neighbourhood.*

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

Construction 6.2 (non-singleton bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A},\mathcal{L})$). *Given two sets of vertices $\mathcal{A}$ and $\mathcal{L}$ with $|\mathcal{A}| \leq 2^{|\mathcal{L}|} - |\mathcal{L}| - 1$,*[1] *the non-single bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A},\mathcal{L})$, denoted $\mathcal{LOG}^*(\mathcal{A},\mathcal{L})$, is the graph of vertex set $\mathcal{A} \cup \mathcal{L}$ and where each vertex of $\mathcal{A}$ has a distinct subset of $\mathcal{L}$ of size at least 2 as its neighbourhood.*

6.2 MIN ID CODE for bipartite graphs

Reduction 6.3 (MIN DISCRIM CODE $\to$ MIN ID CODE for bipartite graphs). *Given an instance $(\mathcal{I},\mathcal{A})$ of MIN DISCRIM CODE, we construct in polynomial time the following bipartite graph $G(\mathcal{I},\mathcal{A})$ on $|\mathcal{I}|+|\mathcal{A}|+9\lceil \log_2(|\mathcal{A}|+1)\rceil+3$ vertices, with vertex set:*

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

and edge set:

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

where $\mathcal{B}(\mathcal{I},\mathcal{A})$ denotes the bipartite incidence graph of $(\mathcal{I},\mathcal{A})$ and $E\left(\mathcal{LOG}(\mathcal{A},\mathcal{L})\right)$ denotes the bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A},\mathcal{L})$ (see Construction 6.1).

The construction is illustrated in Figure 6.1.

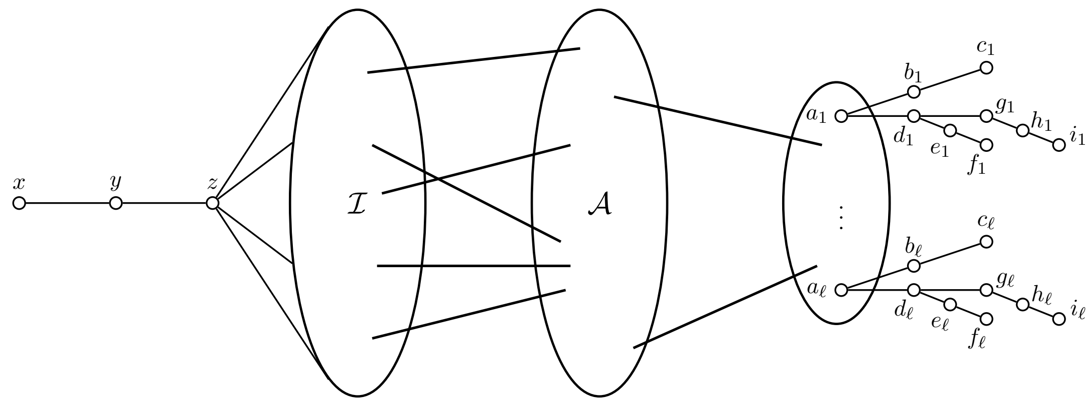


Figure 6.1: Reduction from MIN DISCRIM CODE to MIN ID CODE (with $\ell = \lceil \log_2(|\mathcal{A}|+1)\rceil$).

Theorem 6.4. *Let $(\mathcal{I},\mathcal{A})$ be an instance of MIN DISCRIM CODE, and $G(\mathcal{I},\mathcal{A})$, the bipartite graph constructed using Reduction 6.3. Then, $(\mathcal{I},\mathcal{A})$ has a discriminating code of size at most k if and only if $G(\mathcal{I},\mathcal{A})$ has an identifying code of size at most $k+6\lceil \log_2(|\mathcal{A}|+1)\rceil+2$, and one can construct one using the other in polynomial time.*

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

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

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

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 6\lceil \log_2(|\mathcal{A}| + 1) \rceil + 2$, and is clearly a dominating set. To see that it is an identifying code of $G(\mathcal{I}, \mathcal{A})$, observe that vertex z separates all vertices of $\mathcal{I}$ from all vertices which are not in $\mathcal{I} \cup \{z\}$. 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 $(\mathcal{I}, \mathcal{A})$, all vertices of $\mathcal{I}$ are dominated by a distinct subset of $\mathcal{D}$. Furthermore, due to the bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \{a_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}| + 1) \rceil\})$ (and since each vertex a_j belongs to the code), all vertices of $\mathcal{A}$ are dominated by a unique subset of $\{a_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}| + 1) \rceil\}$. 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 $G(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}| = k + 6\lceil \log_2(|\mathcal{A}| + 1) \rceil + 2$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $G(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}^*| \leq |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}| = 6\lceil \log_2(|\mathcal{A}| + 1) \rceil + 2 \tag{6.1}$$

$$|\mathcal{C}^* \cap \mathcal{I}| = \emptyset. \tag{6.2}$$

To get Condition (6.1), we replace $|\mathcal{C} \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}|$ by $\{x, z\} \cup \{a_j, c_j, d_j, f_j, g_j, i_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}| + 1) \rceil\}$ 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 $|\mathcal{C} \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}| \geq 6\lceil \log_2(|\mathcal{A}| + 1) \rceil + 2$. To see this, note that vertex z is forced by $\{x, y\}$, and $|\mathcal{C} \cap \{x, y\}| \geq 1$ since $\mathcal{C}$ must dominate x. Similarly, for any $j \in \{1, \ldots, \log_2(|\mathcal{A}| + 1)\}$, vertices a_j, d_j, g_j are forced by $\{b_j, c_j\}$, $\{e_j, f_j\}$ 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 (6.2), we replace each vertex $I \in \mathcal{I} \cap \mathcal{C}'$ by some vertex in $\mathcal{A}$. 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 $\mathcal{I}$ are dominated by z and all vertices of $\mathcal{A}$ are dominated by some vertex in $\{a_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}| + 1) \rceil\}$. Hence, I separates I itself from some other vertex I' in $\mathcal{I}$ (indeed, one can check that all other types of pairs which could be separated by I are actually already separated by some vertex of $\mathcal{C}' \cap (V(G(\mathcal{I}, \mathcal{A})) \setminus \mathcal{I})$. But then, the pair $\{I, I'\}$ is unique (suppose I separates I itself from two distinct vertices I' and I'' of $\mathcal{I}$, then I' and I'' would not be separated by $\mathcal{C}'$, a contradiction). Since $(\mathcal{I}, \mathcal{A})$ is identifiable, there must be some vertex A of $\mathcal{A}$ separating I from some I'. Hence we replace I by A. Doing this for every $I \in \mathcal{C}' \cap \mathcal{I}$, 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 (6.2), we have $|\mathcal{C}^* \cap \mathcal{A}| \leq |\mathcal{C}| - 6\lceil \log_2(|\mathcal{A}| + 1) \rceil + 2 = k$.

To finish the proof, we claim that $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I}, \mathcal{A})$. This is easy to observe, as all pairs $\{I, I'\}$ of $\mathcal{I}$ are separated by $\mathcal{C}^*$. By Condition (6.1), they must be separated by some vertex of $\mathcal{A}$ (note that z is adjacent to all vertices of $\mathcal{I}$). Hence $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I}, \mathcal{A})$. ☆

Theorem 6.4 proves that IDENTIFYING CODE restricted to bipartite graphs is NP-hard. In fact, Reduction 6.3 also preserves approximation ratios up to a constant factor, as shown by the following corollary.

Corollary 6.5. *Reduction 6.3 is an AP-reduction with parameter $\alpha = 8$ and* MIN ID CODE *restricted to bipartite graphs is log-APX-complete.*

Proof. We will use Theorem 6.4 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 DISCRIM CODE, and $7c \leq 1 + c(8-1)$; therefore, by Definition 2.8, we have an AP-reduction with $\alpha = 8$. Since MIN DISCRIM CODE is log-APX-complete [71] and by Theorem 2.50, MIN ID CODE is in log-APX, we get the claim.

Let $(\mathcal{I}, \mathcal{A})$ be an instance of MIN DISCRIM CODE with optimal value OPT, and let $G(\mathcal{I}, \mathcal{A})$ be the bipartite graph constructed using Reduction 6.3. By Theorem 6.4, we have:

$$\gamma^{\mathrm{ID}}(G(\mathcal{I}, \mathcal{A})) \leq OPT + 6\lceil \log_2(|\mathcal{A}| + 1)\rceil + 2. \tag{6.3}$$

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

$$|\mathcal{C}| \leq c\gamma^{\mathrm{ID}}(G(\mathcal{I}, \mathcal{A})). \tag{6.4}$$

By Theorem 6.4, we can compute in polynomial time a discriminating code $\mathcal{D}$ of $(\mathcal{I}, \mathcal{A})$. Using Inequalities 6.3 and 6.4 together with the fact that $\lceil \log_2(|\mathcal{A}|)\rceil \leq OPT \leq |\mathcal{D}|$ (Theorems 2.19 and 2.20), we get:[2]

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

☆

6.3 MIN ID CODE for split graphs

In this section, we use a reduction from MIN DISCRIM CODE to MIN ID CODE for split graphs similar to Reduction 6.3.

Reduction 6.6 (MIN DISCRIM CODE → MIN ID CODE for split graphs)**.** *Given an instance $(\mathcal{I}, \mathcal{A})$ of* MIN DISCRIM CODE*, we construct in polynomial time the following split graph $Sp(\mathcal{I}, \mathcal{A})$ on $|\mathcal{I}| + |\mathcal{A}| + 6\lceil \log_2(|\mathcal{A}| + 1)\rceil + 1$ vertices, with vertex set $V(Sp(\mathcal{I}, \mathcal{A})) = K \cup S$ (K is a clique and S, an independent set). More specifically:*

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

$Sp(\mathcal{I}, \mathcal{A})$ has edge set:

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

[2]For the last line inequality, we assume here that $OPT \geq 2$.

where $\mathcal{B}(\mathcal{I}, \mathcal{A})$ denotes the bipartite incidence graph of $(\mathcal{I}, \mathcal{A})$ and $E(\mathcal{LOG}^*(\mathcal{A}, \mathcal{L}))$ denotes the non-singleton bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \mathcal{L})$ (see Construction 6.2).

The construction is illustrated in Figure 6.2.

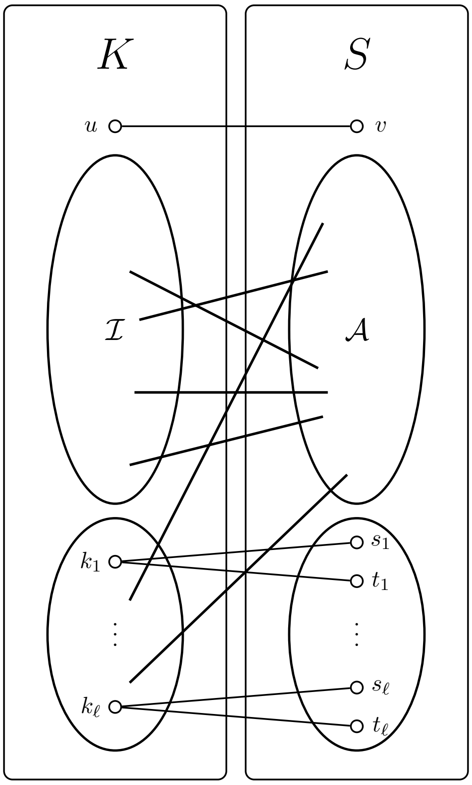


Figure 6.2: Reduction from MIN DISCRIM CODE to MIN ID CODE (with $\ell = 2\lceil \log_2(|\mathcal{A}|+1)\rceil$).

Theorem 6.7. *Let $(\mathcal{I}, \mathcal{A})$ be an instance of* MIN DISCRIM CODE, *and $Sp(\mathcal{I}, \mathcal{A})$, the split graph constructed using Reduction 6.6. Then, $(\mathcal{I}, \mathcal{A})$ has a discriminating code of size at most k if and only if $Sp(\mathcal{I}, \mathcal{A})$ has an identifying code of size at most $k + 4\lceil \log_2(|\mathcal{A}|+1)\rceil + 1$, and one can construct one using the other in polynomial time.*

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

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

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 2\lceil \log_2(|\mathcal{A}|+1)\rceil + 1$ and is a dominating set of $Sp(\mathcal{I}, \mathcal{A})$. To see that it is also an identifying code of $Sp(\mathcal{I}, \mathcal{A})$, 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 $(\mathcal{I}, \mathcal{A})$, 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 $\mathcal{A}$, each vertex of $\mathcal{A}$ is dominated by a distinct subset of vertices of $\{k_j \mid 1 \leq j \leq 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$ 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 $Sp(\mathcal{I}, \mathcal{A})$ with $|\mathcal{C}| = k + 4\lceil \log_2(|\mathcal{A}| + 1)\rceil + 1$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $Sp(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}^*| \leq |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap (V(Sp(\mathcal{I},\mathcal{A})) \setminus (\mathcal{I} \cup \mathcal{A}))| = 4\lceil \log_2(|\mathcal{A}|+1)\rceil + 1 \tag{6.5}$$
$$|\mathcal{C}^* \cap \mathcal{I}| = \emptyset. \tag{6.6}$$

To get Condition (6.5), we replace $|\mathcal{C} \cap \{V(Sp(\mathcal{I},\mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}|$ by $\{u\} \cup \{k_j, t_j \mid 1 \leq j \leq 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$ 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 $|\mathcal{C} \cap \{V(Sp(\mathcal{I},\mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}| \geq 4\lceil \log_2(|\mathcal{A}|+1)\rceil + 1$. To see this, note that for any $j \in \{1, \ldots, 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$, $|\mathcal{C} \cap \{k_j, s_j, t_j\}| \geq 2$. Indeed, s_j, t_j are false twins and must be separated by $\mathcal{C}$. Hence, 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 (6.6), we note that each vertex $I \in \mathcal{I} \cap \mathcal{C}'$ 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 $\mathcal{I}$ are dominated (e.g. by u) and all vertices of $\mathcal{A}$ are dominated by some vertex in $\{k_j \mid 1 \leq j \leq 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$. 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 $\{k_j \mid 1 \leq j \leq 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$), 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 $\mathcal{A}$ since no other vertex from S can be adjacent to I. But all pairs in $\mathcal{A}$ are separated by some vertex in $\{k_j \mid 1 \leq j \leq 2\lceil \log_2(|\mathcal{A}|+1)\rceil\}$, a contradiction. Removing every $I \in \mathcal{C}' \cap \mathcal{I}$ in this way, 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 (6.6), we have $|\mathcal{C}^* \cap \mathcal{A}| \leq |\mathcal{C}| - 4\lceil \log_2(|\mathcal{A}|+1)\rceil + 1 = k$.

To finish the proof, we claim that $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I},\mathcal{A})$. This is easy to observe, as all pairs $\{I, I'\}$ of $\mathcal{I}$ are dominated and separated by $\mathcal{C}^*$. By Condition (6.5), they must be separated by some vertex of $\mathcal{A}$. Hence $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I},\mathcal{A})$. ☆

Theorem 6.7 proves that IDENTIFYING CODE restricted to split graphs is NP-hard. In fact, Reduction 6.6 also preserves approximation ratios up to a constant factor, as shown by the following corollary.

Corollary 6.8. *Reduction 6.6 is an AP-reduction with parameter $\alpha = 6$ and* MIN ID CODE *restricted to split graphs is log-APX-complete.*

Proof. The proof is the same as the proof of Corollary 6.5, therefore we omit the details. We use Theorem 6.7 to show that any c-approximation algorithm for MIN ID CODE for split graphs can be transformed into a $5c$-approximation algorithm for MIN DISCRIM CODE, which is log-APX-complete (and since $5c \leq 1 + c(6-1)$, in Definition 2.8 we have $\alpha = 6$). Since by Theorem 2.50, MIN ID CODE is in log-APX, this proves the claims.

Given an instance $(\mathcal{I},\mathcal{A})$ of MIN DISCRIM CODE with optimal value OPT, let $Sp(\mathcal{I},\mathcal{A})$ be the split graph constructed using Reduction 6.6. By Theorem 6.7, we have:

$$\gamma^{\text{ID}}(Sp(\mathcal{I},\mathcal{A})) \leq OPT + 4\lceil \log_2(|\mathcal{A}|+1)\rceil + 1.$$

By using this fact and Theorem 6.7 in the same way as in the proof of Corollary 6.5, we obtain the theorem. ☆

6.4 MIN ID CODE for DSP graphs

In this section, we use a simple reduction from MIN ID CODE itself to MIN ID CODE for DSP graphs.

Reduction 6.9 (MIN ID CODE $\to$ MIN ID CODE for DSP graphs)**.** *Given an identifiable graph G on n vertices, we construct in polynomial time the graph G_{DSP} on $n+2$ vertices, with vertex set $V(G_{DSP}) = V(G) \cup \{u, v\}$ and edge set $E(G_{DSP}) = E(G) \cup \{u, x \mid x \neq u\}$. G_{DSP} is obviously a DSP graph, since it has a universal vertex, u.*

The construction is illustrated in Figure 6.3.

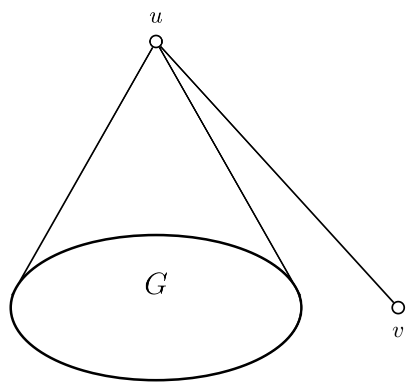


Figure 6.3: Reduction from MIN ID CODE to MIN ID CODE for DSP graphs.

Theorem 6.10. *Let G be an identifiable graph on n vertices and G_{DSP}, the DSP graph constructed using Reduction 6.9. Then, G has an identifying code of size at most k if and only if G_{DSP} has an identifying code of size at most $k+1$, and one can construct one using the other in polynomial time.*

Proof. **Sufficient side** ($\Rightarrow$) Let $\mathcal{C}$ be an identifying code of G. It is easy to check that $\mathcal{C} \cup \{v\}$ is an identifying code of G_{DSP}: all vertices within $V(G)$ are identified by $\mathcal{C}$ as they were in G; vertex v is dominated only by itself; vertex u is the only vertex dominated by the whole of $\mathcal{C} \cup \{v\}$.

Necessary side ($\Leftarrow$) Let $\mathcal{C}_{DSP}$ be an identifying code of G_{DSP}. Observe that $|\mathcal{C}_{DSP} \cap \{u, v\}| \geq 1$ since v must be dominated. Hence if $\mathcal{C}_{DSP} \setminus \{u, v\}$ is an identifying code of G, we are done. Let us assume the contrary. Note that necessarily $u \in \mathcal{C}_{DSP}$ since v does not dominate any vertex of $V(G_{DSP}) \setminus \{u, v\}$. But u is a universal vertex, hence u does not separate any pair of vertices of $V(G_{DSP}) \setminus \{u, v\}$. Therefore, $\mathcal{C}_{DSP} \setminus \{u\}$ is a separating code, but does not dominate some vertex $x \in V(G_{DSP}) \setminus \{u, v\}$: we have $N[x] \cap \mathcal{C}_{DSP} = \{u\}$. This implies that $v \in \mathcal{C}_{DSP}$ (otherwise x and v are not separated by $\mathcal{C}_{DSP}$). But then $(\mathcal{C}_{DSP} \setminus \{u, v\}) \cup \{x\}$ is an identifying code of G of size $|\mathcal{C}_{DSP}| - 1$. This completes the proof. ☆

Theorem 6.10 proves that IDENTIFYING CODE restricted to DSP graphs is NP-hard, but Reduction 6.9 also trivially preserves approximation ratios up to a constant factor, leading to the following corollary.

Corollary 6.11. MIN ID CODE *restricted to DSP graphs is log-APX-complete.*

6.5 MIN ID CODE for 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 of graphs (when assumed to be connected) is a subclass of DSP graphs since any pair of adjacent vertices belonging each to a distinct one among the two cliques, forms a dominating shortest path.

Reduction 6.12 (MIN DISCRIM CODE $\to$ MIN ID CODE for co-bipartite graphs)**.** *Given an instance $(\mathcal{I}, \mathcal{A})$ of* MIN DISCRIM CODE, *we construct in polynomial time the following co-bipartite graph $G(\mathcal{I}, \mathcal{A})$ on $|\mathcal{I}| + |\mathcal{A}| + 6\lceil \log_2(|\mathcal{A}| + 1) \rceil$ vertices, with vertex set $V(G(\mathcal{I}, \mathcal{A})) = K^1 \cup K^2$, where K^1 and K^2 are two cliques over the following sets of vertices:*

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

$G(\mathcal{I}, \mathcal{A})$ has edge set:

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

where $\mathcal{B}(\mathcal{I}, \mathcal{A})$ denotes the bipartite incidence graph of $(\mathcal{I}, \mathcal{A})$ and $E(\mathcal{LOG}(\mathcal{A}, \mathcal{L}))$ denotes the bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \mathcal{L})$ (see Construction 6.1).

The construction is illustrated in Figure 6.4.

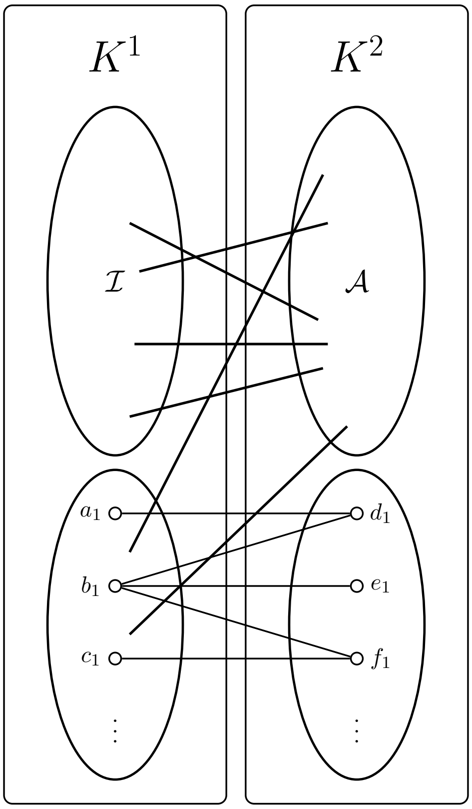


Figure 6.4: Reduction from Min Discrim Code to Min Id Code (with $\ell = \lceil \log_2(|\mathcal{A}| + 1) \rceil$).

Theorem 6.13. *Let $(\mathcal{I}, \mathcal{A})$ be an instance of* Min Discrim Code, *and $G(\mathcal{I}, \mathcal{A})$, the bipartite graph constructed using Reduction 6.12. Then, $(\mathcal{I}, \mathcal{A})$ has a discriminating code of size at most k if and only if $G(\mathcal{I}, \mathcal{A})$ has an identifying code of size at most $k + 5\lceil \log_2(|\mathcal{A}| + 1) \rceil - 2$, and one can construct one using the other in polynomial time.*

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

Sufficient side ($\Rightarrow$) Let $\mathcal{D} \subseteq \mathcal{A}$ be a discriminating code of $(\mathcal{I}, \mathcal{A})$, $|\mathcal{D}| = k$. 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:

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

One can easily check that $\mathcal{C}(\mathcal{D})$ has size $k + 5\lceil \log_2(|\mathcal{A}| + 1) \rceil - 2$ and is a dominating set of $G(\mathcal{I}, \mathcal{A})$. Let us show that it is also an identifying code of $G(\mathcal{I}, \mathcal{A})$. First of all, due to the bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \{a_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}|+1) \rceil\})$, each vertex of $\mathcal{A}$ is dominated by a distinct subset of vertices of $\{a_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}|+1) \rceil\}$; 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 $(\mathcal{I}, \mathcal{A})$. Within $V(G) \setminus (\mathcal{A} \cup \mathcal{I})$, 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 $\{a_k \mid 1 \leq k \leq \lceil \log_2(|\mathcal{A}|+1) \rceil\}$. Similarly, for $i \neq j$, 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 $i \neq j$, 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 $i \neq 1$, 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. Furtermore, 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 $G(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}| = k + 5\lceil \log_2(|\mathcal{A}|+1) \rceil - 2$. We first "normalize" $\mathcal{C}$ by constructing an identifying code $\mathcal{C}^*$ of $G(\mathcal{I}, \mathcal{A})$, $|\mathcal{C}^*| \leq |\mathcal{C}|$, such that the two following properties hold:

$$|\mathcal{C}^* \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}| = 5\lceil \log_2(|\mathcal{A}|+1) \rceil - 2 \tag{6.7}$$
$$|\mathcal{C}^* \cap \mathcal{I}| = \emptyset. \tag{6.8}$$

To get Condition (6.7), we first replace $|\mathcal{C} \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}|$ by $\{a_j, b_j, c_j, d_j, f_j \mid 1 \leq j \leq \lceil \log_2(|\mathcal{A}|+1) \rceil\} \setminus \{b_1, f_1\}$ 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 $|\mathcal{C}'| \leq |\mathcal{C}|$. Indeed, we had $|\mathcal{C} \cap \{V(G(\mathcal{I}, \mathcal{A})) \setminus \{\mathcal{I} \cup \mathcal{A}\}\}| \geq 5\lceil \log_2(|\mathcal{A}|+1) \rceil - 2$. To see this, note that for any $j \in \{1, \ldots, \log_2(|\mathcal{A}|+1)\}$, vertices a_j, c_j are forced by $\{d_j, e_j\}$ and $\{e_j, f_j\}$, respectively, and $|\mathcal{C} \cap \{d_j, e_j\}| \geq 1$ since $\mathcal{C}$ must separate b_j from c_j. Finally, consider the two sets $F = \{f_j \mid j \in \{1, \ldots, \log_2(|\mathcal{A}|+1)\}\}$ and $B = \{b_j \mid j \in \{1, \ldots, \log_2(|\mathcal{A}|+1)\}\}$. 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 $1 \leq i, j \leq \lceil \log_2(|\mathcal{A}|+1) \rceil$ (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 (6.8), we replace each vertex $I \in \mathcal{I} \cap \mathcal{C}'$ by some vertex in $\mathcal{A}$. 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 $K^1 \cup \mathcal{A}$ are dominated by a_1. Hence, I separates I itself from some other vertex I' in $\mathcal{I}$ (indeed, one can check that all other types of pairs which could be separated by I are actually already separated by some vertex of $\mathcal{C}' \cap (V(G(\mathcal{I}, \mathcal{A})) \setminus \mathcal{I})$. But then, the pair $\{I, I'\}$ is unique (suppose I separates I itself from two distinct vertices I' and I'' of $\mathcal{I}$, then I' and I'' would not be separated by $\mathcal{C}'$, a contradiction). Since $(\mathcal{I}, \mathcal{A})$ is identifiable, there must be some vertex A of $\mathcal{A}$ separating I from some I'. Hence we replace I by A. Doing this for every $I \in \mathcal{C}' \cap \mathcal{I}$, 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 (6.8), we have $|\mathcal{C}^* \cap \mathcal{A}| \leq |\mathcal{C}| - 5\lceil \log_2(|\mathcal{A}|+1) \rceil + 2 = k$. To complete the proof, we claim that $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I}, \mathcal{A})$. This is easy to observe, as all pairs $\{I, I'\}$ of $\mathcal{I}$ are separated by $\mathcal{C}^*$. By Condition (6.7), they must be separated by some vertex of $\mathcal{A}$. Hence $\mathcal{C}^* \cap \mathcal{A}$ is a discriminating code of $(\mathcal{I}, \mathcal{A})$. ☆

Theorem 6.13 proves that IDENTIFYING CODE restricted to co-bipartite graphs is NP-hard.

In fact, Reduction 6.12 also preserves approximation ratios up to a constant factor, as shown by the following corollary.

Corollary 6.14. *Reduction 6.12 is an AP-reduction with parameter $\alpha = 7$ and* MIN ID CODE *restricted to co-bipartite graphs (and therefore to quasi-line graphs and to AT-free graphs) is* log-APX*-complete.*

Proof. The proof is the same as the proofs of Corollaries 6.5 and 6.8, therefore we omit the details. We use Theorem 6.13 to show that any c-approximation algorithm for MIN ID CODE for co-bipartite graphs can be transformed into a $6c$-approximation algorithm for MIN DISCRIM CODE (and since $3c \leq 1 + c(4-1)$, in Definition 2.8 we have $\alpha = 7$), which is log-APX-complete. This proves the claims.

Given an instance $(\mathcal{I}, \mathcal{A})$ of MIN DISCRIM CODE with optimal value OPT, let $G(\mathcal{I}, \mathcal{A})$ be the co-bipartite graph constructed using Reduction 6.12. By Theorem 6.13, we have:

$$\gamma^{\text{ID}}(G(\mathcal{I}, \mathcal{A})) \leq OPT + 5\lceil \log_2(|\mathcal{A}| + 1)\rceil - 2.$$

By using this fact and Theorem 6.13 in the same way as in the proof of Corollary 6.5, we obtain the theorem. ☆

6.6 Conclusion

In this chapter, we gave new reductions proving that MIN ID CODE is log-APX-complete for bipartite graphs, split graphs, and co-bipartite graphs. These reductions are easier to understand than the previously known ones. The last case is particularly interesting since MIN DOM SET is easily solvable for co-bipartite graphs (any connected co-bipartite graph has a dominating set of size at most 2), and remains polynomial-time solvable even for the class of DSP graphs [136] (this class also includes asteroidal-triple free graphs).

We conclude by an observation regarding induced $K_{1,\ell}$-free graphs. It is known that MIN DOM SET is $(\ell - 1)$-approximable in induced $K_{1,\ell}$-free graphs [53].[3] It is easily observed that any co-bipartite graph is a quasi-line graph, since its whole vertex set can be partitioned into two cliques. Hence Corollary 6.14 shows that a similar result as the one for MIN DOM SET in induced $K_{1,\ell}$-free graphs does not hold for MIN ID CODE.

[3]Indeed, let I be any maximal independent set of an induced $K_{1,\ell}$-free graph G. We have $\frac{\alpha(G)}{\ell-1} \leq \gamma(G) \leq |I| \leq \alpha(G)$, where $\alpha(G)$ and $\gamma(G)$ denote the sizes of a maximum independent set and a minimum dominating set, respectively. The two last inequalities are straightforward. To see the first inequality, let us repeat the argument of [53]: consider an arbitrary independent set I and an arbitrary dominating set D of G, and let $Z = I \cap D$. Since any vertex of $I \setminus Z$ has a neighbour in D, but any vertex of D has at most $\ell - 1$ neighbours in I, we have $|I \setminus Z| \leq (\ell - 1)|D \setminus Z|$ and it follows that $|I| = |I \setminus Z| + |Z| \leq (\ell - 1)|D \setminus Z| + |Z| \leq (\ell - 1)|D|$. In particular, this holds for a maximum independent set and a minimum dominating set.

Chapter 7

Graph classes for which MIN ID CODE is APX-hard or IDENTIFYING CODE is NP-complete

IN this chapter, we prove that MIN ID CODE is APX-complete in several graph classes by constructing L-reductions to this problem. We also obtain some corollaries and one independent theorem about the decision problem IDENTIFYING CODE along the way.

In Section 7.1, we extend Theorem A.14 to bipartite graphs of maximum degree 4 by giving an L-reduction from MIN VERTEX COVER for subcubic graphs (Theorem 7.8). As a side result, we obtain that IDENTIFYING CODE is NP-complete for planar bipartite graphs of maximum degree 4 (Theorem 7.7). We then give another L-reduction from MIN DOM SET for subcubic bipartite graphs to MIN ID CODE for bipartite graphs of maximum degree 5. This reduction implies that IDENTIFYING CODE is NP-complete for chordal bipartite graphs (Theorem 7.10).

In Section 7.2, we define the natural class of *split graphs with bounded maximum CS-degree* and show that MIN ID CODE is APX-complete in this class (recall that by Corollary 6.8, it is log-APX-hard for the class of all split graphs). This result is proved using an L-reduction from MAX ($\leq 3, \leq 3$)-SAT (see Theorem 7.19).

We then investigate the class of line graphs in Section 7.3, proving in Corollary 7.21 that MIN EDGE-ID CODE (and therefore MIN ID CODE restricted to line graphs) is 4-approximable. This follows from combinatorial bounds that we proved in Section 5.3. We complement this result by showing that MIN EDGE-ID CODE is APX-hard. More precisely, we reduce MAX ($\leq 3, \leq 3$)-SAT to MIN EDGE-ID CODE for bipartite graphs of maximum degree 3 and arbitrarily large girth and prove that it is an L-reduction in Theorem 7.38. This implies that MIN EDGE-ID CODE is APX-complete in this class, and that MIN ID CODE is APX-complete when restricted to perfect line graphs of maximum degree 4. This reduction also implies that EDGE-IDENTIFYING CODE is NP-complete even when restricted to bipartite planar graphs of maximum degree 3 and arbitrarily large girth, and that IDENTIFYING CODE is NP-complete for perfect planar line graphs of maximum degree 4 (Theorem 7.36).

Finally, in Section 7.4, we prove that MIN ID CODE is NP-complete when restricted to interval graphs by a reduction from 3-DIMENSIONAL MATCHING (Theorem 7.50).

The result about split graphs (Section 7.2) is joint work with A. Kosowski, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov from [FKM+12]. The results about line graphs (Section 7.3) appeared in [FGN+12] (joint work with S. Gravier, R. Naserasr, A. Parreau and P. Valicov) as an NP-completeness reduction; we choose to present it as a more powerful L-reduction. Moreover, in [FGN+12], only the proof for the case of girth 8 appeared, mentioning that the same proof could be done for arbitrary girth; herein, we present such a general proof. The result of Section 7.4 is joint work with A. Kosowski, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov from [FKM+12]. The other results are solely the author's work; they are new and have not appeared elsewhere.

7.1 Min Id Code for bipartite graphs of small maximum degree and Identifying Code for planar bipartite graphs of maximum degree 4 and for chordal bipartite graphs

In this section, we improve Theorems 2.42 from [6], 2.43 from [9], Theorem 2.44 from [158] by showing that Identifying Code is NP-complete for planar bipartite graphs of maximum degree 4. We also improve Theorem 2.51 from [94] by showing that Min Id Code is APX-hard in bipartite graphs of maximum degree 4. The result of Theorem 2.51 from [94] is for non-bipartite graphs of maximum degree 8, and the authors asked whether it could be extended to bipartite graphs. Finally, we show that Identifying Code is NP-complete for chordal bipartite graphs. Note that the class of chordal bipartite graphs is quite interesting for the following reason: Dominating Set is NP-complete for this class [157], but the related problem Total Dominating Set is polynomial-time solvable [70].

7.1.1 A reduction from Min Vertex Cover

We first present a reduction from Min Vertex Cover to Min Id Code.

Reduction 7.1 (Min Vertex Cover → Min Id Code)**.** *Given a graph G, we construct the graph G' on vertex set*

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

and edge set

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

The construction is illustrated in Figure 7.1.

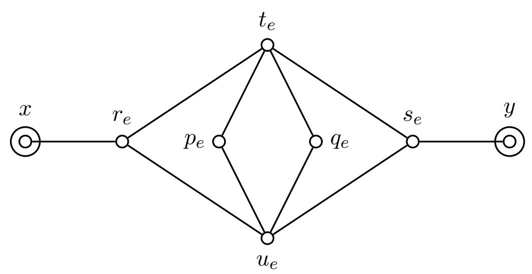


Figure 7.1: Reduction gadget for edge $e = \{x, y\}$ in Reduction 7.1 from Min Vertex Cover to Min Id Code. The original vertices of G, x and y, are circled.

For the following claims, let G be a graph and G', the graph obtained from G using Reduction 7.1.

Claim 7.2. *Let VC be a vertex cover of G. Using VC, one can build an identifying code of G' of size at most $|VC| + 3|E(G)|$.*

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

We can easily check that $\mathcal{C}$ is an identifying code of G': if an original vertex x of G belongs to VC, x is separated from every vertex that is non-adjacent to x by x itself, and from each of its neighbours (r_e or s_e for some edge e of G) by vertex t_e. If x does not belong to VC, all its neighbours in G belong to VC; hence all the neighbours of x in G' are some $s_e, s_{e'}, s_{e''}$, where e, e', e'' are the three edges incident to x in G. By the construction of $\mathcal{C}$, all three vertices belong to $\mathcal{C}$. Hence x is separated from every other vertex in G'. Finally, for each edge e of G, vertices $p_e, q_e, r_e, s_e, t_e, u_e$ are separated by all vertices of $V(G') \setminus \{p_e, q_e, r_e, s_e, t_e, u_e\}$ by either t_e or q_e; moreover it is easy to check that they are correctly separated from each other. ☆

Claim 7.3. *Let $\mathcal{C}$ be an identifying code of G'. For each $e \in E(G)$, we have:*

$$|\mathcal{C} \cap \{p_e, q_e, r_e, s_e, t_e, u_e\}| \geq 3.$$

Proof. Note that t_e, u_e are false twins, hence one of them (say t_e) belongs to $\mathcal{C}$. Similarly, p_e, q_e are false twins and one of them belongs to $\mathcal{C}$, say q_e. Now, t_e, q_e need to be separated, hence one of p_e, r_e, s_e, u_e belongs to $\mathcal{C}$. ☆

Claim 7.4. *Let $\mathcal{C}$ be an identifying code of G'. For each $e = \{x, y\} \in E(G)$, we have:*

$$|\mathcal{C} \cap \{x, y, p_e, q_e, r_e, s_e, t_e, u_e\}| \geq 4.$$

Proof. By contradiction, suppose $|\mathcal{C} \cap \{x, y, p_e, q_e, r_e, s_e, t_e, u_e\}| = 3$. By the same arguments as in the proof of Claim 7.3, we can assume $t_e, q_e \in \mathcal{C}$, and $|\mathcal{C} \cap \{p_e, r_e, s_e, u_e\}| = 1$. We derive a contradiction for each case. If p_e or u_e belong to $\mathcal{C}$, r_e, s_e are not separated. If $r_e \in \mathcal{C}$, p_e, s_e are not separated. If $s_e \in \mathcal{C}$, p_e, r_e are not separated. ☆

Claim 7.5. *Let $\mathcal{C}$ be an identifying code of G'. From $\mathcal{C}$, we can build an identifying code $\mathcal{C}'$ with $|\mathcal{C}'| \leq |\mathcal{C}|$ such that for each $e = \{x, y\} \in E(G)$, we have $|\mathcal{C}' \cap \{p_e, q_e, r_e, s_e, t_e, u_e\}| = 3$.*

Proof. Assume that $|\mathcal{C} \cap \{p_e, q_e, r_e, s_e, t_e, u_e\}| \geq 4$. By the same arguments as in the previous proofs, we may also assume that $t_e, q_e \in \mathcal{C}$. Now, if $p_e \in \mathcal{C}$, one can check that $\mathcal{C}' := \mathcal{C} \setminus \{p_e\}$ is still an identifying code, since p_e, q_e are false twins and $q_e \in \mathcal{C}'$. If $|\mathcal{C}' \cap \{p_e, q_e, r_e, s_e, t_e, u_e\}| = 3$, we are done. Otherwise, we have $r_e \in \mathcal{C}'$ or $s_e \in \mathcal{C}'$. In the former case, let $\mathcal{C}' := (\mathcal{C}' \setminus \{s_e, u_e\}) \cup \{y\}$; in the latter case, $\mathcal{C}' := (\mathcal{C}' \setminus \{r_e, u_e\}) \cup \{x\}$. One can check that $\mathcal{C}'$ is still an identifying code: indeed, assume that we had $r_e \in \mathcal{C}'$ (the other case follows by symmetry). Then we have r_e, t_e, q_e, y in the new code $\mathcal{C}'$. It is easy to see that vertices $x, p_e, q_e, r_e, s_e, t_e, u_e$ are separated from each other and remain separated from all other vertices. However, y might have been separated from one of its neighbours by s_e. But all neighbours of y are of the form $r_{e'}$ or $s_{e'}$ fro some edge e' incident to y in G, and hence they are separated from y by $t_{e'}$. ☆

Claim 7.6. *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 $|\mathcal{C}| - 3|E(G)|$.*

Proof. Use Claim 7.5 to build code $\mathcal{C}'$ such $|\mathcal{C}'| \leq |\mathcal{C}|$ and for each $e = \{x, y\} \in E(G)$, we have $|\mathcal{C}' \cap \{p_e, q_e, r_e, s_e, t_e, u_e\}| = 3$. By this property and Claim 7.4, we have $|\mathcal{C}' \cap \{x, y\}| \geq 1$. Hence $VC = \mathcal{C}' \setminus \{p_e, q_e, r_e, s_e, t_e, u_e \mid e \in E(G)\}$ is a vertex cover of G with $|VC| \leq |\mathcal{C}'| - 3|E(G)| \leq |\mathcal{C}| - 3|E(G)|$. ☆

These claims are enough to give a new proof that IDENTIFYING CODE is NP-complete:

Theorem 7.7. IDENTIFYING CODE *is NP-complete, even when restricted to planar bipartite graphs of maximum degree 4.*

Proof. We apply Reduction 7.1 to VERTEX COVER for planar subcubic graphs, which is known to be NP-complete [87]. Given a planar subcubic graph G, it is easy to check that G' is planar, has maximum degree at most 4 (due to vertices t_e, u_e in the edge gadget), and is bipartite, since the edge gadget for edge $e = \{x, y\}$ is bipartite, with x, y in the same part. Claim 7.2

applied on a optimal vertex cover shows that $\gamma^{\text{ID}}(G') \leq \tau(G) + 3|E(G)|$. Claim 7.6 applied on a optimal identifying code shows that $\tau(G) \leq \gamma^{\text{ID}}(G') - 3|E(G)|$. Hence we get that $\gamma^{\text{ID}}(G') = \tau(G) + 3|E(G)|$, completing the proof. ☆

In fact, we can show that Reduction 7.1 applied to VERTEX COVER restricted to subcubic graphs is an L-reduction.

Theorem 7.8. *Reduction 7.1 applied to graphs of maximum degree 3 is an L-reduction with parameters $\alpha = 10$ and $\beta = 1$. Therefore,* MIN ID CODE *is APX-complete, even for bipartite graphs of maximum degree at most 4.*

Proof. Let G be a graph of maximum degree 3 and G' the graph constructed from G using Reduction 7.1. We have to prove Properties 1 and 2 from Definition 2.4.

First of all, observe that by Claim 7.2, given an optimal vertex cover VC^* of G, we can construct an identifying code $\mathcal{C}$ with $\gamma^{\text{ID}}(G') \leq |\mathcal{C}| \leq |VC^*| + 3|E(G)| = \tau(G) + 3|E(G)|$. Similarly, by Claim 7.6, given an optimal identifying code $\mathcal{C}^*$ of G', we can construct a vertex cover VC of G such that $\tau(G) \leq |VC| \leq |\mathcal{C}^*| - 3|E(G)| = \gamma^{\text{ID}}(G) - 3|E(G)|$. Hence we have:

$$\gamma^{\text{ID}}(G') = \tau(G) + 3|E(G)|. \tag{7.1}$$

Property 1.
Since G has maximum degree 3, each vertex can cover at most three edges, hence we have $\tau(G) \geq \frac{|E(G)|}{3}$, so $|E(G)| \leq 3\tau(G)$. Using Equality (7.1), we get:

$$\gamma^{\text{ID}}(G') = \tau(G) + 3|E(G)| \leq 10\tau(G),$$

which proves Property 1 of Definition 2.4.

Property 2.
Let $\mathcal{C}$ be an identifying code of G'. Using Claim 7.6 applied to $\mathcal{C}$, we obtain a vertex cover VC with $|VC| \leq |\mathcal{C}| - 3|E(G)|$. By Equality (7.1), we have $-\tau(G) = 3|E(G)| - \gamma^{\text{ID}}(G')$. So we obtain:

$$\begin{aligned} |VC| - \tau(G) \leq& |\mathcal{C}| - 3|E(G)| + 3|E(G)| - \gamma^{\text{ID}}(G') \\ |\tau(G) - |VC|| \leq& |\gamma^{\text{ID}}(G') - |\mathcal{C}||, \end{aligned}$$

which proves Property 2 of Definition 2.4.

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 [53]. By construction, the graphs built from subcubic graphs in Reduction 7.1 are bipartite and of maximum degree 4 (see the proof of Theorem 7.7 for the precise arguments). ☆

7.1.2 A reduction from MIN DOM SET

We now give another reduction, this times from MIN DOM SET. Since the proof of validity is very similar to the one of the previous subsection, we leave it to Appendix A.8.

Reduction 7.9 (MIN DOM SET $\to$ 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 7.2.

One can show that Reduction 7.9 is in fact an L-reduction, leading to the following theorem:

Theorem 7.10. IDENTIFYING CODE *is NP-complete, even when restricted to chordal bipartite graphs.*

The proof can be found in Appendix A.8.

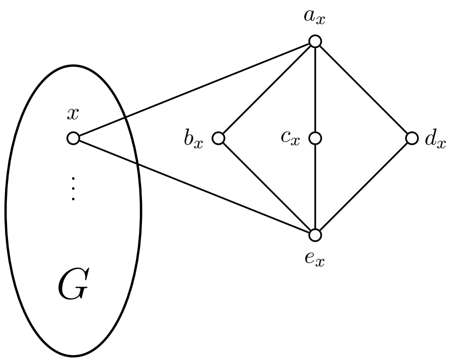


Figure 7.2: Reduction from MIN DOM SET to MIN ID CODE.

7.2 MIN ID CODE for split graphs of bounded maximum CS-degree

We saw in Chapter 6 that MIN ID CODE is log-APX-hard in split graphs. In this section, we show that when the split graphs are given a further restriction, one can approximate MIN ID CODE within a constant factor. However, we provide a reduction from MAX SAT to MIN ID CODE for split graphs to show that more efficient approximations are not tractable. The reduction of this section is similar (however simpler) than the one of the next section on line graphs (Section 7.3), and can be regarded as a "starter" for the latter.

The class of split graphs of given maximum degree Δ is, not a very rich one. Indeed, the vertex set of a split graph G can be partitioned into a clique K and an independent set S, but if G has maximum degree Δ, K must be of order at most $\Delta + 1$. Hence, we consider instead split graphs G for which the adjacencies of each vertex across the (K, S)-partition of G are in bounded number.

Definition 7.11. *Let G be a split graph. We say that G has* maximum CS-degree[1] Δ *if $V(G)$ can be partitioned into a clique K and an independent set S such that for each vertex $v \in V(G)$:*

- $v \in K \Rightarrow |N(v) \cap S| \leq \Delta$, *and*
- $v \in S \Rightarrow |N(v) \cap K| \leq \Delta$.

7.2.1 MIN ID CODE for split graphs of bounded maximum CS-degree is in APX

We showed in Corollary 6.8 that MIN ID CODE is log-APX-complete even when restricted to split graphs. The following theorem shows that this is not the case for split graphs of bounded maximum CS-degree.

Theorem 7.12. MIN ID CODE *is $O(\ln(\Delta))$-approximable when restricted to split graphs of maximum CS-degree at most Δ.*

Proof. Let G be a split graph and K, S the clique and the independent set forming a partition of $V(G)$ yielding the right bound on the maximum CS-degree of G.

First of all, we may assume that $|K|$ is unbounded, as otherwise, there would be only a bounded number of possible neighbourhoods within K for vertices of S. Then, if $V(G)$ is unbounded, many of the vertices in S would be false twins and we could easily solve MIN ID CODE exactly in polynomial time using the fact that among a set of mutually false twins, all but one necessarily belong to any identifying code (see Chapter 1.1.2).

Notice that the size of the symmetric difference between the closed neighbourhoods of two vertices of S is at most $2(\Delta + 1)$, and the size of the symmetric difference between the closed neighbourhoods of two vertices of K, at most 2Δ.

[1] This notation stands for maximum Clique-Stable-degree.

We reduce the problem of dominating all vertices of S and separating all pairs of vertices within S and within K to k-BOUNDED MIN SET COVER, which is $(1+\ln(k))$-approximable [127], as follows (the reduction is very similar to Reduction 2.23 from the Chapter 2). First of all, notice that for each pair u, v of vertices of $(K \times K) \cup (S \times S)$, the number of vertices that separates u, v is at most $2(\Delta + 1)$ for a pair in S, ans 2Δ for a pair in K. Similarly, each vertex of S is dominated by at most $\Delta + 1$ vertices. We define $(X, \mathcal{S})$ as an instance of k-BOUNDED MIN SET COVER, with X, the set of pairs of $(K \times K) \cup (S \times S)$ and all vertices of S, and where each set of $\mathcal{S}$ stands for a vertex of the graph and contains the pairs of $(K \times K) \cup (S \times S)$ that this vertex separates, as well as all the vertices of S that it dominates. By the previous remark, this is an instance of k-BOUNDED MIN SET COVER with $k \leq 2(\Delta + 2)$, and thus it is $O(\ln(\Delta))$-approximable.

Let $\mathcal{C}$ be a code obtained using the previous algorithm, and suppose that $\mathcal{C}$ is not an identifying code of G. Then, either some vertex of K is not dominated, or some pair u, v from $K \times S$ is not separated by $\mathcal{C}$. In the former case, picking an arbitrary vertex from K is enough to dominate all vertices of K. In the latter case, there is no code vertex within $K \setminus N(v)$, so pick some arbitrary vertex x from $K \setminus N(v)$ to separate u, v. Now, all pairs k, s from $K \times S$ that are still not separated must be such that $x \in N(s)$ (otherwise x would separate them); since x has at most Δ neighbours within S, there are at most $\Delta - 1$ vertices of S participating to such a pair. Let S' be the set of these vertices. We have $|N(S')| \leq \Delta^2$. Since $|K|$ is unbounded with respect to Δ, $K \neq N(S')$ and we can pick one additional vertex of $K \setminus N(S')$ in order to separate each unseparated pair k, s.

Hence we added at most two vertices from K to $\mathcal{C}$, and we get an identifying code of G. We do not lose the asymptotic approximation factor $O(\ln(\Delta))$ by just adding these two vertices. ☆

7.2.2 MIN ID CODE for split graphs of bounded maximum CS-degree is APX-hard

We now reduce MAX SAT to MIN ID CODE for split graphs. We will show that this reduction is an L-reduction.

Reduction 7.13 (MAX SAT $\to$ MIN ID CODE for split graphs)**.** *Given an instance $(X, \mathcal{Q})$ of* MAX SAT *consisting of a set $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ of clauses over a set $X = \{x_1, \ldots, x_n\}$ of boolean variables, we construct in polynomial time the split graph $Sp(X, \mathcal{Q})$ on $11|X| + 4|\mathcal{Q}|$ vertices, with vertex set $V(Sp(X, \mathcal{Q})) = K \cup S$ (K is a clique and S is an independent set). More specifically, we have:*

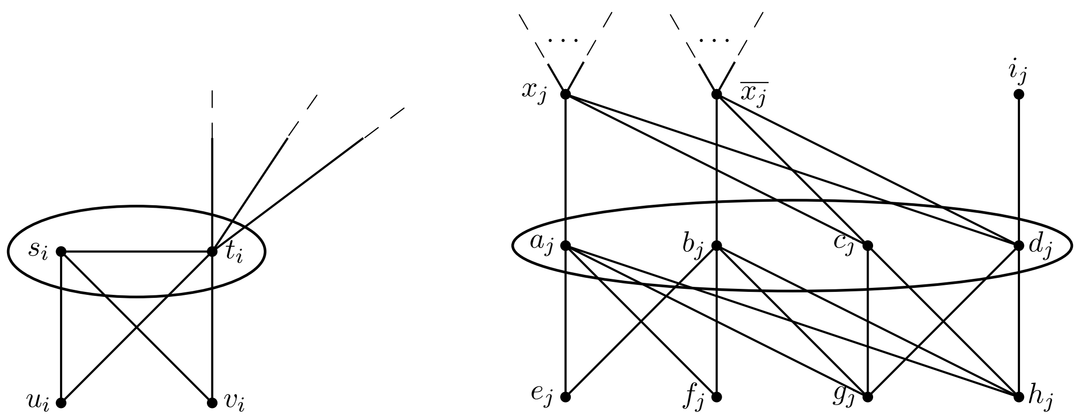
$$K = \{s_i, t_i \mid Q_i \in \mathcal{Q}\} \cup \{a_j, b_j, c_j, d_j \mid x_j \in X\}$$
$$S = \{u_i, v_i \mid Q_i \in \mathcal{Q}\} \cup \{x_j, \overline{x_j}, e_j, f_j, g_j, h_j, i_j \mid x_j \in X\}.$$

Moreover, $Sp(X, \mathcal{Q})$ has edge set:

$$\begin{aligned}
E\left(Sp(X, \mathcal{Q})\right) = \{&\{s_i, u_i\}, \{s_i, v_i\}, \{t_i, u_i\}, \{t_i, v_i\} \mid Q_i \in \mathcal{Q}\} \\
\cup \ \{&\{x_j, a_j\}, \{x_j, c_j\}, \{x_j, d_j\}, \{\overline{x_j}, b_j\}, \{\overline{x_j}, c_j\}, \{\overline{x_j}, d_j\}, \\
&\{a_j, e_j\}, \{a_j, f_j\}, \{a_j, g_j\}, \{a_j, h_j\}, \{b_j, e_j\}, \{b_j, f_j\}, \{b_j, g_j\}, \{b_j, h_j\}, \\
&\{c_j, g_j\}, \{c_j, h_j\}, \{d_j, g_j\}, \{d_j, h_j\}, \{d_j, i_j\} \mid x_j \in X\} \\
\cup \ \{&t_i, x_j \mid x_j \in Q_i, Q_i \in \mathcal{Q}, x_j \in X\} \\
\cup \ \{&t_i, \overline{x_j} \mid \overline{x_j} \in Q_i, Q_i \in \mathcal{Q}, x_j \in X\} \\
\cup \ \{&a, b \mid a, b \in K, a \neq b\}.
\end{aligned}$$

The construction is illustrated in Figure 7.3.

The intuition behind Reduction 7.13 is that for each variable $x_j \in X$, the choice made between vertices $x_j, \overline{x_j}$ represents the fact that x_j is "true" (vertex x_j is in the code) or "false" (vertex $\overline{x_j}$ is in the code). For each clause Q_i, vertex t_i is adjacent to the vertices corresponding to the literals of Q_i. Since s_i and t_i can only be separated by these vertices, this models the fact



(a) Clause gadget (b) Variable gadget

Figure 7.3: Reduction gadgets for clause Q_i and variable x_j. Vertices within the ellipses belong to the clique of the split graph.

that at least one of them has to be "true" (i.e. belongs to the code). The basic idea is inspired from the reduction in [60], where the decision problem Identifying Code was proved to be NP-complete for the first time.

We apply Reduction 7.13 to the problem Max (≤ 3,≤ 3)-SAT, where each clause has at most three literals, and each literal appears at most three times. Note that this restriction implies that the constructed split graphs have maximum CS-degree at most 5. We point out the fact that one may assume that each variable x_i appears at least once as a positive literal, and at least once as a negative literal ($\overline{x_i}$). Indeed, otherwise it is easy to satisfy the clauses containing x_i and one may remove these clauses and x_i to get a smaller equivalent instance.

Let $(X, \mathcal{Q})$ be an instance of Max (≤ 3,≤ 3)-SAT, let $s : X \to \{0, 1\}$ be a truth assignment of the variables of X, and let $\mathcal{C}$ be an identifying code of the graph $Sp(X, \mathcal{Q})$ defined in Reduction 7.13. We first prove some helpful claims.

Claim 7.14. *One can construct an identifying code $\mathcal{C}(s)$ of $Sp(X, \mathcal{Q})$ of size at most $6|X| + 3|\mathcal{Q}| - cost(s)$.*

Proof. Construct $\mathcal{C}(s)$ as follows: for each clause $Q_i \in \mathcal{Q}$, vertices s_i and u_i belong to $\mathcal{C}(s)$. For each variable $x_j \in X$, vertices c_j, f_j, h_j, i_j belong to $\mathcal{C}(s)$. Now, if variable x_j has value "true" in s, vertices a_j and x_j belong to $\mathcal{C}(s)$. Otherwise, vertices b_j and $\overline{x_j}$ belong to $\mathcal{C}(s)$. Finally, consider the set of $|\mathcal{Q}| - cost(s)$ clauses which are not satisfied by s. For each such clause Q_i, the corresponding pair of vertices s_i, t_i is not yet identified by $\mathcal{C}(s)$. It is sufficient to add a vertex corresponding to one of the literals of clause Q_i to $\mathcal{C}(s)$. In total, we have first considered six vertices per variable gadget and two vertices per clause gadget. In the last step, we have considered at most $|\mathcal{Q}| - cost(s)$ additional vertices, hence $|\mathcal{C}(s)| \leq 6|X| + 3|\mathcal{Q}| - cost(s)$.

We show that $\mathcal{C}(s)$ is a valid identifying code of $Sp(X, \mathcal{Q})$: first of all, it is easy to see that it is a dominating set. Furthermore, all vertices of K are dominated by all the vertices of $K \cap \mathcal{C}(s)$, hence (since we can assume that there are at least one clause and one variable) each vertex of K is separated from each vertex of S by this set of vertices. Hence we only need to check separation of pairs of vertices within S and within K. In fact, similarly, it is easy to see that any two vertices of S and any two vertices of K belonging to different clause or variable gadgets are separated by $\mathcal{C}(s)$. Now, in each clause gadget (for a clause Q_i), v_i is dominated only by s_i, and u_i is dominated by both s_i and itself, hence u_i, v_i are separated. Next, s_i, t_i are separated by some vertex representing a literal of Q_i (the construction ensures that at least one of them belongs to $\mathcal{C}(s)$). For each variable gadget of variable x_j, observe that each vertex among i_j, f_j, h_j is separated from each other variable of S in the gadget by itself. A similar fact holds for the vertex among $x_j, \overline{x_j}$ that belongs to $\mathcal{C}(s)$ Let us assume, without loss of generality, that $x_j \in \mathcal{C}(s)$ and $\overline{x_j} \notin \mathcal{C}(s)$; hence, $a_j \in \mathcal{C}(s)$ and $b_j \notin \mathcal{C}(s)$. Then e_j is dominated by a_j only; $\overline{x_j}$ is dominated only by c_j; g_j is dominated by both a_j, c_j. Finally, it remains to check that a_j, b_j, c_j, d_j are separated from each other. Both a_j, b_j are separated from both c_j, d_j by f_j; c_j, d_j are separated by i_j; a_j, b_j are separated by the vertex among $x_j, \overline{x_j}$ which belongs to $\mathcal{C}(s)$. ☆

The next two claims will help us to lower-bound the intersection between an identifying code of $Sp(X, \mathcal{Q})$ and given parts of the graph.

Claim 7.15. *We have* $|\mathcal{C} \cap (V(G) \setminus \bigcup_{x_j \in X}\{x_j, \overline{x_j}\})| \geq 5|X| + 2|\mathcal{Q}|$.

Proof. For each clause $Q_i \in \mathcal{Q}$, $\mathcal{C}$ contains either vertex u_i or vertex v_i since $\mathcal{C}$ separates u_i from v_i (u_i, v_i are false twins). Say $u_i \in \mathcal{C}$, then v_i must be dominated; hence, either v_i, s_i or t_i belongs to $\mathcal{C}$. Similarly, for each variable $x_j \in X$, we have $|\mathcal{C} \cap \{e_j, f_j\}| \geq 1$ and $|\mathcal{C} \cap \{g_j, h_j\}| \geq 1$ since e_j, f_j and g_j, h_j are false twins: say e_j and g_j belong to $\mathcal{C}$. But f_j and h_j must be dominated. If they are both dominated by the same vertex only (a_j or b_j) they are not separated, hence $|\mathcal{C} \cap \{a_j, b_j, c_j, d_j, f_j, h_j\}| \geq 2$. Finally, vertex i_j belongs to $\mathcal{C}$ since it is forced by c_j, d_j, completing the proof. ☆

Claim 7.16. *Let* $x_j \in X$. *We have* $|\mathcal{C} \cap \{x_j, \overline{x_j}\}| \geq 1$.

Proof. The claim follows from the fact that $N[a_j] \ominus N[b_j] = \{x_j, \overline{x_j}\}$. ☆

The next claim allows us to "normalize" a given identifying code of $Sp(X, \mathcal{Q})$.

Claim 7.17. *Using* $\mathcal{C}$, *one can construct an identifying code* $\mathcal{C}'$ *with* $|\mathcal{C}'| \leq |\mathcal{C}|$ *and* $|\mathcal{C} \cap (V(G) \setminus \bigcup_{x_j \in X}\{x_j, \overline{x_j}\})| = 5|X| + 2|\mathcal{Q}|$.

Proof. For each clause $Q_i \in \mathcal{Q}$, replace $\mathcal{C} \cap \{s_i, t_i, u_i, v_i\}$ by $\{s_i, u_i\}$. For each variable $x_j \in X$, replace $\mathcal{C} \cap \{c_j, d_j, e_j, f_j, g_j, h_j, i_j\}$ by $\{c_j, f_j, h_j, i_j\}$. Finally, if $x_j \in \mathcal{C}$, replace $\mathcal{C} \cap \{a_j, b_j\}$ by $\{a_j\}$. Otherwise, replace it by $\{b_j\}$. Using similar arguments as in the proof of validity of $\mathcal{C}(s)$ (last paragraph of the proof of Claim 7.14), one can easily check that the constructed code is still an identifying code. ☆

Claim 7.18. *Using* $\mathcal{C}$, *one can construct a truth assignment* $s = s(\mathcal{C})$ *of the variables of* X *such that* $cost(s) \geq |\mathcal{Q}| - (|\mathcal{C}| - 6|X| - 2|\mathcal{Q}|)$.

Proof. Let us first build code $\mathcal{C}'$ from $\mathcal{C}$ using Claim 7.17. By Claim 7.16 (the second inequality being trivial), we have:

$$|X| \leq |\mathcal{C}' \cap \bigcup_{x_j \in X} \{x_j, \overline{x_j}\}| \leq 2|X|. \tag{7.2}$$

We construct $s(\mathcal{C})$ as follows. For each variable $x_j \in X$, if $\mathcal{C}' \cap \{x_j, \overline{x_j}\} = \{x_j\}$, variable x_j is set to "true". If $\mathcal{C}' \cap \{x_j, \overline{x_j}\} = \{\overline{x_j}\}$, variable x_j is set to "false". Otherwise, $\mathcal{C}' \cap \{x_j, \overline{x_j}\} = \{x_j, \overline{x_j}\}$. We know that x_j appears at most three times as a literal and, as observed previously, it appears at least once in its negated form and at least once in its non-negated form. Hence, among $\{x_j, \overline{x_j}\}$ we choose the literal which appears most times in $\mathcal{Q}$, and set it to "true". Doing this, since the other literal appears at most once in $\mathcal{Q}$, at most one clause may remain unsatisfied. By Claim 7.17 and Inequality (7.2), there are exactly $|\mathcal{C}| - 6|X| - 2|\mathcal{Q}|$ such variables, yielding the claim. ☆

Claims 7.14 and 7.18 show that $(X, \mathcal{Q})$ is satisfiable if and only if $\gamma^{\text{ID}}(Sp(X, \mathcal{Q})) = 6|X| + 2|\mathcal{Q}|$, giving a proof that IDENTIFYING CODE is NP-hard even for split graphs of maximum CS-degree 5. However we are able to show the following stronger result:

Theorem 7.19. *Reduction 7.13 applied to the restricted version* MAX (≤ 3,≤ 3)-SAT *of* MAX SAT *is an L-reduction with parameters* $\alpha = 44$ *and* $\beta = 1$. *Therefore,* MIN ID CODE *is APX-complete when restricted to split graphs of maximum CS-degree 5.*

Proof. Let $(X, \mathcal{Q})$ be an instance of MAX (≤ 3,≤ 3)-SAT. We have to prove Properties 1 and 2 from Definition 2.4.

Property 1.
Since each variable appears in at most three clauses, we have:

$$|\mathcal{Q}| \leq 3|X|. \tag{7.3}$$

Consider the truth assignment s with all variables "true". Since each variable x_i appears at least once as a positive literal, at least one clause is satisfied thanks to variable x_i. Since each clause contains at most three literals, we get that $OPT(X, \mathcal{Q}) \geq cost(s) \geq \frac{|X|}{3}$, that is:

$$|X| \leq 3 \cdot OPT(X, \mathcal{Q}). \tag{7.4}$$

Using Inequalities (7.3) and (7.4) together with Claim 7.14, we obtain:

$$\begin{aligned}\gamma^{\text{ID}}(Sp(X, \mathcal{Q})) &\leq 6|X| + 3|\mathcal{Q}| - OPT(X, \mathcal{Q}) \\ &\leq 18 \cdot OPT(X, \mathcal{Q}) + 27 \cdot OPT(X, \mathcal{Q}) - OPT(X, \mathcal{Q}) \\ &= 44 \cdot OPT(X, \mathcal{Q})\end{aligned}$$

which proves Property 1 of Definition 2.4.

Property 2.
Let $\mathcal{C}$ be an identifying code of $Sp(X, \mathcal{Q})$ and $\mathcal{C}^*$ be a minimum identifying code of $Sp(X, \mathcal{Q})$, that is $|\mathcal{C}^*| = \gamma^{\text{ID}}(Sp(X, \mathcal{Q}))$. We consider the code $\mathcal{C}'$ built using $\mathcal{C}$ and Claim 7.17. We also assume that $|\mathcal{C}^* \cap (V(G) \setminus \bigcup_{x_j \in X}\{x_j, \overline{x_j}\})| = 5|X| + 2|\mathcal{Q}|$ using Claim 7.17.

Following Claim 7.16, for each variable $x_j \in X$, we have $1 \leq |\mathcal{C}' \cap \{x_j, \overline{x_j}\}| \leq 2$ and $1 \leq |\mathcal{C}^* \cap \{x_j, \overline{x_j}\}| \leq 2$. Hence $|\mathcal{C}' \cap \bigcup_{x_j \in X}\{x_j, \overline{x_j}\}| = (1+\gamma)|X|$ and $|\mathcal{C}^* \cap \bigcup_{x_j \in X}\{x_j, \overline{x_j}\}| = (1+\rho)|X|$ for some $\gamma, \rho \in [0, 1]$.

By Claim 7.15 and since $|\mathcal{C}' \cap (V(G) \setminus \bigcup_{x_j \in X}\{x_j, \overline{x_j}\})| = |\mathcal{C}^* \cap (V(G) \setminus \bigcup_{x_j \in X}\{x_j, \overline{x_j}\})| = 5|X| + 2|\mathcal{Q}|$, we have $\gamma \geq \rho$ and:

$$|\mathcal{C}'| - |\mathcal{C}^*| = (\gamma - \rho)|X| \tag{7.5}$$

Applying Claim 7.18 to $\mathcal{C}'$, which has size $6|X| + 2|\mathcal{Q}| + \gamma|X|$, we can construct the truth assignment $s(\mathcal{C}')$ of the variables of X such that:

$$cost(s(\mathcal{C}')) \geq |\mathcal{Q}| - \gamma|X|. \tag{7.6}$$

Furthermore, we claim that the following holds:

$$OPT(X, \mathcal{Q}) \leq |\mathcal{Q}| - \rho|X|. \tag{7.7}$$

Indeed, suppose not. Then, there would be a truth assignment s^* of the variables of X satisfying strictly more than $|\mathcal{Q}| - \rho|X|$ clauses. But then by Claim 7.14 there would be an identifying code of size at most $6|X| + 3|\mathcal{Q}| - cost(s^*) < 6|X| + 2|\mathcal{Q}| + \rho|X| = |\mathcal{C}^*|$, a contradiction since $\mathcal{C}^*$ is a minimum identifying code.

By combining Inequalities (7.6), (7.7) and Equality (7.5), we get:

$$OPT(X, \mathcal{Q}) - cost(s(\mathcal{C}')) \leq |\mathcal{Q}| - \rho|X| - (|\mathcal{Q}| - \gamma|X|) = |\mathcal{C}'| - |\mathcal{C}^*|,$$

which proves Property 2 of Definition 2.4.

Now, since MAX (≤ 3,≤ 3)-SAT is APX-hard [162], our reduction implies that MIN ID CODE for split graphs of maximum CS-degree 5 is APX-hard. Since by Theorem 7.12 it is also in APX, it is APX-complete. ☆

7.3 MIN ID CODE for line graphs

In this section, we investigate the computational complexity of MIN EDGE-ID CODE or, equivalently, MIN ID CODE for line graphs. For the formal definition and a study of edge-identifying codes, we refer to Section 5.3. We use some of the results of Chapter 5. This work was published as part of [FGN+12] under the point of vue of the decision problem IDENTIFYING CODE. Here, we extend it to the (non-)approximability of MIN ID CODE.

7.3.1 MIN ID CODE for line graphs is 4-approximable

We showed in Theorem 5.27 from Section 5 that for any edge-identifiable graph G, and any inclusionwise minimal edge-identifying code $\mathcal{C}_E$ of G, it holds:

$$\frac{|V(G)|}{2} \leq \gamma^{\text{EID}}(G) \leq |\mathcal{C}_E| \leq 2|V(G)| - 3.$$

This chain of inequalities naturally implies a nice algorithmic result. Indeed, note that one can construct an inclusionwise minimal edge-identifying code in polynomial time in a greedy fashion: start with $\mathcal{C}_E$ as the whole edge set, and for each edge e, check whether $\mathcal{C}_E \setminus \{e\}$ is an edge-identifying code. If yes, let $\mathcal{C}_E := \mathcal{C}_E \setminus \{e\}$ and continue until no such edge exists. We get the two following (equivalent) results:

Corollary 7.20. MIN EDGE-ID CODE *is 4-approximable.*

Corollary 7.21. MIN ID CODE *restricted to line graphs is 4-approximable.*

7.3.2 MIN ID CODE for line graphs is APX-hard

In the following, we reduce MAX (≤ 3,≤ 3)-SAT to MIN EDGE-ID CODE. We use this reduction to prove that EDGE-IDENTIFYING CODE is NP-complete, even when restricted to planar bipartite graphs of maximum degree 3 and arbitrarily large girth. We use the same reduction to show that MIN EDGE-ID CODE is APX-hard, even when restricted to bipartite graphs of maximum degree 3 and arbitrarily large girth. The basic ideas and the structure of the proof of the reduction of this section are the same as the ones of Section 7.2, which is also from MAX SAT. However, since the class of graphs is more restrictive, the proof is also longer and more intricate. Most of the technical proofs can be found in Appendix A.9.

We first need to define a generic sub-gadget (denoted P-gadget) that will be needed for the reduction.

Definition 7.22. *The P-*gadget *is the tree on vertex set $\{a, b, c, d, e, f\}$ and edge set $\{\{a, b\}, \{b, c\}, \{b, d\}, \{d, e\}, \{e, f\}\}$.*

In order to have more compact figures, we will use the representation of this gadget as drawn in Figure 7.4, where the P-gadget is illustrated. We will say that a P-gadget is *attached* at some vertex x in some graph G if x is identified with vertex a of the P-gadget as depicted in the figure. When speaking of a P-gadget as a subgraph of a graph G, we always mean that it forms an induced subgraph of G, that is, there are no other edges within the gadget than those of Definition 7.22. Moreover, vertex a is the only vertex of the P-gadget which may be joined by an edge to other vertices outside the gadget.

In order to describe our reduction from MAX (≤ 3,≤ 3)-SAT, we define the *clause gadget* and *variable gadget* that will be used in the reduction.

Definition 7.23. *Let $Q_i \subseteq \{l_{i_1}, l_{i_2}, l_{i_3}\}$ be a boolean clause over at most three literals and $\lambda \geq 1$, an integer. Let $V_{i_j} = \{l^1_{i_j}, \ldots, l^{2\lambda}_{i_j}\}$ and $E_{i_j} = \bigcup_{1 \leq k \leq 2\lambda-1} \{l^k_{i_j}, l^{k+1}_{i_j}\}$ for $1 \leq j \leq 3$.*

The clause gadget *$G(Q_i, \lambda)$ is the graph constructed from the vertices of the vertex set*

$$\{q_0, q_1, q_2, q_3\} \bigcup_{l_{i_j} \in Q_i} V_{i_j}$$

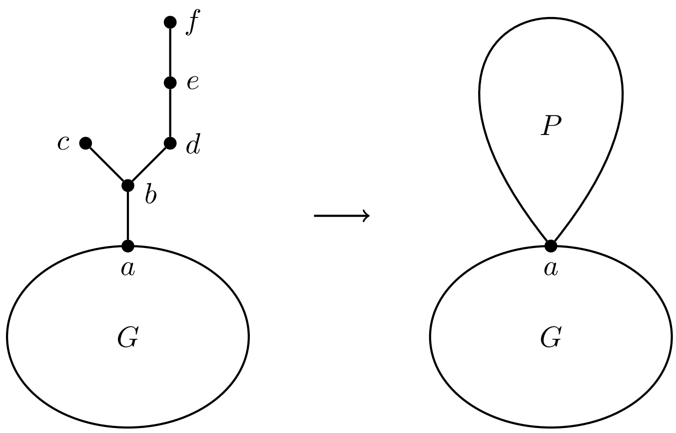


Figure 7.4: The generic P-gadget and its compact representation.

and edge set

$$\{\{q_0,q_1\},\{q_0,q_2\},\{q_0,q_3\}\}\bigcup_{l_{i_j}\in Q_i}\{q_{\min(j,2)},l_{i_j}^{2\lambda}\}\bigcup_{l_{i_j}\in Q_i}E_{i_j},$$

with the addition of the vertices and edges of $2\lambda|Q_i|+1$ *copies of the* P*-gadget attached at vertices* q_3 *and* $l_{i_j}^k$ *for all* $l_{i_j}\in Q_i$ *and* $1\le k\le 2\lambda$.

Definition 7.24. *Let* x_j *be a boolean variable and* $\mu\ge 2$*, an integer. We assume that* x_j *is used at most three times, once in its negative form* ($\overline{x_j}$)*, and once or twice in its positive form* (x_j)*. The* variable gadget $G(x_j,\mu)$ *is the graph constructed from the vertices of the vertex set*

$$\{x_j^1,\overline{x_j}^2,x_j^3\}\cup\bigcup_{1\le k\le 2\mu}\{a_k,b_k,c_k\}\cup\bigcup_{4\le k\le 2\mu}\{y_k,z_k\},$$

and edge set

$$\begin{aligned}&\{\{a_1,x_j^1\},\{a_2,\},\overline{x_j}^2,\{a_3,x_j^3\}\}\\ &\cup\bigcup_{1\le k\le 2\mu}\{\{a_k,b_k\},\{b_k,c_k\},\{b_k,a_{(k \bmod 2\mu)+1}\}\}\\ &\cup\bigcup_{4\le k\le 2\mu}\{\{a_k,y_k\},\{y_k,z_k\}\},\end{aligned}$$

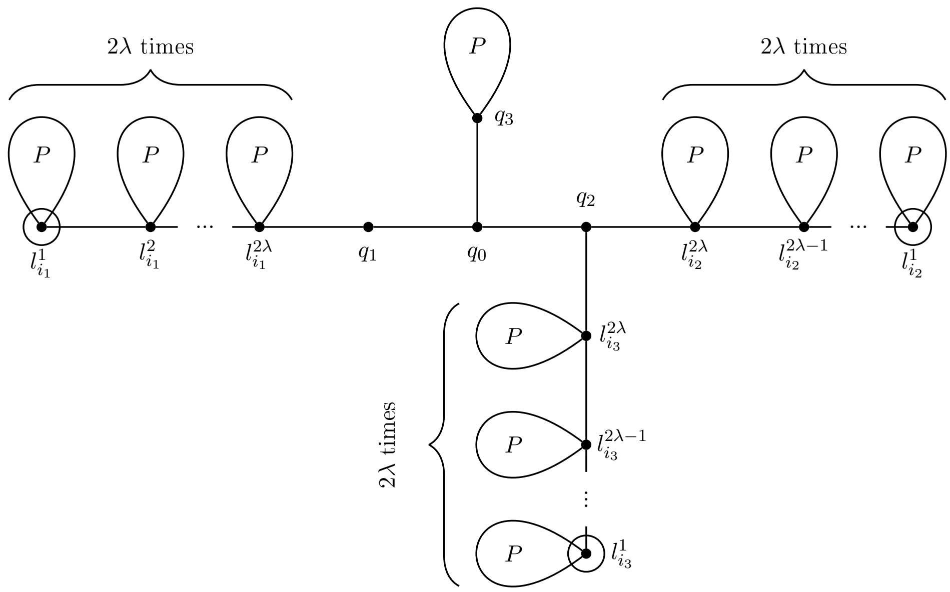
with the addition of the vertices and edges of $4\mu-3$ *copies of the* P*-gadget attached at vertices* c_k *for all* $1\le k\le 2\mu$ *and vertices* z_k *for all* $4\le k\le 2\mu$*. Moreover, if* x_j *is used only twice, we attach an additional* P*-gadget to vertex* x_j^3.

The clause gadget and variable gadget from Definitions 7.23 and 7.24 are illustrated in Figure 7.5.

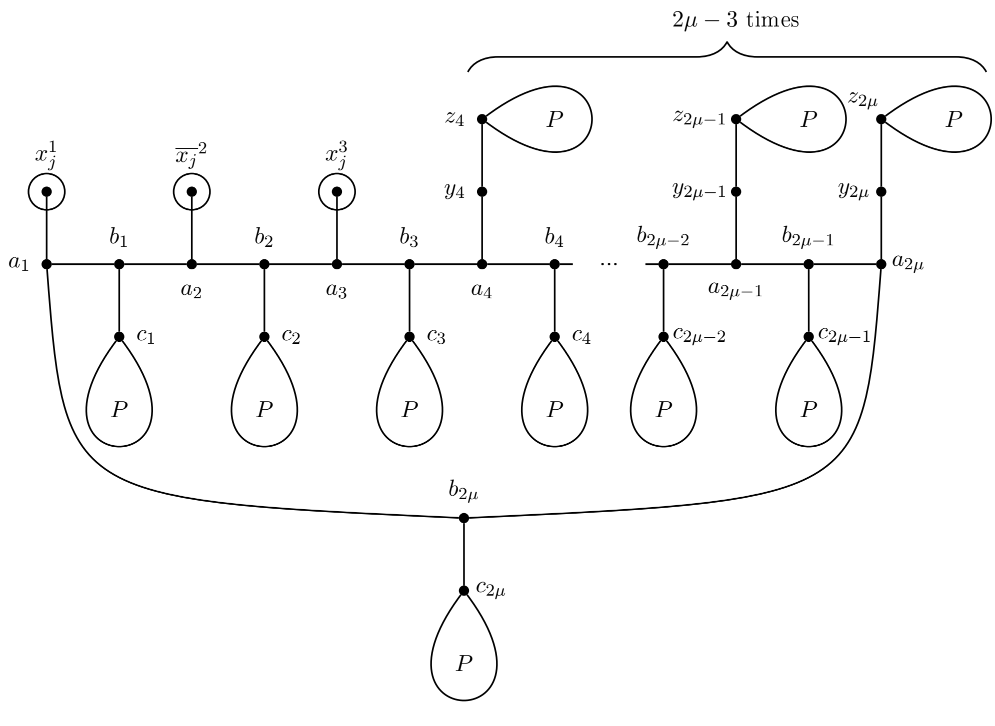
Reduction 7.25 (MAX ($\le 3,\le 3$)-SAT $\to$ MIN EDGE-ID CODE for bipartite graphs of maximum degree 3 and arbitrarily large girth)**.** *Let* $(X,\mathcal{Q})$ *be an instance of* MAX ($\le 3,\le 3$)-SAT *consisting of a set* $\mathcal{Q}=\{Q_1,\ldots,Q_m\}$ *of clauses over a set* $X=\{x_1,\ldots,x_n\}$ *of boolean variables and two integers* $\lambda\ge 1$ *and* $\mu\ge 2$*. We construct in polynomial time the graph* $G(X,\mathcal{Q},\lambda,\mu)$ *on at most* $(36\lambda+9)m+(30\mu-18)n$ *vertices, with vertex set*

$$\bigcup_{Q_i\in\mathcal{Q}}V(G(Q_i,\lambda))\cup\bigcup_{x_j\in X}V(G(x_j,\mu)).$$

For each $Q_i\in\mathcal{Q}$ *and* $x_j\in X$*, subgraphs* $G(Q_i,\lambda)$ *and* $G(x_j,\mu)$ *are copies of the* clause gadget *and* variable gadget *from Definitions 7.23 and 7.24.*



(a) Clause gadget $G_{Q_i}(\lambda)$



(b) Variable gadget $G_{x_j}(\mu)$

Figure 7.5: Reduction gadgets for clause Q_i and variable x_j.

In addition, for each clause $Q_i \subseteq \{l_{i_1}, l_{i_2}, l_{i_3}\}$ where $l_{i_k} \in \{x_{j(i_k)}, \overline{x_{j(i_k)}}\}$ and for each literal $l_{i_k} \in Q_i$ $(1 \leq k \leq 3)$ we connect $G(Q_i, \lambda)$ with $G(x_{i_k}, \mu)$ by identifying vertex $l^1_{i_k}$ from $G(Q_i, \lambda)$ with one of the vertices $x^1_{j(i_k)}, x^3_{j(i_k)}$ from $G(x_{j(i_k)}, \mu)$ if $l_{i_k} = x_{j(i_k)}$, and with vertex $\overline{x_{j(i_k)}}^2$ if $l_{i_k} = \overline{x_{j(i_k)}}$. We do this in such a way that for each vertex among $x^1_{j(i_k)}, x^3_{j(i_k)}$ from $G(x_{i_k}, \mu)$ is identified to at most one vertex of some clause gadget.[2]

[2]Note that this is possible due to the fact that in $(X, \mathcal{Q})$, which is an instance of MAX (≤ 3,≤ 3)-SAT, each variable appears once in its negated form and once or twice in its non-negated form.

The intuition behind Reduction 7.25 is that for each variable $x_j \in X$, the choice made between the sets of edges $\{\{a_1, x_j^1\}, \{a_3, x_j^3\}, \{a_5, y_5\}, \ldots, \{a_{2\mu-1}, y_{2\mu-1}\}\}$ and $\{\{a_2, \overline{x_j}^2\}, \{a_4, y_4\}, \ldots, \{a_{2\mu}, y_{2\mu}\}\}$ represents the fact that x_j is "true" (the first set is a subset of the code code) or "false" (the second set is a subset of the code). For each clause Q_i, edges $\{q_0, q_1\}$ and q_0, q_2 need to be separated by one of the edges of $\{q_1, l_{i_1}^{2\lambda}\}, \{q_2, l_{i_2}^{2\lambda}\}, \{q_2, l_{i_3}^{2\lambda}\}$. The choice made between these edges indicates which of the corresponding literals are set to "true". This choice is "transmitted" to the variable gadgets using the paths $l_{i_k}^{2\lambda}, \ldots, l_{i_k}^{1}$.

Let us now show some properties of the graphs constructed using Reduction 7.25.

Proposition 7.26. *Let $(X, \mathcal{Q})$ be an instance of* MAX $(\leq 3, \leq 3)$-SAT *and $G(X, \mathcal{Q}, \lambda, \mu)$, the graph constructed in Reduction 7.25. $G(X, \mathcal{Q}, \lambda, \mu)$ is bipartite, has maximum degree 3 and has girth* $\min\{4\mu, 8(\lambda + 1)\}$.

Proof. Note that $G(X, \mathcal{Q}, \lambda, \mu)$ has the same structure as the bipartite incidence graph $B(X, \mathcal{Q})$ of $(X, \mathcal{Q})$. One can easily check that no odd cycle is created in the construction. It is also easy to check from the construction that there is no vertex incident to four edges or more.

Finally, for the girth, observe that $G(x_j, \mu)$ has a unique cycle of length exactly 4μ. Since the girth of $B(X, \mathcal{Q})$ is at least 4, it follows that the minimum length of a cycle between some clause gadgets (at least two) and some variable gadgets (at least two) is at least $4(2\lambda + 1) + 2 + 2 = 8(\lambda + 1)$. ☆

We now prove a few claims on the gadgets and Reduction 7.25. First, consider a P-gadget P_G attached at some vertex a in some edge-identifiable graph G. We make the following claims.

Claim 7.27. *At least three edges of P_G belong to any edge-identifying code of G.*

Proof. This follows from the fact that $\{d, e\}$ is forced by $\{b, c\}$ and $\{c, d\}$. Similarly $\{c, d\}$ is forced by $\{d, e\}$ and $\{e, f\}$. Finally, in order to separate $\{c, d\}$ and $\{d, e\}$, one has to take at least one of $\{a, c\}$, $\{b, c\}$ or $\{e, f\}$. ☆

Claim 7.28. *Any edge-identifying code of G contains an edge of $G[V(G) \setminus V(P_G)]$ incident to vertex a.*

Proof. This follows from the fact that edge $\{a, c\}$ must be separated from edge $\{b, c\}$. ☆

The following claim helps one to "normalize" the intersection between a given edge-identifying code $E(P_G)$. The proof is in Appendix A.9.

Claim 7.29. *Let $\mathcal{C}_E$ be an edge-identifying code of G. One gets an identifying code $\mathcal{C}'_E$ with $|\mathcal{C}'_E| \leq |\mathcal{C}_E|$ by replacing $\mathcal{C}_E \cap E(P_G)$ by the three edges $\{\{b, c\}, \{b, d\}, \{d, e\}\}$.*

For the next claims, we let $(X, \mathcal{Q})$ be an instance of MAX $(\leq 3, \leq 3)$-SAT, and $s : X \to \{0, 1\}$, a truth assignment of the variables of X. Moreover, let $\mathcal{C}_E$ be an edge-identifying code of the graph $G(X, \mathcal{Q}, \lambda, \mu)$ defined in Reduction 7.25. Note that we make a simplifying assumption in order to simplify our proofs: assume that each clause of $\mathcal{C}$ contains exactly three literals, and that each variable of X appears exactly three times in $\mathcal{C}$ (once negated and twice unnegated). This assumption can actually not be made formally because such instances are solvable in polynomial time, see Theorem 2.13. However if not making it, we would have to count separately clauses containing two or three literals, and variables appearing twice or three times, but the whole proof would remain the same.

Claim 7.30. *One can construct an edge-identifying code $\mathcal{C}(s)$ of $G(X, \mathcal{Q}, \lambda, \mu)$ of size at most $(17\mu - 12)|X| + (21\lambda + 5)|\mathcal{Q}| - cost(s)$.*

Proof. Construct an edge-identifying code $\mathcal{C}(s)$ as follows. First of all, in each of the $(6\lambda + 1)|X| + (4\mu - 3)|\mathcal{Q}|$ P-gadgets, edges $\{b, c\}, \{c, d\}, \{a, c\}$ belong to $\mathcal{C}(s)$. For each clause $Q_i = \{l_{i_1, l_{i_2}, l_{i_3}}\} \in \mathcal{Q}$, edge $\{q_0, q_3\}$ belongs to $\mathcal{C}(s)$. For each literal l_{i_k}, $1 \leq k \leq 3$, if l_{i_k} is "true" in s, the λ edges $l_{i_k}^1, l_{i_k}^3, \ldots, l_{i_k}^{2\lambda-1}$ belong to $\mathcal{C}(s)$. Otherwise, the λ edges $l_{i_k}^2, l_{i_k}^4, \ldots, l_{i_k}^{2\lambda}$ belong to $\mathcal{C}(s)$. For each variable $x_j \in X$, in addition to the edges inside the P-gadget, all edges

$\{b_k, c_k\}$ $(1 \le k \le 2\mu)$ and $\{y_k, z_k\}$ $(4 \le k \le 2\mu)$ belong to $\mathcal{C}(s)$. If x_j is "true" in s, the μ edges $\{a_1, x_j^1\}, \{a_3, x_j^3\}, \{a_5, y_5\}, \dots, \{a_{2\mu-1}, y_{2\mu-1}\}$ belong to $\mathcal{C}(s)$. Otherwise, the μ edges $\{a_2, \overline{x_j}^2\}, \{a_4, y_4\}, \dots, \{a_{2\mu}, y_{2\mu}\}$ do. Finally, for each clause $Q_i = \{l_{i_1}, l_{i_2}, l_{i_3}\}$ among the $|\mathcal{Q}| - cost(s)$ unsatisfied clauses, we arbitrarily add one of the edges $\{q_1, l_{i_1}^{2\lambda}\}, \{q_2, l_{i_2}^{2\lambda}\}, \{q_2, l_{i_3}^{2\lambda}\}$.

The validity of the construction is proved in Claim A.16 of Appendix A.9. ☆

In the following, for each $x_j \in X$, let E_j denote the set of edges

$$\{\{a_1, x_j^1\}, \{a_2, \overline{x_j}^2\}, \{a_3, x_j^3\}, \{a_4, y_4\}, \dots, \{a_{2\mu}, y_{2\mu}\}\}$$

of $G(x_j, \mu)$. Moreover, we denote by E_j^+, the set of edges of "odd index" of E_j: $E_j^+ = \{\{a_1, x_j^1\}, \{a_3, x_j^3\}, \{a_5, y_5\}, \dots, \{a_{2\mu-1}, y_{2\mu-1}\}\}$. Similarly, we let $E_j^- = \{\{a_2, \overline{x_j}^2\}, \{a_4, y_4\}, \dots, \{a_{2\mu}, y_{2\mu}\}\}$. Finally, we define the set $A_j = \bigcup_{1 \le k \le 2\mu} \{\{a_k, b_k\}, \{b_k, a_{(k \bmod 2\mu)+1}\}\}$.

The proofs of the next four claims can be found in Appendix A.9.

The next two claims are used to lower-bound the intersection between $\mathcal{C}_E$ and specific parts of $G(x_j, \mu)$.

Claim 7.31. *We have* $|\mathcal{C}_E \cap (V(G(x_j, \mu)) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| \ge (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$.

Claim 7.32. *Let* $x_j \in X$. *We have* $|\mathcal{C}_E \cap (E_j \cup A_j)| \ge \mu$. *Moreover if* $|\mathcal{C}_E \cap (E_j \cup A_j)| = \mu$, *then either* $|\mathcal{C}_E \cap (E_j \cup A_j)| = E_j^+$, *or* $|\mathcal{C}_E \cap (E_j \cup A_j)| = E_j^-$.

The next two claims allow one to "normalize" $\mathcal{C}_E$.

Claim 7.33. *Using* $\mathcal{C}_E$, *one can construct an edge-identifying code* $\mathcal{C}'_E$ *with* $|\mathcal{C}'_E| \le |\mathcal{C}_E|$ *and such that for each variable* $x_j \in X$, $|\mathcal{C}_E \cap (E_j \cup A_j)| \le \mu + 1$.

Claim 7.34. *Using* $\mathcal{C}_E$, *one can construct an edge-identifying code* $\mathcal{C}'_E$ *with* $|\mathcal{C}'_E| \le |\mathcal{C}_E|$ *and* $|\mathcal{C}_E \cap (V(G) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| = (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$.

The following claim shows how to build a good truth assignment from edge-identifying code $\mathcal{C}_E$.

Claim 7.35. *Using* $\mathcal{C}_E$, *one can construct a truth assignment* $s = s(\mathcal{C}_E)$ *of the variables of* X *such that* $cost(s) \ge |\mathcal{Q}| - (|\mathcal{C}_E| - ((17\mu - 12)|X| - (21\lambda + 4)|\mathcal{Q}|))$.

Proof. Let us first build code $\mathcal{C}'_E$ from $\mathcal{C}_E$ using Claims 7.33 and 7.34. Note that by Claims 7.32 and 7.33, we have for each variable $x_j \in X$, $|\mathcal{C}'_E \cap (E_j \cup A_j)| \in \{\mu, \mu + 1\}$.

$$\mu|X| \le |\mathcal{C}'_E \cap \bigcup_{x_j \in X} (E_j \cup A_j)| \le (\mu + 1)|X|. \tag{7.8}$$

We construct $s(\mathcal{C}_E)$ as follows. For each variable $x_j \in X$, if $\mathcal{C}'_E \cap (E_j \cup A_j) = \mu$, then by Claim 7.32 either $\mathcal{C}'_E \cap (E_j \cup A_j) = E_j^+$ or $\mathcal{C}'_E \cap (E_j \cup A_j) = E_j^-$. In the former case, variable x_j is set to "true". In the latter case, it is set to "false". Otherwise $(\mathcal{C}'_E \cap (E_j \cup A_j) = \mu + 1)$, we set x_j to "true". Doing this, at most one clause may remain unsatisfied. By Claim 7.34 and Inequality (7.8), there are exactly $|\mathcal{C}_E| - ((17\mu - 12)|X| - (21\lambda + 4)|\mathcal{Q}|)$ such variables. Since all other clauses are necessarily satisfied due to the structure of $\mathcal{C}'_E$, we get the claim. ☆

We are now ready to use the previous claims to show how to apply Reduction 7.25. We first give a result on the decision problem Edge-Identifying Code:

Theorem 7.36. Edge-Identifying Code *is* NP*-complete even when restricted to bipartite planar graphs of maximum degree 3 and arbitrarily large girth.*

Proof. We apply Reduction 7.25 to instances $(X, \mathcal{Q})$ of the restricted version of $(\le 3, \le 3)$-SAT, Planar $(= 3, \le 3)$-SAT, where each variable appears exactly three times in the formula (once negated, twice non-negated), and the clause-variable bipartite incidence graph $B(X, \mathcal{Q})$ is planar (see Subsection 2.3.5 for a precise definition). Note that using the planarity of $B(X, \mathcal{Q})$, we can find a planar embedding of it in polynomial time [50, 122]. Using this embedding, it is easy to construct the graph $G(X, \mathcal{Q}, \lambda, \mu)$ (for arbitrarily large λ, μ) such that it is planar since $G(X, \mathcal{Q}, \lambda, \mu)$ has the same underlying structure as $B(X, \mathcal{Q})$.

Now, we can apply Claims 7.30 and 7.35 to an optimal identifying code and an optimal truth assignment to show that $(X, \mathcal{Q})$ is satisfiable if and only if $\gamma^{\text{ID}}(G(X, \mathcal{Q}, \lambda, \mu)) = (17\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$. ☆

Recall that a graph is *perfect* if and only if for each of its induced subgraphs H, the chromatic number of H equals the clique number of H. Recall that a line graph $\mathcal{L}(G)$ is perfect if and only if G has no odd cycles of length more than 3, (see Theorem 2.3) — this applies to $G(X, \mathcal{Q}, \lambda, \mu)$, which is bipartite. Moreover, one can check that the line graph of $G(X, \mathcal{Q}, \lambda, \mu)$ is planar and has maximum degree 4. Therefore, the following corollary follows:

Corollary 7.37. IDENTIFYING CODE *is NP-complete even when restricted to perfect planar line graphs of maximum degree 4.*

Using the claims, we are able to show that Reduction 7.25 is in fact an L-reduction (the proof is given in Appendix A.9).

Theorem 7.38. *For any $\lambda \geq 1$ and $\mu \geq 2$, Reduction 7.25 is an L-reduction with parameters $\alpha = 51\mu + 201\lambda + 8$ and $\beta = 1$. Hence* MIN EDGE-ID CODE *is APX-complete when restricted to bipartite graphs of maximum degree 3 and arbitrarily large girth, and* MIN ID CODE *is APX-complete when restricted to perfect line graphs of maximum degree 4.*

7.4 IDENTIFYING CODE for interval graphs is NP-complete

In this section, we show that IDENTIFYING CODE is NP-complete when restricted to interval graphs by reducing 3-DIMENSIONAL MATCHING to it. In what follows, when considering an interval graph, we will refer to its vertices as intervals (from the corresponding intersection model). In order to describe the reduction, we first define the following gadget:

Definition 7.39. *In an interval graph G, a P_5-gadget P is a set $V(P)$ of five intervals a_P, b_P, c_P, d_P, e_P whose intersection subgraph induces a P_5, and such that each interval of $V(G) \setminus V(P)$ either contains all intervals of $V(P)$, or does not intersect with any of them.*

A P_5-gadget is represented in Figure 7.6, where we also give a compact graphic representation that will be used in later figures.

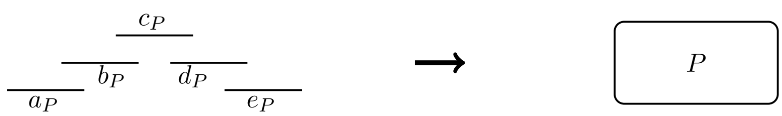


Figure 7.6: A P_5-gadget P and its compact representation.

We now define another useful gadget:

Definition 7.40. *Given a set S of two or three pairs of adjacent intervals in an interval graph G, we define a* transmitter gadget *$Tr(S)$ as follows. $V(Tr(S)) = \{u, uv^1, uv^2, v, vw^1, vw^2, w\} \cup V(P(u)) \cup V(P(uv)) \cup V(P(v)) \cup V(P(vw)) \cup V(P(w))$, where $P(u), P(uv), P(v), P(vw), P(w)$ are P_5-gadgets, and we have the following properties:*

- *Intervals $\{u, uv^1, uv^2, v, vw^1, vw^2, w\}$ appear in this order and induce a path;*
- *Gadgets $P(u), P(v), P(w)$ are included in u, v, w respectively, and no interval of $\{u, uv^1, uv^2, v, vw^1, vw^2, w\}$ other than u (v, w, respectively) intersects $V(P(u))$ ($V(P(v))$, $V(P(w))$, respectively);*
- *Similarly, $P(uv)$ and $P(vw)$ are included in the intersection of uv^1, uv^2 and vw^1, vw^2, respectively, and no other interval of $V(Tr(S))$ intersects with a vertex of $P(uv)$, $P(vw)$, respectively;*

- *All pairs of S are separated by a unique interval among u, w, and no other interval of $Tr(S)$ separates any pair of S;*

- *At least one pair of S is separated by u, and at least one, by w;*

- *The pair $\{uv^1, uv^2\}$ can only be separated by either u or v, and the pair $\{vw^1, vw^2\}$ can only be separated by either v or w (i.e. no other interval of G separates these pairs).*

For an illustration and a succinct graphical representation of a transmitter gadget, see Figure 7.7.

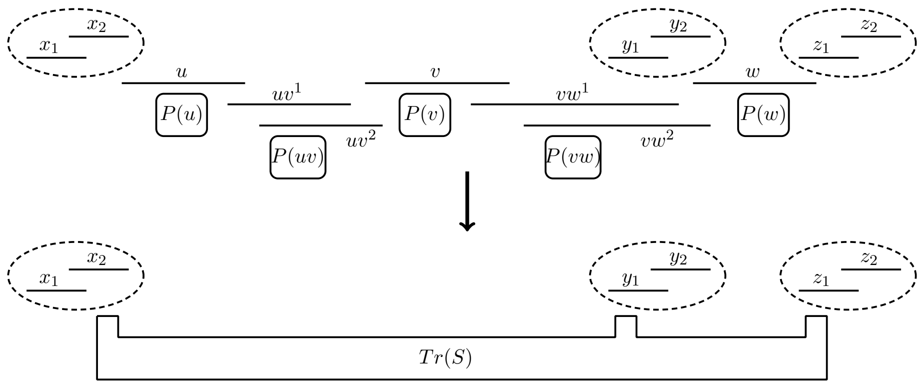


Figure 7.7: A transmitter gadget $Tr(S)$ and its representation, with $S = \{\{x_1, x_2\}, \{y_1, y_2\}, \{z_1, z_2\}\}$

We can now define gadgets modelling each element of $A \cup B \cup C$, and each triple of $\mathcal{T}$ from some instance of 3-DIMENSIONAL MATCHING.

Definition 7.41. *Let x be an element of $A \cup B \cup C$. The* element gadget *$G_{el}(x)$ is defined as follows: $V(G_{el}(x)) = \{f_x, g_x\} \cup V(P(x))$, where $P(x)$ is a P_5-gadget. Intervals f_x, g_x intersect each other, and $P(x)$ is included in their intersection.*

An element gadget is depicted in Figure 7.8.

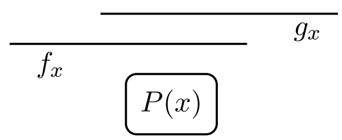


Figure 7.8: Element gadget $G_{el}(x)$

Definition 7.42. *Let $T_i = \{a_{j(i)}, b_{k(i)}, c_{\ell(i)}\}$ be a triple of $\mathcal{T}$. The* triple gadget *$G_t(T_i)$ is defined as follows: $V(G_t(T_i)) = \{p_1, p_2, q_1, q_2, r_1, r_2, s_1, s_2\} \cup V(P(p)) \cup V(P(q)) \cup V(P(r)) \cup V(P(s)) \cup V(Tr(p,q)) \cup V(Tr(r,s)) \cup V(Tr(s,a)) \cup V(Tr(p,r,b)) \cup V(Tr(q,r,c))$, where:*

- *$p = \{p_1, p_2\}$, $q = \{q_1, q_2\}$, $r = \{r_1, r_2\}$, $s = \{s_1, s_2\}$, $a = \{f_{a_{j(i)}}, g_{a_{j(i)}}\}$, $b = \{f_{b_{k(i)}}, g_{b_{k(i)}}\}$ and $c = \{f_{c_{\ell(i)}}, g_{c_{\ell(i)}}\}$;*

- *the two intervals of each pair p, q, r, s intersect with each other;*

- *$P(p)$, $P(q)$, $P(r)$ and $P(s)$ are P_5-gadgets that are included in the intersection of the two intervals of each pair p,q,r, and s, respectively;*

- *$Tr(p,q)$, $Tr(r,s)$, $Tr(s,a)$, $Tr(p,r,b)$ and $Tr(q,r,c)$ are transmitter gadgets. Moreover, their intervals intersect in such a way that for two distinct transmitter gadgets $Tr(x,y,z)$ and $Tr(t,u,v)$, the two intervals of each of the pairs $\{uv^1, uv^2\}$, $\{vw^1, vw^2\}$, x, y and z of $Tr(x,y,z)$ both intersect the same set of intervals from gadget $Tr(t,u,v)$. This can be easily done by placing and "stretching" the intervals appropriately.*

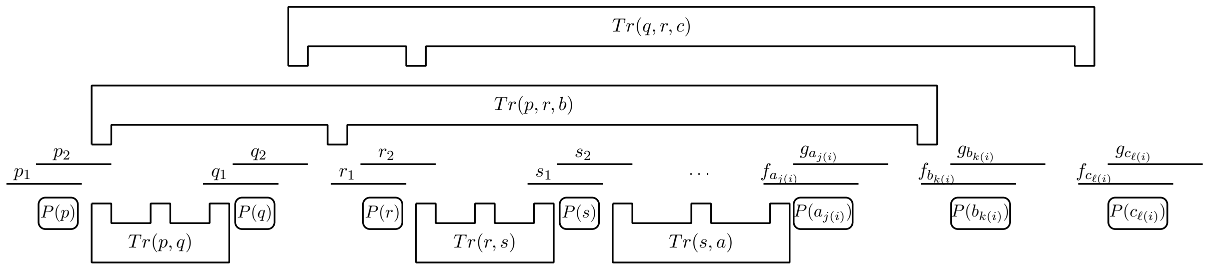


Figure 7.9: A triple gadget $G_t(T_i)$ with $T_i = \{a_{j(i)}, b_{k(i)}, c_{\ell(i)}\}$ together with the element gadgets $G_{el}(a_{j(i)})$, $G_{el}(b_{k(i)})$ and $G_{el}(c_{\ell(i)})$

An illustration of a triple gadget is given in Figure 7.9.

We are now ready to describe the reduction.

Reduction 7.43 (3-Dimensional Matching $\to$ Identifying Code for interval graphs)**.** *Given an instance of* 3-Dimensional Matching *consisting of element sets $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\}, C = \{c_1, \dots, c_n\}$ and triple set $\mathcal{T} = \{T_1, \dots, T_t\}$, we construct in polynomial time the interval graph $Int(A, B, C, \mathcal{T})$. This graph is defined as follows:*

- *For each element x of $A \cup B \cup C$, $Int(A, B, C, \mathcal{T})$ contains an element gadget $G_{el}(x)$ as defined in Definition 7.41. The intervals of any two distinct element gadgets are disjoint. Moreover we assume that the gadgets are positioned consecutively on the real line in the order $G_{el}(a_1), \dots, G_{el}(a_n)$, $G_{el}(b_1), \dots, G_{el}(b_n)$, $G_{el}(c_1), \dots, G_{el}(c_n)$ (though this order does not matter).*

- *For each triple $T_i = \{a_{j(i)}, b_{k(i)}, c_{\ell(i)}\}$ of $\mathcal{T}$, $Int(A, B, C, \mathcal{T})$ contains a triple gadget $G_t(T_i)$ that is connected with the gadgets $G_{el}(a_{j(i)})$, $G_{el}(b_{k(i)})$ and $G_{el}(c_{\ell(i)}\})$ as described in Definition 7.42. The triple gadgets have to intersect; however they must be intersecting in a specific way (in order to prevent "bad" interactions in the reduction). More specifically, for two distinct triples T_i, T_j, the two intervals of each of the pairs $\{uv^1, uv^2\}$, $\{vw^1, vw^2\}$, $\{x_1, x_2\}$, $\{y_1, y_2\}$ and $\{z_1, z_2\}$ of each transmitter gadget of $G_t(T_i)$ both intersect the same set of intervals from gadget $G_t(T_j)$. This can be done by placing and "stretching" the intervals appropriately. All triple gadgets are placed consecutively on the real line (on the left of all element gadgets), in the order: $G_t(T_1), \dots, G_t(T_t)$.*

First of all, we note that $Int(A, B, C, \mathcal{T})$ is identifiable, provided that each element from $A \cup B \cup C$ is contained in some triple of $\mathcal{T}$, that is, $(A, B, C, \mathcal{T})$ is a feasible instance of 3-Dimensional Matching. In order to determine the properties of Reduction 7.43, we first prove a few claims on the gadgets and the reduction itself.

Claim 7.44. *Let G be a identifiable interval graph containing a P_5-gadget P and let $\mathcal{C}$ be an identifying code of G. Then $|\mathcal{C} \cap V(P)| \geq 3$. Moreover, one can always construct an identifying code $\mathcal{C}'$ from $\mathcal{C}$ with $|\mathcal{C}'| \leq |\mathcal{C}|$ and $\mathcal{C} \cap V(P) = \{a_P, c_P, e_P\}$.*

Proof. Vertex c_P belongs to $\mathcal{C}$ since it is the only vertex separating a_P, b_P. In order to separate b_P, c_P, either a_P or d_P belongs to $\mathcal{C}$; similarly, in order to separate c_P, d_P, either b_P or e_P belong to $\mathcal{C}$. Since by the definition of a P_5-gadget, any interval from $V(G) \setminus V(P)$ either intersects all intervals of P or none, the composition of $\mathcal{C} \cap V(P)$ does not influence the separation or domination of intervals other than the ones of P (provided $\mathcal{C} \cap V(P) \neq \emptyset$). Now it is easy to see that $\{a_P, c_P, e_P\}$ correctly separate and dominate all intervals of P. Moreover, all intervals that intersect P are dominated by the three of a_P, c_P, e_P, but this is not the case for any interval from P. This implies that they are correctly separated from any other interval of G. Hence, replacing $\mathcal{C} \cap V(P)$ by $\{a_P, c_P, e_P\}$ in $\mathcal{C}$ yields the claimed identifying code $\mathcal{C}'$. ☆

Given an interval graph G containing a P_5-gadget P, we call the set $\{a_P, c_P, e_P\}$ the *locally normalized identifying code* of P.

We continue with a few claims regarding the other gadgets.

Claim 7.45. *Let G be an identifiable interval graph with a transmitter gadget $Tr(S)$.*
(1) When considering the union of the locally normalized identifying codes of the five P_5-gadgets in $Tr(S)$, all vertices of $Tr(S)$ are dominated, and all pairs of vertices from $Tr(S)$ are separated, except for the pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$.
(2) In an identifying code $\mathcal{C}$ of G, either $v \in \mathcal{C}$, or both $u, w \in \mathcal{C}$.

Proof. For the first part of the claim, all intervals of the P_5-gadgets are identified following Claim 7.44. Each other interval is dominated by the vertices of the P_5-gadget that intersect with it. All intervals having a "private" P_5-gadget are separated from all other intervals; only the vertices of the pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$ share their two P_5-gadgets: $P(uv)$ and $P(vw)$, respectively.

For the second part of the claim, we observe that v is the only interval that can separate both $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$; if $v \notin \mathcal{C}$, only u can separate $\{uv^1, uv^2\}$ and only w can separate $\{vw^1, vw^2\}$. ☆

In what follows, we let $(A, B, C, \mathcal{T})$ be an instance of 3-DIMENSIONAL MATCHING with $|A| = |B| = |C| = n$ and $|\mathcal{T}| = t$, and we consider the graph $G = Int(A, B, C, \mathcal{T})$ constructed in Reduction 7.43.

Claim 7.46. *Let $\mathcal{C}^-$ be the union of the locally normalized identifying codes of all P_5-gadgets of $G = Int(A, B, C, \mathcal{T})$. Then all intervals are dominated by $\mathcal{C}^-$, and all pairs of intervals of G are separated by $\mathcal{C}^-$, except for:*

- *the pairs $\{p_1, p_2\}$, $\{q_1, q_2\}$, $\{r_1, r_2\}$, $\{s_1, s_2\}$ of each triple gadget $G_t(T_i)$;*
- *the pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$ of each of the five transmitter gadgets of each triple gadget $G_t(T_i)$;*
- *the pair $\{f_x, g_x\}$ of each element gadget $G_{el}(x)$.*

Proof. By Claims 7.45 and 7.44, the claim holds for the intervals of each P_5-gadget and each transmitter gadget. In fact, every interval i that is not in a P_5-gadget contains the intervals of some P_5-gadget P, hence i is dominated by the vertices of $P \cap \mathcal{C}^-$. Furthermore, one can check that except for the pairs listed in the claim, each pair i, j of such intervals includes a distinct set of P_5-gadgets, hence i, j are separated by the code vertices of some P_5-gadget. ☆

The next claim proves the first side of the reduction: given a 3-dimensional matching of a (feasible) instance $(A, B, C, \mathcal{T})$ of 3-DIMENSIONAL MATCHING, we can construct an identifying code of $G = Int(A, B, C, \mathcal{T})$ having a certain size.

Claim 7.47. *If $(A, B, C, \mathcal{T})$ has a 3-dimensional matching $\mathcal{M}$, then $G = Int(A, B, C, \mathcal{T})$ has an identifying code of size at most $12n + 94t - 2|\mathcal{M}|$.*

Proof. Let $\mathcal{C}^-$ be the union of the locally normalized identifying codes of all P_5-gadgets of G. Since each element gadget contains one P_5-gadget and each triple gadget contains twenty-nine P_5-gadgets (five in each of its five transmitter gadgets, and four other ones), we have $|\mathcal{C}^-| = 9n + 87t$.

Let us now define the sets $\mathcal{C}_{\mathcal{M}}$ (for the triples in $\mathcal{M}$), $\mathcal{C}_{\mathcal{T}\setminus\mathcal{M}}$ (for the triples out of $\mathcal{M}$) and $\mathcal{C}_{um}$ (for the elements that are unmatched by $\mathcal{M}$) as follows.

For each triple T_i of $\mathcal{M}$, the intervals u, w of each of the three transmitter gadgets $Tr(s, a)$, $Tr(p, r, b)$ and $Tr(q, r, c)$ of $G_t(T_i)$ belong to $\mathcal{C}_{\mathcal{M}}$, and interval v of each of the two remaining transmitter gadgets $Tr(p, q)$ and $Tr(r, s)$ of $G_t(T_i)$ belong to $\mathcal{C}_{\mathcal{M}}$. This yields $|\mathcal{C}_{\mathcal{M}}| = 8|\mathcal{M}|$.

For each triple T_i of $\mathcal{T} \setminus \mathcal{M}$, interval v of each of the three transmitter gadgets $Tr(s, a)$, $Tr(p, r, b)$ and $Tr(q, r, c)$ of $G_t(T_i)$ belong to $\mathcal{C}_{\mathcal{T}\setminus\mathcal{M}}$, and the intervals u, w of each of the two remaining transmitter gadgets $Tr(p, q)$ and $Tr(r, s)$ of $G_t(T_i)$ belong to $\mathcal{C}_{\mathcal{T}\setminus\mathcal{M}}$. Hence $|\mathcal{C}_{\mathcal{T}\setminus\mathcal{M}}| = 7(t - |\mathcal{M}|)$.

For each element x of $A \cup B \cup C$ that does not belong to any triple of $\mathcal{M}$, some arbitrary interval separating the intervals f_x, g_x in $G_{el}(x)$ belongs to $\mathcal{C}_{um}$ (such an element exists since

$(A, B, C, \mathcal{T})$ is feasible, hence there must be a triple gadget with a transmitter gadget that has an interval intersecting f_x but not g_x). We have $|\mathcal{C}_{um}| \leq 3n - 3|\mathcal{M}|$.

Now, we claim that the code $\mathcal{C} = \mathcal{C}^- \cup \mathcal{C}_{\mathcal{M}} \cup \mathcal{C}_{\mathcal{T}\setminus\mathcal{M}} \cup \mathcal{C}_{um}$ is a valid identifying code of G. Note that we have $|\mathcal{C}| \leq 9n + 87t + 8|\mathcal{M}| + 7(t - |\mathcal{M}|) + 3n - 3|\mathcal{M}| = 12n + 94t - 2|\mathcal{M}|$.

By Claim 7.46, we know that $\mathcal{C}^- \subseteq \mathcal{C}$ dominates all vertices and separates all pairs but the pairs $\{p_1, p_2\}$, $\{q_1, q_2\}$, $\{r_1, r_2\}$, $\{s_1, s_2\}$ of each triple gadget $G_t(T_i)$, the pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$ of each of the five transmitter gadgets of each triple gadget $G_t(T_i)$, and the pair $\{f_x, g_x\}$ of each element gadget $G_{el}(x)$. However, observe that for each transmitter gadget, either vertex v, or both vertices u, w belong to the code. Hence, all pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$ are separated. Similarly, for each triple T_i from $\mathcal{M}$, the vertices from $V(G_t(T_i)) \cap \mathcal{C}_{\mathcal{M}}$ separate the pairs $\{p_1, p_2\}$, $\{q_1, q_2\}$, $\{r_1, r_2\}$, $\{s_1, s_2\}$ in $G_t(T_i)$. Moreover, they separate all pairs $\{f_x, g_x\}$ such that x is covered by T_i in $\mathcal{M}$. Furthermore, the pairs $\{f_x, g_x\}$ for elements x that are not covered by any triple of $\mathcal{M}$, are separated by $\mathcal{C}_{um}$. Finally, for any triple T_j of $\mathcal{T}\setminus\mathcal{M}$, the vertices of $V(G_t(T_j)) \cap \mathcal{C}_{\mathcal{T}\setminus\mathcal{M}}$ separate pairs $\{p_1, p_2\}$, $\{q_1, q_2\}$, $\{r_1, r_2\}$, $\{s_1, s_2\}$ in $G_t(T_j)$. Hence, $\mathcal{C}$ is a valid identifying code of G. ☆

Claim 7.48. *Let $\mathcal{C}$ be an identifying code of $G = Int(A, B, C, \mathcal{T})$. In each triple gadget $G_t(T_i)$ of G, there are at least two transmitter gadgets for which $|\mathcal{C} \cap \{u, v, w\}| = 2$. If there are exactly two, then these gadgets must be $Tr(p, q)$ and $Tr(r, s)$. Otherwise, we can assume that there are exactly three of them, and that these three can be $Tr(s, a)$, $Tr(p, r, b)$ and $Tr(q, r, c)$.*

Proof. In $G_t(T_i)$, the intervals from the pairs p, q, r, s as defined in Definition 7.42 need to be separated. Observe that the only way to separate the intervals from p is to use either interval u of $Tr(p, q)$, or interval u of $Tr(p, r, b)$. Similarly, to separate pair q, we have to use either interval u of $Tr(q, r, c)$, or interval w of $Tr(p, q)$; to separate pair r, we need either interval u of $Tr(r, s)$, or interval w of $Tr(p, r, b)$. Finally, to separate pair s, we need either interval w of $Tr(r, s)$, or interval u of $Tr(s, a)$.

Recall that by Claim 7.45(2), for each transmitter gadget $Tr(S)$, if $|\mathcal{C} \cap \{u, v, w\}| \geq 2$, we can assume that $\{u, w\} \subseteq (\mathcal{C} \cap \{u, v, w\})$; otherwise, $\mathcal{C} \cap \{u, v, w\} = \{v\}$. Hence, when $|\mathcal{C} \cap \{u, v, w\}| < 2$, the intervals of $Tr(S)$ do not help separating any of the pairs p, q, r, s. Furthermore, one can easily check (using the previous paragraph) that the only way to separate all four pairs using only two transmitter gadgets with $\mathcal{C} \cap \{u, v, w\} = \{u, w\}$ is to take $Tr(p, q)$ and $Tr(r, s)$. If we assume that there are at least three of them, taking $Tr(s, a)$, $Tr(p, r, b)$ and $Tr(q, r, c)$ to have $\mathcal{C} \cap \{u, v, w\} = \{u, w\}$ separates all four pairs p, q, r, s. Moreover, by our construction, this does not cause any conflict with other pairs. ☆

We are now ready to prove the other important claim of our proof.

Claim 7.49. *Let $(A, B, C, \mathcal{T})$ be an instance of* 3-DIMENSIONAL MATCHING. *If the graph $G = Int(A, B, C, \mathcal{T})$ has an identifying code $\mathcal{C}$ of size at most $10n + 94t$, then $(A \cup B \cup C, \mathcal{T})$ has a perfect 3-dimensional matching.*

Proof. First of all, by Claim 7.44, each P_5-gadget has at least three intervals in $\mathcal{C}$; again by Claim 7.44, we may assume that each P_5-gadget has *exactly* three intervals in $\mathcal{C}$ (otherwise it is easy to transform $\mathcal{C}$ so that this property holds). Since in total there are $29t + 3n$ P_5-gadgets in G, they account for $87t + 9n$ intervals of $\mathcal{C}$.

Now, following Claim 7.45(2), each of the five transmitter gadgets in each triple gadget $G_t(T_i)$ have at least one vertex among $\{u, v, w\}$ belonging to $\mathcal{C}$. Moreover, by Claim 7.48, we can assume that either two or three of these transmitter gadgets are such that $|\mathcal{C} \cap \{u, v, w\}| = 2$ (we call such a transmitter gadget *full*). Hence, if $G_t(T_i)$ has two full transmitter gadgets, the intervals of these transmitter gadgets that belong to $\mathcal{C}$ and that do not belong to P_5-gadgets account for $2 \cdot 2 + 3 \cdot 1 = 7$ intervals. If $G_t(T_i)$ has three full transmitter gadgets, they account for $3 \cdot 2 + 2 \cdot 1 = 8$ intervals.

Let t^+ be the number of triple gadgets that have three full transmitter gadgets. By the previous paragraph, the total number of intervals of $\mathcal{C}$ among the intervals $\{u, v, w\}$ of all transmitter gadgets account for $8t^+ + 7(t - t^+) = 7t + t^+$ intervals. By the first paragraph of this proof, we have $7t + t^+ \leq |\mathcal{C}| - 87t - 9n$ and by assumption, $|\mathcal{C}| = 10n + 94t$. These two facts imply $t^+ \leq n$. In fact we can assume that $t^+ = n$, indeed we can assume that there is no code vertex (other than in a P_5-gadget) outside of a triple gadget (otherwise, one could just remove this vertex and still get an identifying code).

Now, by Claim 7.48, for each of the $t^+ = n$ triples that have three full transmitter gadgets, these transmitter gadgets can be assumed to be $Tr(s, a)$, $Tr(p, r, b)$ and $Tr(q, r, c)$. By Claim 7.45(2), for each of these transmitter gadgets, vertex w belongs to $\mathcal{C}$. Since all pairs $\{f_x, g_x\}$ of each of the $3n$ element gadgets $G_{el}(x)$ are separated by $\mathcal{C}$ and each transmitter gadget separates at most one such pair, each pair $\{f_x, g_x\}$ is separated by vertex w of exactly one transmitter gadget. Hence, selecting the set T of t^+ triples with three full transmitter gadgets of $G_t(T)$ yields a perfect 3-dimensional matching of $(A \cup B \cup C, \mathcal{T})$. ☆

We can now prove the following theorem:

Theorem 7.50. IDENTIFYING CODE *restricted to interval graphs (and hence, to (un)directed path graphs, to trapezoid and to strongly chordal graphs) is* NP*-complete.*

Proof. We know that IDENTIFYING CODE is in NP; additionally, we show that an instance $(A, B, C, \mathcal{T})$ with $|A| = |B| = |C| = n$ and $|\mathcal{T}| = t$ of 3-DIMENSIONAL MATCHING has a perfect 3-dimensional matching if and only if $Int(A, B, C, \mathcal{T})$ has an identifying code of size $10n + 94t$. For the first side, a perfect matching of $(A, B, C, \mathcal{T})$ has necessarily n triples; hence by Claim 7.47, $Int(A, B, C, \mathcal{T})$ has an identifying code of size $12n + 94t - 2n = 10n + 94t$. The other side is proved in Claim 7.49. ☆

7.5 Conclusion

In this chapter, we exhibited several new classes of graphs for which MIN ID CODE is APX-complete or IDENTIFYING CODE is NP-complete.

In particular, we proved that MIN ID CODE is APX-complete for bipartite graphs of maximum degree 4, and that IDENTIFYING CODE is NP-complete for planar bipartite graphs of maximum degree 4. It was shown that MIN DOM SET is APX-complete in subcubic bipartite graphs [53] and in cubic graphs [1]. Can we say the same for MIN ID CODE? Moreover, note that a reduction from MIN VERTEX COVER to MIN ID CODE for subcubic graphs of girth 9 is provided in [9]. We observed (Theorem 2.52) that this reduction is in fact an L-reduction, and proves the APX-hardness of MIN ID CODE for subcubic graphs. However, this reduction relies on a gadget using 9-cycles and therefore it is not suitable for bipartite graphs.

Question 7.51. *Is* MIN ID CODE *APX-complete when restricted to subcubic bipartite graphs? When restricted to cubic (bipartite) graphs?*

Similarly, DOMINATING SET is NP-complete when restricted to planar bipartite subcubic graphs of arbitrarily large girth [186].[3] Does a similar result hold for IDENTIFYING CODE? Note that it is shown in [6] that IDENTIFYING CODE is NP-complete when restricted to planar graphs of maximum degree 4 and arbitrarily large girth.

Question 7.52. *Is* IDENTIFYING CODE *NP-complete when restricted to planar bipartite subcubic graphs? If yes, does this hold when we further restrict this class to the graphs which, additionally, also have arbitrarily large girth?*

We also proved that MIN ID CODE is 4-approximable in the class of line graphs. Recall that a graph is a line graph if and only if it does not admit any induced subgraph from a list $\mathcal{B}$ of nine elements first determined by Beineke (see Theorem 2.2), that is, line graphs are exactly the induced $\mathcal{B}$-free graphs. Hence the following is a natural question:

[3] This result is derived from an easy reduction from DOMINATING SET for subcubic planar graphs (known to be NP-complete [88]) to DOMINATING SET itself by subdividing the edges of the graph $3k$ times for an arbitrary k such that $3k$ is odd.

Question 7.53. *For which minimal subsets $S_{\mathcal{B}}$ of $\mathcal{B}$ can* MIN ID CODE *for $S_{\mathcal{B}}$-free graphs be c-approximated for some constant c?*

Another superclass of line graphs to potentially investigate is the class of quasi-line graphs (we recall that a graph is quasi-line if the neighbourhood of each vertex can be partitioned into two cliques). However, recall that we showed in the conclusion of Chapter 6 (Corollary 6.14) that MIN ID CODE restricted to co-bipartite graphs (and therefore also to quasi-line graphs) is log-APX-hard.

In fact, we note once again (as in the discussion around Question 5.37 in Chapter 5.3) that among the graphs of the list $\mathcal{B}$ of nine forbidden subgraphs characterizing line graphs, only four ($B_2, B_3 = K_5^-, B_6, B_8 = P_6^2$ in Figure 2.15) are not co-bipartite, hence we deduce the following proposition regarding Question 7.53:

Proposition 7.54. *Let $S_{\mathcal{B}}$ be a subset of $\mathcal{B}$. If* MIN ID CODE *is c-approximable in $S_{\mathcal{B}}$-free graphs for some constant c, then $S_{\mathcal{B}}$ contains at least one of the graphs $B_2, B_3 = K_5^-, B_6, B_8 = P_6^2$.*

The fact that IDENTIFYING CODE is NP-complete for interval graphs is particularly interesting since many problems (such as DOMINATING SET) are in P when restricted to interval graphs. However, we leave the question of the complexity of approximating MIN ID CODE for interval graphs open. In particular we have not managed to prove that Reduction 7.43 is an L-reduction, as opposed to the reductions of the previous sections.[4] A possible way to do so would be to extend Claim 7.49 so that a (not necessarily perfect) matching of an appropriate size can be constructed from an identifying code, as was done for example in Claim 7.35 for Reduction 7.25 for MIN ID CODE for line graphs.

Question 7.55. *Is* MIN ID CODE *restricted to interval graphs APX-hard? Is it in APX?*

Moreover, a graph class that, like the one of interval graphs, is also a subclass of trapezoid graphs is the class of permutation graphs. It is a class where many hard problems become polynomial-time solvable. This is the case for DOMINATING SET [83]. Hence, we ask what happens for IDENTIFYING CODE:

Question 7.56. *What is the complexity of* IDENTIFYING CODE *for permutation graphs?*

We end this conclusion by noting that MIN DOM SET and related problems admit PTAS algorithms in the class of planar graphs. This was shown by using layerwise decomposition of these graphs [15], and has been extended to any class of graphs excluding a fixed minor using the technique of *bidimensionality* [72].

Question 7.57. *Does* MIN ID CODE *admit a PTAS for planar graphs? If yes, does this hold for classes of graphs excluding a fixed minor?*

[4] Since the maximization version of 3-DIMENSIONAL MATCHING, MAX 3-DIM MATCHING, is APX-complete even in its variant where each element appears in at most two triples [51], this would prove that MIN ID CODE is APX-hard even for interval graphs.

Chapter 8

Graph classes where MIN ID CODE is in PTAS or in PO

IN this chapter, we first discuss the computational complexity of MIN ID CODE for unit interval graphs in Section 8.1. We prove in Proposition 8.4 that this restriction of MIN ID CODE can be reduced to an interesting covering problem (that we call MIN LADDER CYCLE COVER) whose complexity is unknown and which seems to be unstudied in the literature. Despite the fact that we are not able to solve this problem, we however provide a PTAS for MIN LADDER CYCLE COVER which implies a PTAS for MIN ID CODE for unit interval graphs (Theorem 8.7).

In Section 8.2, we extend known results about IDENTIFYING CODE for graphs of bounded tree-width to EDGE-IDENTIFYING CODE using Courcelle's theorem, proving that this problem can be solved in linear time for graphs of bounded tree-width (Corollary 8.11).

Finally, in Section 8.3, we define a class of graphs, SC-graphs, for which solving IDENTIFYING CODE is computationally easy but solving DOMINATING SET is hard (see Corollary 8.14 and Theorem 8.15). No such class was previously known.

The results about unit interval graphs are joint work from [FKM+12] with A. Kosowski, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov. The core idea of the PTAS was suggested by N. E. Young in an online discussion [202]. The ones about line graphs of graphs of bounded tree-width appeared in [FGN+12] (joint work with S. Gravier, R. Naserasr, A. Parreau and P. Valicov). The ones of Section 8.3 are solely the author's work; they are new and did not appear previously.

8.1 IDENTIFYING CODE for unit interval graphs

We saw in Section 7.4 that IDENTIFYING CODE is NP-complete when restricted to interval graphs, but the interval graphs constructed in Reduction 7.43 are far from being unit interval graphs. We now discuss this case.

First of all, recall that by Theorem 5.11, for any identifiable unit interval graph G, we have $\frac{|V(G)|+1}{2} \leq \gamma^{\text{ID}}(G) \leq |V(G)|$. Hence, taking as an identifying code the whole vertex set, we have a trivial 2-approximation:

Corollary 8.1. MIN ID CODE *is 2-approximable when restricted to the class of unit interval graphs.*

8.1.1 Reducing MIN ID CODE to MIN LADDER CYCLE COVER

We can relate IDENTIFYING CODE for unit interval graphs to another problem, that will allow us to improve Corollary 8.1.

Let us call the grid graph $P_2 \Box P_m$, denoted L_m, *ladder graph*. We consider the vertex set of L_m as $\{1, 2\} \times \{1, \ldots, m\}$. We call the edges coming from P_2, *step edges* of L_m (a step edge $\{(1, i), (2, i)\}$ is denoted e_i); the other edges are *side edges* of L_m. Note that any cycle of L_m is determined by the two step edges e_i, e_j that it contains; we denote this cycle by $S_{i,j}$. A *cycle cover* of L_m is a set $\mathcal{S}$ of cycles that covers the whole graph L_m, i.e. $\bigcup_{S\in\mathcal{S}} E(S) = E(L_m)$. An example of a cycle cover of L_5 with three cycles is given in Figure 8.1.

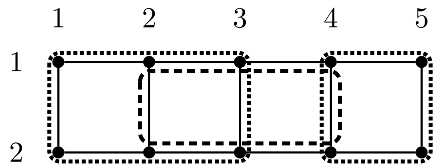


Figure 8.1: The ladder L_5 and one of its cycle covers (dotted cycles).

We define the following related decision and optimization problems:

LADDER CYCLE COVER
INSTANCE: An integer m and an integer k, and a set $\mathcal{S}$ of cycles of L_m.
QUESTION: Is there a set $\mathcal{S}' \subseteq \mathcal{S}$ of size k which is a cycle cover of L_m?

MIN LADDER CYCLE COVER
INSTANCE: An integer m and a set $\mathcal{S}$ of cycles of L_m.
SOLUTION: A cycle cover $\mathcal{S}' \subseteq \mathcal{S}$ of L_m.
MEASURE: The size $|\mathcal{S}'|$ of the cycle cover.

In the following, given a unit interval graph G on n vertices, we consider its vertex set $V(G) = \{1, \ldots, n\}$, where $1 \leq \ldots \leq n$ is the natural ordering of the corresponding intervals introduced in Section 5.2. For the sake of simplicity, we restrict our analysis to connected unit interval graphs; since any identifying code of a disconnected graph is the union of identifying codes of its connected components, this study extends easily to the general case. We will need the following lemma.

Lemma 8.2. *Let G be a connected identifiable unit interval graph. Let i be a vertex of G and let i_l, i_r be the two vertices of G such that i_l is the neighbour of i of smallest index in G, and i_r is the neighbour of i of biggest index in G. Then the set $\mathcal{D}_i$ of vertices that are dominated by i is exactly $\{i_l, \ldots, i_r\}$. Moreover, the set $\mathcal{P}_i$ of pairs of consecutive vertices separated by i is exactly:*

$$\mathcal{P}_i = \begin{cases} \{\{i_{l-1}, i_l\}, \{i_r, i_{r+1}\}\} & \text{if } i_l \neq 1 \text{ and } i_r \neq n, \\ \{\{i_r, i_{r+1}\}\} & \text{if } i_l = 1 \text{ and } i_r < n, \\ \{\{i_{l-1}, i_l\}\} & \text{if } i_l > 1 \text{ and } i_r = n, \\ \emptyset & \text{if } i_l = 1 \text{ and } i_r = n. \end{cases}$$

Proof. For $\mathcal{D}_i$, the claim is obvious. For $\mathcal{P}_i$, we have seen in Proposition 5.10 that i may separate at most two pairs of consecutive vertices. Vertex i will indeed separate the pair of vertices that consists of i_l and i_{l-1} if i_{l-1} exists (and similarly, i_r and i_{r+1} if i_{r+1} exists), i.e. if $i_l \neq 1$ ($i_r \neq n$, respectively). ☆

Reduction 8.3 (MIN ID CODE for unit interval graphs → MIN LADDER CYCLE COVER). *Given a connected identifiable unit interval graph G on n vertices, we construct a set $\mathcal{S}(G) = \{S_1, \ldots, S_n\}$ of n cycles of L_{n+1}. For each vertex i of G, we have a cycle $S(i)$. Let i_l, i_r be the two vertices of G such that i_l is the neighbour of i of smallest index in G, and i_r is the neighbour of i of biggest index in G. Then, $S_i = S_{i_l, i_r+1}$.*

Proposition 8.4. *Let G be a connected identifiable unit interval graph on n vertices. A set $\mathcal{C} = \{c_1, \ldots, c_k\}$ of k vertices of G is an identifying code of G if and only if the set of cycles $\{S(c_1), \ldots, S(c_k)\} \subseteq \mathcal{S}(G)$ as defined in Reduction 8.3 is a cycle cover of L_{n+1}.*

Proof. In Reduction 8.3, we let each step edge e_j among $\{e_2, \ldots, e_n\}$ correspond to the pair $\{j-1, j\}$ of consecutive vertices of G. For each $i \in \{2, \ldots, n-1\}$, each pair of side edges $\{(1,i),(1,i+1)\}$ and $\{(2,i),(2,i+1)\}$ represents vertex i of G. Finally, vertex 1 is represented by step edge e_1 and side edges $\{(1,1),(1,2)\}$ and $\{(2,1),(2,2)\}$; similarly, vertex n is represented by step edge e_{n+1} and side edges $\{(1,n),(1,n+1)\}$ and $\{(2,n),(2,n+1)\}$. Observe that these sets form a partition of $E(L_{n+1})$ into n sets, each corresponding to a vertex of G.

By Proposition 5.12, a set $\mathcal{C}$ of vertices of G is an identifying code if and only if it separates all pairs of consecutive vertices of G and it dominates G. Using Lemma 8.2 and the previous paragraph, we now observe that there is a one-to-one correspondence between objects of G that need to be separated or dominated (i.e. vertices and pairs of consecutive vertices) and the subsets of the partition of $E(L_{n+1})$ described above. Moreover, each cycle $S(i)$ of $\mathcal{S}(G)$ covers exactly the subset of $E(L_{n+1})$ that corresponds to the elements of $\mathcal{D}_i \cup \mathcal{P}_i$ as defined in Lemma 8.2. This completes the proof. ☆

By Proposition 8.4, if LADDER CYCLE COVER is in P, then IDENTIFYING CODE for unit interval graphs is also in P. However, we do not know the complexity of LADDER CYCLE COVER.

Let us make a few remarks. First of all, in MIN LADDER CYCLE COVER, if one is asking to cover only side edges of L_m, this problem can be reduced to MIN DOM SET in a certain interval graph (the interval graph defined by the restriction of the input cycles to the side edges of the form $\{(1,i),(1,i+1)\}$, together with a small additional interval for each side edge). This problem is solvable in linear time [29].

Similarly, if we ask, in MIN LADDER CYCLE COVER, to cover only step edges, then the corresponding problem is polynomial-time solvable, as it can be reduced to MIN EDGE COVER.[1] As it will be used later on, we formalize this reduction as follows.

Reduction 8.5 ("Covering step edges of a ladder" → MIN EDGE COVER). *Given a ladder L_m and a set $\mathcal{S}$ of cycles of L_m, we construct the graph $G(L_m, \mathcal{S})$, where the vertex set of $G(L_m, \mathcal{S})$ is the set $\{e_1, \ldots, e_m\}$ of step edges of G, and there is an edge between two vertices e_i, e_j if and only if the cycle $S_{i,j}$ belongs to $\mathcal{S}$.*

The following proposition is now trivial:

Proposition 8.6. *Let $\mathcal{S}$ be a set of cycles of L_m. A subset $\mathcal{C}$ of $\mathcal{S}$ covers all step edges of L_m if and only if $G = G(L_m, \mathcal{S})$ as defined in Reduction 8.5 has an edge cover of size $|\mathcal{C}|$. Hence, finding a minimum-size set of cycles from $\mathcal{S}$ can be done in time $O\left(\sqrt{|V(G)|} \cdot |E(G)|\right) = O\left(m^{5/2}\right)$.*

We also remark that the complexity of IDENTIFYING CODE for unit interval graphs does not capture the whole complexity of LADDER CYCLE COVER. Indeed, the instances of LADDER CYCLE COVER given by Reduction 8.3 are restricted by the structure of unit interval graphs. For example, the input set of cycles of L_m has $m-1$ elements, and one can check that certain configurations of input cycles are not allowed (e.g. a cycle passing through two consecutive step edges, or a set of "cyclic" cycles such as, for instance, three cycles of the form $S_{i,j}$, $S_{j,k}$, $S_{i,k}$). Hence, LADDER CYCLE COVER might be NP-complete, but IDENTIFYING CODE for unit interval graphs, polynomial-time solvable.

8.1.2 MIN ID CODE for unit interval graphs is in PTAS

When it comes to the optimization problem MIN LADDER CYCLE COVER, we are able to show that it belongs to the class PTAS. The proof of this theorem was suggested by N. E. Young in [202] on the theoretical computer science research-level answers and questions website `http:`

[1]Recall from Corollary 2.15 that MIN EDGE COVER is solvable in time $O\left(\sqrt{|V(G)|} \cdot |E(G)|\right)$ for an input graph G by using an algorithm solving MAX MATCHING.

//cstheory.stackexchange.com, answering one of my questions on the complexity of LADDER CYCLE COVER. The proof which follows is a completed and formalized version of this sketch.

Theorem 8.7. MIN LADDER CYCLE COVER *is in PTAS.*

Proof. Let L_m be a ladder, $\mathcal{S}$ be a set of cycles of L_m, and $\epsilon > 0$. We will give a polynomial-time approximation scheme solving MIN LADDER CYCLE COVER, i.e. we compute a cycle cover having size $(1+\epsilon)OPT(L_m)$ in polynomial time (for fixed ϵ).

The algorithm can be sketched as follows: we "slice" the ladder L_m into roughly $m\epsilon$ sections, each centered around some specific step edge. We then use these sections to compute a set of locally optimal covers. Among all these covers, the smallest one is chosen using a dynamic programming-like technique.

First of all, we assume that the input instance has a feasible solution, as this is easy to check in polynomial time beforehand.

Let $k = \lfloor \frac{m\epsilon}{4} \rfloor$. We mark every $\frac{m}{k}$'th step edge: consider $a_1, \ldots, a_k$, where for each i, $1 \le i \le k$, we let $a_i = e_{i(m/k)}$. Moreover, v_i denotes the vertex $(1, i(m/k))$ of a_i. Consider the set $\mathcal{S}_i$ of cycles of $\mathcal{S}$ that contain vertex v_i. Each such cycle is of the form $S_{j,k}$ with $j \le i(m/k)$ and $k \ge i(m/k)$. Let $\ell(i)$ be the minimum first index, and let $r(i)$ be the maximum second index of any of these cycles: $\ell(i) = \min_{S_{j,k} \in \mathcal{S}_i} j$ and $r(i) = \max_{S_{j,k} \in \mathcal{S}_i} k$. Moreover let $S_{\ell(i)}, S_{r(i)}$ be two (arbitrary) cycles of $\mathcal{S}_i$ containing edge $e_{\ell(i)}$, $e_{r(i)}$, respectively. We call the subgraph of L_m induced by all vertices $(1,a)$ and $(2,a)$ of index a with $\ell(i) \le a \le r(i)$, the *block* around a_i, denoted B_i. Moreover, for two consecutive blocks B_i, B_{i+1} $(1 \le i \le k-1)$, the subgraph of L_m that lies between these two blocks and that is not included within any block is called a *block neighbourhood* and is denoted N_i. The similar subgraphs lying before B_1 and after B_k are denoted N_0 and N_k, respectively. We note that we might have some subgraph N_i that is empty, e.g. if the blocks B_i, B_{i+1} are overlapping. An illustration of blocks and block neighbourhoods is given in Figure 8.2.

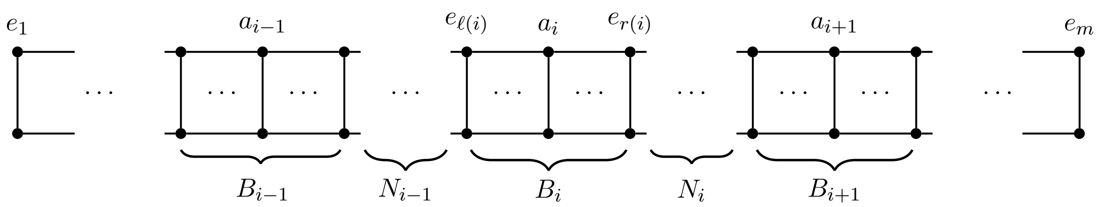


Figure 8.2: Dividing L_m into blocks and neighbourhood blocks.

Given a block neighbourhood N_i and given any set $\mathcal{S}_i$ of cycles of L_m *each including a step edge of* N_i, we say that $\mathcal{S}_i$ is *valid* for N_i if it covers all edges of N_i, i.e. $E(N_i) \subseteq \bigcup_{S \in \mathcal{S}_i} E(S)$ (if N_i is empty then $\emptyset$ is the only valid set of cycles for N_i). The collection of all sets of cycles that are valid for N_i is denoted $\mathcal{V}_i$.

Consider a block B_i and the two surrounding block neighbourhoods N_{i-1} and N_i. For each pair $\mathcal{S}_1 \in \mathcal{V}_{i-1}$, $\mathcal{S}_2 \in \mathcal{V}_i$ of valid sets of cycles, we denote by $\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)$ a minimum-size set of cycles of $\mathcal{S}$ required in order to cover the *step* edges of B_i that are not covered by any cycle of $\mathcal{S}_1$ nor $\mathcal{S}_2$. We remark that $\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)$ can be computed using Reduction 8.5 to MIN EDGE COVER.

We now define a directed acyclic graph $\overrightarrow{D}$ having vertex set:

$$\{s,t\} \cup \bigcup_{i \in \{0,\ldots,k\}} \mathcal{V}_i$$

and arc set:

$$\begin{aligned} A(\overrightarrow{D}) = & \{(s, \mathcal{S}_0) \mid \mathcal{S}_0 \in \mathcal{V}_0\} \\ & \cup \bigcup_{i \in \{1,\ldots,k-1\}} \{(\mathcal{S}_1, \mathcal{S}_2) \mid \mathcal{S}_1 \in \mathcal{V}_i, \mathcal{S}_2 \in \mathcal{V}_{i+1}\} \\ & \cup \{(\mathcal{S}_k, t) \mid \mathcal{S}_k \in \mathcal{V}_k\}. \end{aligned}$$

Moreover, each arc $(s, \mathcal{S}_0)$ starting from s is assigned weight 0; each arc $(\mathcal{S}_1, \mathcal{S}_2)$ with $\mathcal{S}_1 \in \mathcal{V}_{i-1}, \mathcal{S}_2 \in \mathcal{V}_i$ for some $i \in \{1, \dots, k-1\}$ is assigned weight $|\mathcal{S}_1| + |\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)|$; each arc $(\mathcal{S}_k, t)$ is assigned weight $|\mathcal{S}_k|$. Digraph $\overrightarrow{D}$ is illustrated in Figure 8.3.

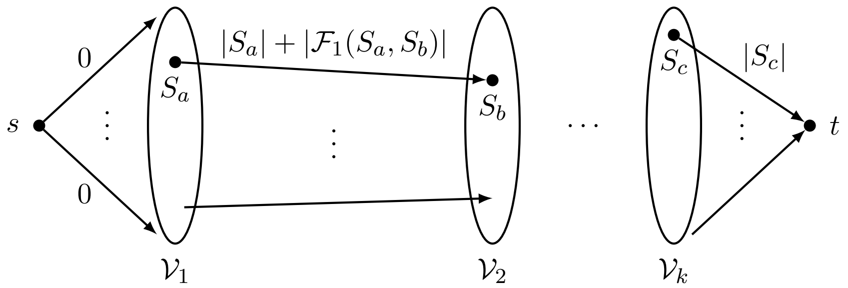


Figure 8.3: The directed acyclic graph $\overrightarrow{D}$.

We can now describe the PTAS: see Algorithm 8.4.

Algorithm 8.4 PTAS for Min Ladder Cycle Cover

Input: A ladder L_m, a set of cycles $\mathcal{S}$ of L_m, some $\epsilon > 0$.

1: $\mathcal{C} \leftarrow \emptyset$
2: Compute the edges $a_1, \dots, a_k$, the cycles $S_{\ell(i)}, S_{r(i)}$ for each $i \in \{1, \dots, k\}$, the blocks $B_1, \dots, B_k$ and the block neighbourhoods $N_0, \dots, N_k$.
3: Add all cycles $S_{\ell(i)}, S_{r(i)}$ to $\mathcal{C}$.
4: **for** each i from 0 to k **do**
5:     Compute the collection $\mathcal{V}_i$ of all valid sets of cycles of N_i
6: **end for**
7: **for** each i from 1 to k **do**
8:     **for** each pair $\mathcal{S}_1, \mathcal{S}_2$ with $\mathcal{S}_1 \in \mathcal{V}_{i-1}$, $\mathcal{S}_2 \in \mathcal{V}_i$ **do**
9:         Compute the set $\mathcal{F}(\mathcal{S}_1, \mathcal{S}_2)$ using Reduction 8.5 to Min Edge Cover.
10:     **end for**
11: **end for**
12: Construct the directed acyclic graph $\overrightarrow{D}$
13: Compute a shortest weighted directed path P from s to t in $\overrightarrow{D}$
14: Add to $\mathcal{C}$ all cycles given by the union of the cycles of the vertices of P, together with the union of all cycles of $\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)$ for each arc $(\mathcal{S}_1, \mathcal{S}_2)$ of P
15: **return** $\mathcal{C}$

Let us proceed with the analysis of Algorithm 8.4. We begin with analyzing its running time. First of all, step 2 of the algorithm takes $O(k + m^2) = O(\epsilon m + m^2)$ time, since there are $O(k) = O(m\epsilon)$ blocks and block neighbourhoods to compute, and at most $\binom{m}{2} = O(m^2)$ cycles in $\mathcal{S}$.

Notice that each neighbourhood block N_i contains at most $\frac{m}{k}$ step edges, since there are at most $\frac{m}{k}$ step edges between any two edges a_i, a_{i+1} with $i \in \{1, \dots, k-1\}$ (and similarly, between e_1, a_1 and a_k, e_m, respectively). Hence, for each neighbourhood block N_i, there are at most $\binom{m/k}{2}$ cycles containing a step edge of N_i; indeed, by definition of a block, such a cycle cannot contain vertex v_i nor v_{i+1}. As a consequence, we have that for each $i \in \{1, \dots, k\}$, $|\mathcal{V}_i| \leq 2^{\binom{m/k}{2}} = 2^{O(1/\epsilon^2)}$. Hence, computing all $k+1$ sets $\mathcal{V}_i$ ($i \in \{0, \dots, k\}$) in step 5 takes $(k+1) \cdot 2^{O(1/\epsilon^2)} = O(m \cdot \epsilon \cdot 2^{O(1/\epsilon^2)})$ time.

Similarly, each block B_i has at most $2\frac{m}{k}$ step edges, hence we can compute $\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)$ by using $|\mathcal{V}_{i-1}| \cdot |\mathcal{V}_i|$ times Reduction 8.5 to Min Edge Cover on a graph on $2\frac{m}{k} = O\left(\frac{1}{\epsilon}\right)$ vertices (and therefore a number of edges at most quadratic in this order). By Proposition 8.6, each call to the algorithm for Min Edge Cover then takes $O\left(\frac{1}{\epsilon^{5/2}}\right)$ time. Hence, computing all sets $\mathcal{F}_i(\mathcal{S}_1, \mathcal{S}_2)$ in step 9 takes $O(k \cdot 2^{O(1/\epsilon^2)} \frac{1}{\epsilon^{5/2}}) = O\left(m \frac{2^{O(1/\epsilon^2)}}{\epsilon^{3/2}}\right)$ time.

Computing the digraph $\overrightarrow{D}$ in step 12 takes $O(k \cdot 2^{O(1/\epsilon^2)}) = O(m\epsilon 2^{O(1/\epsilon^2)})$ time since $\overrightarrow{D}$ has at most $(k+1)2^{O(1/\epsilon^2)} + 2$ vertices and all the weights have been computed in step 9.

Finally, finding a shortest path from s to t in $\overrightarrow{D}$ at step 13 takes time $O(|V(\overrightarrow{D})| + |A(\overrightarrow{D})|)$ (see e.g. [16, Theorem 2.3.4]), that is, $O\left(m^2\epsilon^2 2^{O(1/\epsilon^2)^2}\right) = O\left(m^2\epsilon^2 2^{O(1/\epsilon^2)}\right)$.

Hence in total, we have a running time of $O\left(2^{O(1/\epsilon^2)}\left(\frac{m}{\epsilon^{3/2}} + \epsilon^2 m^2\right)\right)$ (from steps 5 and 13).

At this point, let us prove the correctness of the algorithm. At step 3 of the algorithm, for each of the k step edges $a_1, \ldots, a_k$, we add two cycles to our solution; they account for $2k \leq \frac{2m\epsilon}{4} \leq \epsilon \cdot OPT(L_m)$. Indeed, we have $OPT(L_m) \geq \frac{m}{2}$ since each cycle can cover at most two step edges of L_m, and there are m step edges. The cycles that we have already considered cover all side edges within all the blocks. Hence it remains to prove that all step edges and all side edges from the block neighbourhoods are covered by the computed solution.

We remark that in *any* solution to MIN LADDER CYCLE COVER, the step edges of each neighbourhood block N_i can *only* be covered by the edges of some set of $\mathcal{V}_i$. This implies that in any solution, for each block B_i, some set $\mathcal{S}_1$ from $\mathcal{V}_{i-1}$ and some set $\mathcal{S}_2$ from $\mathcal{V}_i$ will be subsets of this solution. Moreover, the step edges of any block B_i can only be covered by cycles from B_i or from the two neighbouring neighbourhood blocks. Hence, given the two sets $\mathcal{S}_1 \in \mathcal{V}_{i-1}$ and $\mathcal{S}_2 \in \mathcal{V}_i$, computing a set of cycles using Reduction 8.5 to MIN EDGE COVER in step 5 is optimal. Hence, choosing a solution among all sequences from $\mathcal{V}_0 \times \ldots \times \mathcal{V}_k$ will provide a solution of size at most $OPT(L_m)$ that will cover all edges of the neighbourhood blocks and all step edges of the blocks. This is exactly what is done in steps 13 and 14.

Hence in total we obtain that $|\mathcal{C}| \leq (1+\epsilon)OPT(L_m)$ and $\mathcal{C}$ is a valid cycle cover of L_m. Furthermore, the algorithm runs in quadratic time $O(m^2)$ when ϵ is fixed. This completes the proof. ☆

By Reduction 8.3 and Proposition 8.4, we get the following immediate corollary:

Corollary 8.8. MIN ID CODE *is in PTAS when restricted to unit interval graphs.*

8.2 EDGE-IDENTIFYING CODE for graphs of bounded tree-width

We have seen that IDENTIFYING CODE can be solved in linear time when the instances are restricted to graphs of given tree-width or clique-width (see Proposition 2.48 and Corollary 2.49).

We can use Corollary 2.49 in the context of EDGE-IDENTIFYING CODE as well:

Corollary 8.9. EDGE-IDENTIFYING CODE *can be solved in linear time for trees.*

Proof. It is known that *block graphs* (which are those graphs for which every bi-connected component is a clique) have clique-width at most 3 [92]. Observe that the class of line graphs of trees is exactly the class of claw-free block graphs. Since EDGE-IDENTIFYING CODE is the same problem as IDENTIFYING CODE when restricted to line graphs, the result follows from Corollary 2.49. ☆

We can however use the same ideas than for Proposition 2.48 in order to extend Corollary 8.9.

Proposition 8.10. *Given a graph G and an integer k, let $\mathcal{EID}(G,k)$ be the property that $\gamma^{EID}(G) \leq k$. Property $\mathcal{EID}(G,k)$ can be expressed in MSOL(τ_2).*

Proof. For convenience, the graph is encoded as a set V of vertices, a set E of edges and two unary predicates $a, b : E \to V$ such that for each edge xy, either $a(xy) = x$ and $b(xy) = y$ or $b(xy) = x$ and $a(xy) = y$. We first define two auxiliary binary relations over $E \times E$: $\neq$ and $\mathcal{I}^*$, where $\neq$ is the difference relation and $\mathcal{I}^*$ is an extension of the incidence relation where edges are not necessarily distinct:

- $e \neq f := \big(a(e) \neq a(f) \wedge a(e) \neq b(f)\big) \vee \big(b(e) \neq a(f) \wedge b(e) \neq b(f)\big)$
- $e\mathcal{I}^* f := a(e) = a(f) \vee a(e) = b(f) \vee b(e) = b(f) \vee b(e) = a(f)$

Now we define the MSOL(τ_2) logic formula which expresses that the graph has an edge-identifying code of size at most k. Note that $|\mathcal{C}| \leq k$ is an MSOL(τ_2) operation when k is fixed.

$$\exists \mathcal{C}, \mathcal{C} \subseteq E, |\mathcal{C}| \leq k, \big(\forall e \in E, \exists f \in \mathcal{C} \wedge e\mathcal{I}^* f\big) \wedge$$
$$\Big(\forall e \in E, \forall f \in E, e \neq f, \exists g \in \mathcal{C}, \big((e\mathcal{I}^* g \wedge \neg(f\mathcal{I}^* g)) \vee (f\mathcal{I}^* g \wedge \neg(e\mathcal{I}^* g))\big)\Big)$$

☆

Corollary 8.11. EDGE-IDENTIFYING CODE *can be solved in linear time for all classes of graphs having their tree-width bounded by a constant. Equivalently,* IDENTIFYING CODE *can be solved in linear time in classes of line graphs of graphs having their tree-width bounded by a constant.*

Note that because of Theorem 2.12, Corollary 8.11 cannot be extended easily to graphs having their clique-width bounded by a constant; indeed, it seems not possible to express $\mathcal{EID}(G, k)$ in MSOL(τ_1) since EDGE-IDENTIFYING CODE deals with edges identifying edges.

8.3 A class of graphs for which IDENTIFYING CODE is in P but DOMINATING SET is NP-complete

An interesting question is whether there are classes of graphs for which the complexities of the decision problems IDENTIFYING CODE and DOMINATING SET differ. To our knowledge, such a result was not known in the literature. We have seen in Chapter 6.5 that this is the case, for example, for the class of co-bipartite graphs (where DOMINATING SET is trivially solvable, but MIN ID CODE is even log-APX-hard). Note that this class is included in the class of AT-free graphs, which itself is a subclass of the class of DSP graphs, in which DOMINATING SET is solvable in polynomial time (an $O(n^7)$ algorithm for DOMINATING SET for DSP graphs on n vertices is given in [136]).

In this section, we construct a somewhat artificial, but large, class of graphs for which the converse holds: DOMINATING SET is NP-complete, but IDENTIFYING CODE is solvable in polynomial time. We call these graphs *SC-graphs*. The name comes from the fact that the hardness of the problem follows from their similarity to instances of the SET COVER problem, from which they are built.

Definition 8.12. *A graph G is said to be an* SC-graph *if it can be built from a bipartite graph with parts S and T and an additional set S' with $|S'| = 2|S|$ such that:*

- *for each vertex x of S, there is a path x, u_x, v_x of length 2 starting at x with $u_x, v_x \in S'$, $deg_G(u_x) = 2$ and $deg_G(v_x) = 1$, and*
- *each vertex of T has a distinct neighbourhood within S, and this neighbourhood has at least two elements.*

An example of an SC-graph is pictured in Figure 8.5.

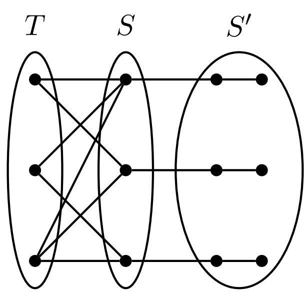


Figure 8.5: Example of an SC-graph.

Proposition 8.13. *Let G be an SC-graph built from a bipartite graph with parts S and T, with S_1, the set of all degree 1-vertices of the pendant paths attached to the vertices of S. We have $\gamma^{ID}(G) = 2|S|$ and $S \cup S_1$ is an identifying code of G.*

Proof. Note that all vertices of S are forced. Indeed, each vertex a of S is forced by the two vertices of the path of length 2 attached to a. Moreover, each vertex of degree 1 in each pendant path of length 2 must be dominated, hence $\gamma^{\text{ID}}(G) \geq 2|S|$. To see that $S \cup S_1$ is an identifying code, observe that each vertex a of S is identified by itself. Its neighbour of degree 2 in its pendant path of length 2 is identified by a and its neighbour in S_1. Each vertex of S_1 is only identified by itself. Finally, all vertices of T are identified by their sets of neighbours within S. Since these sets are distinct and of size at least two, $S \cup S_1$ is an identifying code. ☆

Corollary 8.14. *Let G be an SC-graph. Even if the parts S and T are not given, one can compute an optimal identifying code of G in polynomial time.*

Proof. To observe this, observe that the set S_1 of all degree 1-vertices of the pendant paths attached to the vertices of S can be detected easily: search for all vertices of degree 1 having a neighbour of degree 2. If G is an SC-graph, only vertices of S_1 satisfy this property, and it is easy to check whether G is indeed an SC-graph. Then, take as a code, this set S_1 together with each vertex at distance 2 of a vertex of S_1. By Proposition 8.13, this is an optimal identifying code of G. ☆

Theorem 8.15. Dominating Set *is* NP*-complete in planar (bipartite) SC-graphs of maximum degree 4.*

Proof. We reduce Set Cover to Dominating Set for SC-graphs. Let $(X, \mathcal{S})$ be an instance of Set Cover such that each vertex of $\mathcal{S}$ has a distinct neighbourhood within X and at least two neighbours in X. For example, one can take an instance of Vertex Cover for subcubic planar graphs, which is a special case of Set Cover (where X is the set of edges of a simple graph; each set of $\mathcal{S}$ stands for a given vertex and contains all edges incident to it), known to be NP-complete [87]. Let $\mathcal{B}(X, \mathcal{S})$ be the bipartite incidence graph of $(X, \mathcal{S})$, and build the SC-graph G from $\mathcal{B}(X, \mathcal{S})$ with parts $S = \mathcal{S}$ and $T = X$. If $(X, \mathcal{S})$ comes from Vertex Cover for subcubic planar graphs, G is planar and has maximum degree 4.

We claim that $(X, \mathcal{S})$ has a set cover of size k if and only if G has a dominating set of size $k + |\mathcal{S}|$. Let S_1 and S_2 be the sets of all degree 1 and degree 2-vertices of the pendant paths attached to the vertices of S, respectively.

For the first part, let $\mathcal{C} \subseteq \mathcal{S}$ be a set cover of $(X, \mathcal{S})$. One can easily check that the set $\mathcal{C} \cup S_2$ is a dominating set of G.

For the converse, let $\mathcal{D}$ be dominating set of G of size $k + |\mathcal{S}|$. Since each vertex of S_1 needs to be dominated, we have $|\mathcal{D} \cap (S_1 \cup S_2)| \geq |S| = |\mathcal{S}|$. We may in fact assume that $\mathcal{D} \cap (S_1 \cup S_2) = S_2$. We can also assume that $\mathcal{D} \cap T = \emptyset$, since by the previous observation, all vertices of S are dominated by some vertex of S_2: 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 $|\mathcal{D}'| \leq |S|$. Observe that $\mathcal{D}' \cap S$ has to cover all the vertices of T, hence $(X, \mathcal{S})$ has a set cover of size $|\mathcal{D}' \cap S| \leq |\mathcal{D}| - |S| = |\mathcal{D}| - |\mathcal{S}| = k$, which completes the proof. ☆

8.4 Conclusion

In this chapter, we showed in particular that the complexity of Min Id Code for unit interval graphs is tightly linked to the one of Min Ladder Cycle Cover. However the two following questions are still open:

Question 8.16. *What is the complexity of* Ladder Cycle Cover*?*

Question 8.17. *What is the complexity of* Identifying Code *when restricted to unit interval graphs?*

We also showed that MIN ID CODE admits a PTAS for unit interval graphs. We remark that this class is a subclass of the class of unit disk graphs. It is known that DOMINATING SET admits a PTAS in unit disk graphs [124], even when no geometric representation of the input graph is given [159]. So, we ask whether Corollary 8.8 can be extended to this class:[2]

Question 8.18. *Does* MIN ID CODE *admit a PTAS when restricted to unit disk graphs?*

The following weaker question is also open:

Question 8.19. *Is* MIN ID CODE *in APX when restricted to unit disk graphs?*

We showed that IDENTIFYING CODE is in P for SC-graphs, but DOMINATING SET is NP-complete in this class. This calls for the following question:

Question 8.20. *Is there another class of graphs for which* IDENTIFYING CODE *is polynomial-time solvable but* DOMINATING SET *is NP-hard?*

[2]Recall however, that IDENTIFYING CODE is NP-complete when restricted to (planar bipartite) unit disk graphs [158] (Theorem 2.44).

Chapter 9

General conclusion and perspectives

We have studied identifying codes from both combinatorial and algorithmic perspectives.

Regarding the combinatorial side, we have first studied graphs and digraphs having very large identifying code in Chapter 3. This study has led to the precise characterization of the family of finite graphs having as identifying code number their order minus one, as well as the characterization of all infinite graphs, finite digraphs and infinite oriented graphs having their order as identifying code number. These results answer several questions and conjectures from the literature.

We then investigated lower and upper bounds on the identifying code number of graphs of given maximum degree in Chapter 4. We have described all such graphs reaching the lower bound that was known in the literature. We then made the first known study of upper bounds for graphs of given maximum degree, which motivated the formulation of Conjecture 4.4. We gave several upper bounds supporting our conjecture, using two different techniques: the construction of special independent sets on the one hand, and the interplay between two techniques of the probabilistic method on the other hand (the Lovász Local Lemma and the Chenoff bound).

We then turned our attention to identifying codes of graphs belonging to specific graph classes in Chapter 5. We gave upper bounds in terms of the order and the minimum degree for graphs of girth at least 5, using both deterministic and probabilistic approaches and applying these bounds to random regular graphs. We also gave lower bounds on the identifying code number of interval graphs, before studying line graphs via edge-identifying codes, leading to several lower and upper bounds on their identifying code number. These classes of graphs had not been previously studied in the context of identifying codes.

For the algorithmic side, we have studied the computational complexity of the natural decision problem related to identifying codes, IDENTIFYING CODE. We have also studied the complexity of approximating its optimization counterpart, MIN ID CODE. These studies have been done for various graph classes, and we have tried to be as systematic as possible.

In particular, we have provided, in Chapter 6, new reductions of different kinds and exhibited several new graph classes such as bipartite graphs, co-bipartite graphs, split graphs and DSP graphs for which MIN ID CODE is NP-hard to approximate within a sub-logarithmic factor.

We then studied, in Chapter 7, graph classes where MIN ID CODE admits a constant factor approximation algorithm but admits no PTAS (such as bipartite graphs of small maximum degree, split graphs of small maximum CS-degree and line graphs). Our reductions also imply that IDENTIFYING CODE is NP-complete for several classes such as planar bipartite graphs of maximum degree 4, chordal bipartite graphs, or perfect planar line graphs of maximum degree 4. In this chapter, we have also shown that IDENTIFYING CODE remains NP-complete in interval graphs, a result that contrasts with the complexity of other related computational problems such as DOMINATING SET, which is linear-time solvable for interval graphs.

Finally, in Chapter 8, we have shown how to give an efficient PTAS approximation algorithm for MIN ID CODE for unit interval graphs by relating it to a special covering problem. There, we have also shown that IDENTIFYING CODE can be solved exactly in linear time for line graphs of graphs of bounded tree-width using Courcelle's theorem. We have also exhibited the first known graph class for which DOMINATING SET is computationally harder than IDENTIFYING CODE.

The studies of this thesis have raised many questions that merit to be investigated. We have mentioned most of them in the conclusions of the corresponding chapters; let us recall some of the most interesting ones.

- Can we solve Conjecture 4.4, i.e. does there exist a (small) constant c such that every identifiable graph G on n vertices has an identifying of size at most $n - \frac{n}{\Delta(G)} + c$? Can we prove a relaxed bound of the form $n - \frac{n}{\Theta(\Delta(G))}$? Can the conjecture be proved for some large classes of graphs such as subcubic graphs, trees or line graphs?

- We asked in Question 4.53 what is the complexity of deciding whether a given graph G reaches the lower bound of Theorem 2.29, i.e. whether $\gamma^{\text{ID}}(G) = \frac{2|V(G)|}{\Delta(G)+2}$. The same question can be asked for the bound of Theorem 2.24: can it be decided in polynomial-time whether for a given graph G on n vertices, the bound $\gamma^{\text{ID}}(G) = \lceil \log_2(n+1) \rceil$ holds?[1]

- Can we extend the lower bounds of the form $\gamma^{\text{ID}}(G) \geq \Omega\left(\sqrt{|V(G)|}\right)$ that hold for an interval or a line graph G to other graph classes?

- Can we fill the open cases and the cases that are not tight from Tables 1.3, 1.4, 1.5, 1.6 and 1.7 about lower and upper bounds of the identifying code number in specific graph classes? In particular, what is the tight lower bound on γ^{ID} for planar graphs or permutation graphs? What can we say about lower bounds on γ^{ID} for graphs of girth at least 5 and given minimum degree? What about tight upper bounds in the same class when the minimum degree is small?

- Can we fill the open and non-tight cases about the computational complexity of Identifying Code and Min Id Code from Tables 1.8 and 1.9? In particular, what is the precise complexity of Identifying Code for unit interval graphs and in permutation graphs? What is the complexity of approximating Min Id Code for interval graphs, (un)directed path graphs, strongly chordal graphs? Is there a PTAS for Min Id Code for planar graphs or for unit disk graphs?

In addition to these open problems, we also point out that many of the questions that have been answered in this thesis can be asked for similar identification parameters such as the locating-domination number, the identifying *open* code number or the metric dimension, which are related parameters which have also gained a lot of attention. Many combinatorial and algorithmic questions remain open for these parameters as well. For example, the complexity of determining the metric dimension of a planar graph was a long-standing open problem (solved only recently in [74] by showing its NP-completeness), and a recent paper studies the hardness of approximating this parameter [106]. Extending these studies to various classes of graphs, as done in this thesis for identifying codes, is a possible future line of research.

Finally, we also raise the question of the *parameterized complexity*[2] of the identifying code problem, that mostly remains unstudied (we point out that the parameterized complexity of the test cover problem according to four natural parameters has been recently investigated in [39]).

[1] We remark that for this question, a trivial brute-force quasi-polynomial-time algorithm exists: for each of the $\binom{n}{\lceil \log_2(n+1) \rceil}$ subsets of $\lceil \log_2(n+1) \rceil$ vertices of G, check whether it is an identifying code. This algorithm has time complexity $O\left(\binom{n}{\lceil \log_2(n+1) \rceil} n^c\right) = O\left(n^{O(\ln(n))} n^c\right) = O\left(2^{O\left(\ln(n)^2\right)} n^c\right)$, where c is some constant. Hence, the NP-completeness of this problem would imply that 3-SAT can be reduced in polynomial time to it. This would imply the existence of a quasi-polynomial-time algorithm for 3-SAT and therefore violate the well-known *Exponential Time Hypothesis* from [125] (a conjecture implying P$\neq$NP and which states that for any constant c there is no $2^{o(n)} n^c$ algorithm for 3-SAT, where n is the number of variables of the input formula). However, the aforementioned problem is clearly in the class LOGSNP, a subclass of NP defined in [163] whose problems have quasi-polynomial-time algorithms. Maybe one can show that our problem is LOGSNP-complete.

[2] In the framework of parameterized complexity, the algorithmic complexity of a decision problem is not only studied with the size of the input as a parameter, but also according to one or several other parameters that can be measured either on the instance or on the solution (for example, the solution size, or a given graph parameter). This allows for a finer classification than the binary classification of classical decision problems.

Appendix A
Appendix: missing proofs

In this appendix, we gather some proofs that are of minor relevance, repetitive, or that show results from the literature whose proof is unaccessible.

A.1 Proof of Lemma 4.15

Lemma A.1 (Lemma 4.15 [22])**.** *If G is an identifiable graph (infinite or not) not containing A_∞^+ as an induced subgraph, then for every vertex x of G, there is a vertex $y \in N[x]$ such that $G - y$ is identifiable.*

Proof. By contradiction, suppose that x_1 is a vertex that fails the statement of the lemma. Then $G - x_1$ has a pair of twin vertices. We name them y_1 and y_2. Without loss of generality we assume that x_1 is adjacent to y_2 but not to y_1. Now, in $G - y_2$ we must have another pair u, u' of twin vertices. By Lemma 3.20, $x_1 \in \{u, u'\}$, we name the other element x_2 ($x_2 \in N[x_1]$). Note that the subgraph induced on x_1, x_2, y_1, y_2 is isomorphic to A_2. We prove by induction that A_∞^+ is an induced subgraph of G, thus obtaining a contradiction.

To this end suppose A_k on $\{y_1, \dots y_k, x_1, \dots, x_k\}$ is already built such that x_{k-1}, x_k are twins in $G - y_k$ and y_{k-1}, y_k are twins in $G - x_{k-1}$. Then $x_k \in N[x_1]$. Consider $G - x_k$. There must be a pair of twins and, by Lemma 3.20, y_k must be one of them. Let y_{k+1} be the other one. Since y_k and y_{k+1} are twins in $G - x_k$, then y_{k+1} is adjacent to $x_1, \dots, x_k$ and $y_1, \dots, y_k$, in particular $y_{k+1} \in N[x_1]$. Now, there must be a pair of twins in $G - y_{k+1}$ and again by Lemma 3.20 one of them must be x_k, let the other one be x_{k+1}. Since x_k and x_{k+1} are twins in $G - y_{k+1}$, then x_{k+1} is adjacent to $x_1, \dots, x_k$ and not adjacent to $y_1, \dots, y_k$. Thus the graph induced on $\{y_1, \dots, y_{k+1}, x_1, \dots, x_{k+1}\}$ is isomorphic to A_{k+1} with the property that x_k,x_{k+1} are twins in $G - y_{k+1}$ and y_k,y_{k+1} are twins in $G - x_k$. Since this process does not end, we find that A_∞^+ is an induced subgraph of G. ☆

A.2 Proof of Theorem 4.28

We can use the idea of Subsection 4.3.2 to get improved bounds depending on the order and the maximum degree for the class of quasi-line graphs. Recall that a graph G is *quasi-line* if each closed neighbourhood of G can be partitioned into two (not necessarily disjoint) cliques. We use the following improvement of Lemma 4.26:

Lemma A.2. *Let G be an identifiable quasi-line graph of order n without isolated vertices and with maximum degree Δ having $n \cdot NF(G)$ non-forced vertices. Then, there exists an independent set I of G fulfilling the three properties of Proposition 4.25 and having size at least $\frac{n \cdot NF(G)}{2\Delta^2+3\Delta-2}$.*

Proof. The proof starts in the same way as for the proof of Lemma 4.26: we first greedily build an independent set $I_0 \subseteq N$ (where N is the set of non-forced vertices in G). We refer to the proof of Lemma 4.26 for the description of the greedy procedure. Note that since G is quasi-line (and therefore induced claw-free), each vertex can have at most one false twin. Hence, in each step of the procedure, we remove at most $\Delta+2$ vertices from the candidate set and $|I_0| \geq \frac{|N|}{\Delta+2}$.

Now, as in the proof of Lemma 4.26, we build the auxiliary graph G' on vertex set I_0 and construct an independent set $I \subseteq I_0$ of G' which will also be an independent set of G and which fulfills all required properties. Now, we claim that $\Delta(G') \leq 2\Delta$.

Before proving our claim, we point out that each pair u, v of adjacent vertices with $N[u] \ominus N[v] \subseteq I_0$ is such that $N[u] \ominus N[v] = \{x, y\}$ with $x \sim u$ and $y \sim v$. Moreover, the set of vertices of $N[u] \ominus N[v]$ that are adjacent to u form an independent set together with v; since a quasi-line graph is also induced claw-free, the neighbourhood $N(u)$ may only induce an independent set of size at most two, hence there is at most one vertex of $N[u] \ominus N[v]$ adjacent to u. A symmetric argument holds for v. Since we have $|N[u] \ominus N[v]| \geq 2$ (there is no forced vertex in I_0), this proves our claim.

Now, let x be a vertex of I_0 and let $Y = S \cup T$ be the set of all vertices s, t of G such that x belongs to $N[s] \ominus N[t]$ and $x \sim t$ ($s \in S$ and $t \in T$). We note that S and T are not necessarily disjoint. We claim that this set is the union of two cliques. Since $T \subseteq N(x)$ and G is quasi-line, T is the union of two cliques, T^1 and T^2. Let $s, t \in S \times T$ be a pair of vertices in Y; we may assume that $t \in T^1$. By the observation of the previous paragraph, x is the only vertex adjacent to t in $N[s] \ominus N[t]$. Hence s is adjacent to all other neighbours of t, in particular to all vertices in T^1. This proves our claim.

Now, we have that x is joined by an edge to some vertex y in G' only if $\{x, y\} = N[s] \ominus N[t]$ for some s, t from Y. But for each vertex s of S, there can be at most one such vertex playing the role of y. Indeed, suppose there are two vertices $y, y' \in I_0$ such that $y \in N[s] \ominus N[t]$ and $y' \in N[s] \ominus N[t']$. Then y, y' are not adjacent but since G is quasi-line $N(s)$ is the union of two cliques, one of them including either T^1 or T^2, say T^1. Hence, one of y, y' (say y) must be adjacent to all vertices of T^1 (including t) — a contradiction. Therefore, for each vertex s of S, x is joined in G' to at most one vertex y with $y \sim s$. Since $S \subseteq Y$ and Y induces two cliques, and some vertex of each of the two cliques has x as a neighbour, we have $|Y| \leq 2\Delta$. Moreover, x is only joined in G' to vertices that are at distance exactly 2 of x in G. Hence in each of the two cliques there is at least one vertex which does not contribute to the degree of x in G'. Summarizing, x has at most $2(\Delta-1)$ neighbours in G'.

Now, in the same way as in the proof of Lemma 4.26, we build a maximal independent set I of G' that has at size at least $\frac{|I_0|}{\Delta(G')+1} \geq \frac{|I_0|}{2\Delta-1}$.

To summarize, since $|I_0| \geq \frac{|N|}{\Delta+2}$, we have $|I| \geq \frac{n \cdot NF(G)}{2\Delta^2+3\Delta-2}$ and I fulfills all required properties. ☆

We are now ready to prove Theorem 4.28:

Theorem A.3 (Theorem 4.28)**.** *Let G be an identifiable quasi-line graph of order n without isolated vertices and with maximum degree Δ. Then $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^3+5\Delta^2+\Delta-2}$. If G has no forced vertices, $\gamma^{ID}(G) \leq n - \frac{n}{2\Delta^2+3\Delta-2}$.*

Proof. Let I be an independent set as constructed in Lemma 4.26. By Proposition 4.16, we have $NF(G) \geq \frac{1}{\Delta+1}$. Hence, we get $|I| \geq \frac{n}{2\Delta^2+3\Delta-2} \cdot \frac{1}{\Delta+1} = \frac{n}{2\Delta^3+5\Delta^2+\Delta-2}$. If G has no forced vertices, $NF(G) = 1$ and $|I| \geq \frac{n}{2\Delta^2+3\Delta-2}$. In both cases, Proposition 4.25 completes the proof. ☆

A.3 Proof of validity of codes C_a and C_b in Lemma 4.36

Claim A.4. *The sets $\mathcal{C}_a$ and $\mathcal{C}_b$ constructed in teh proof of Lemma 4.36 are (L, R)-quasi-identifying codes without $\mathcal{C}_a$-isolated or $\mathcal{C}_b$-isolated vertices.*

Proof. First of all, note that in both constructions, the final step consists in replacing some $\mathcal{C}_a$-isolated vertices from $\mathcal{C}_a$ (resp. $\mathcal{C}_b$). In order to simplify the proof, let $\mathcal{C}_a^*$ (resp. $\mathcal{C}_b^*$) be the code as it is before this last step. We first prove that $\mathcal{C}_a^*$ (resp. $\mathcal{C}_b^*$) have all desired properties except that there remain $\mathcal{C}_a^*$-isolated (resp. $\mathcal{C}_b^*$-isolated) vertices in L. We then prove that performing the last step transforms it into an (L, R)-quasi-identifying code with all required properties.

It can first be noticed that both $\mathcal{C}_a^*$ and $\mathcal{C}_b^*$ are dominating sets, so point number 1 of Definition 4.32 holds.

Let us now show point number 2 of Definition 4.32 (the separation condition). In both codes, the vertices of all pairs u, v of vertices of $L_1 \cup R_1$ are separated from each other, since $\mathcal{C}_1$ is a subset of both $\mathcal{C}_a^*$ and $\mathcal{C}_b^*$.

Now, suppose that $u \in R_1$ and $v \in L_2 \cup R_2$. By definition of R_1, no vertex of R_1 is adjacent to any vertex of $L_2 \cup R_2$. Therefore, by condition number 3 of Definition 4.32, either u or its neighbour in R_1 belong to $\mathcal{C}_1$, hence u and v are separated.

Thus, it remains to check if u and v are separated when $u \in L_1$ and $v \in L_2 \cup R_2$, and when both u and v belong to $L_2 \cup R_2$. We deal with $\mathcal{C}_a^*$ and $\mathcal{C}_b^*$ separately.

Code $\mathcal{C}_a^*$

- Suppose $u \in L_1$ and $v \in L_2 \cup R_2$. Note that u is dominated by some vertex x within $L_1 \cup R_1$ since $\mathcal{C}_1 \subseteq \mathcal{C}_a^*$. If $v \in L_2$, u and v are separated by x since no vertex of L_2 is adjacent to any vertex of $L_1 \cup R_1$. If $v \in R_2$ and $v \notin \mathcal{C}_a^*$, then u and v are separated by the neighbour of v in R_2, which belongs to $\mathcal{C}_a^*$. Similarly, if u has a neighbour in R_1 belonging to $\mathcal{C}_1$, we are done. Otherwise, it means that $v \in \mathcal{C}_a^*$ and $u \in \mathcal{C}_1$ (otherwise u would not be dominated by $\mathcal{C}_1$). Hence v has another neighbour in L, say u', belonging to $\mathcal{C}_a^*$, and u' separates u from v. Indeed, at step 4 of the construction of $\mathcal{C}_a$, either v already had at least two neighbours in $L \cap \mathcal{C}_a^*$, or an additional one has been added.

- Now, suppose both u and v belong to $L_2 \cup R_2$.

  If both u and $v \in L_2$, they are separated since the whole set L_2, which is independent, belongs to $\mathcal{C}_a^*$.

  If both u and v belong to R_2 and they are not adjacent, they are separated since either themselves or their respective neighbours in R_2 belong to $\mathcal{C}_a^*$ by step 3 of its construction. Otherwise, for the same reason one of them (say u) belongs to the code. It is ensured in step 4 that at least one neighbour of u in L belongs to $\mathcal{C}_a^*$, therefore u and v are separated by this neighbour.

  If $u \in L_2$ and $v \in R_2$ and they are not adjacent, they are separated by u since the whole set L_2 belongs to $\mathcal{C}_a^*$. Otherwise, if $v \notin \mathcal{C}_a^*$, they are separated by the neighbour of v in R_2. Otherwise, again by step 4 of the construction v has a second neighbour in $L \cap \mathcal{C}_a^*$, separating them.

Code $\mathcal{C}_b^*$

- If $u \in L_1$ and $v \in L_2 \cup R_2$, u and v are separated by a neighbour of v belonging to R_2 since the whole set R_2 is in $\mathcal{C}_b^*$.

- Now, suppose $u, v \in L_2 \cup R_2$.

  If both u, v belong to L_2, and they have the same set of neighbours within R, we are done since they do not need to be separated (point number 2 of Definition 4.32). Otherwise, they are separated since all their neighbours within $L \cup R$ belong to R_2, and $R_2 \subseteq \mathcal{C}_b^*$.

If both u, v belong to R_2, u and v are separated by themselves if they are not adjacent. Otherwise, they are separated by a neighbour of one of them in $L \cap \mathcal{C}_b^*$, added at step 3 of the construction.

Finally, if $u \in R_2$ and $v \in L_2$, then u and v are either separated by u if u and v are not adjacent, or by the neighbour of u in R_2 otherwise.

Let us now check point number 3 of Definition 4.32, i.e. that for each pair of adjacent vertices in R, at least one of them belongs to the code. This is true for vertices of R_1 since $\mathcal{C}_1$ is an (L_1, R_1)-quasi-identifying code and therefore fulfills this condition. This is also ensured for vertices of R_2 at step 3 of the construction of $\mathcal{C}_a$ and at step 2 of the construction of $\mathcal{C}_b$.

Hence, we have shown that both $\mathcal{C}_a^*$ and $\mathcal{C}_b^*$ are (L, R)-quasi-identifying codes.

Moreover, there are no $\mathcal{C}_a^*$-isolated (resp. $\mathcal{C}_b^*$-isolated) vertices in R: there are no such vertices in R_1 by Lemma 4.35, and no such vertices in R_2 for $\mathcal{C}_a^*$ by step 4 of its construction, and for $\mathcal{C}_b^*$ as well since $R_2 \subseteq \mathcal{C}_b^*$.

As announced previously, we now have to deal with the last step of the constructions of both $\mathcal{C}_a$ and $\mathcal{C}_b$. It is easily observed that this step does not affect the domination property of both codes. Indeed, the former $\mathcal{C}_a$-,$\mathcal{C}_b$-isolated vertices themselves are now dominated by some neighbour. Moreover each of their neighbours belongs to R, and since $\mathcal{C}_a$ and $\mathcal{C}_b$ are (L, R)-quasi-identifying its own neighbour in R belongs to the code.

Let us prove that the separation condition is still satisfied by $\mathcal{C}_a$ and $\mathcal{C}_b$. Let $\mathcal{C}_x$ ($x \in \{a, b\}$) be the considered code and let $l \in L$ be a $\mathcal{C}_x$-isolated vertex which gets replaced in $\mathcal{C}_x$ by one of its neighbours in R, say r_l. The only vertices which might be affected by the modification, are vertices which were previously dominated by l, i.e. vertices of $N[l]$: assume, by contradiction, that $u \in N[l]$ is no longer separated from some vertex v.

If $u = l$, in $\mathcal{C}_x$, we have $N[l] \cap \mathcal{C}_x = \{r_l\}$. Since $N[v] \cap \mathcal{C}_x = \{r_l\}$ and the neighbour of r_l in R belongs to $\mathcal{C}_x$, $v \in L$. Moreover, observe that v was dominated by a vertex of $\mathcal{C}_x^*$, say v', and $v' \notin N[l]$ since l is $\mathcal{C}_x^*$-isolated. Hence, it means that v was also $\mathcal{C}_x^*$-isolated. But then, in the last step of the construction of $\mathcal{C}_x$, one of l and v, say l, has been considered first and replaced by r_l, leaving them separated by v', a contradiction.

Now, if u is a neighbour of l, $u \in R$ and the neighbour of u in R, call him u', belongs to $\mathcal{C}_x$ by construction. Since $\mathcal{C}_x^*$ is an (L, R)-quasi-identifying code, u' has a neighbour belonging to L and to the code. Hence u and u' are separated, $u \neq r_l$ and v must be a neighbour of u' not belonging to the code. Hence $u \in R_2$ since u' has degree at least 3. Moreover, $v \in L_2$; otherwise, since $\mathcal{C}_1 \subseteq \mathcal{C}_x$, v would be dominated within $\mathcal{C}_1$ and u, v would be separated — a contradiction. Now, if $\mathcal{C}_x = \mathcal{C}_a$, $v \in \mathcal{C}_a$, a contradiction. If $\mathcal{C}_x = \mathcal{C}_b$, $u \in \mathcal{C}_b$, a contradiction too. This completes the proof of the separation property.

Now, note that point number 3 of Definition 4.32 remains verified as no vertex of R is removed from neither $\mathcal{C}_a$ or $\mathcal{C}_b$ in the last step of their construction. Finally, observe that thanks to the last step of the constructions, there are no $\mathcal{C}_x$-isolated ($x \in \{a, b\}$) vertices in L anymore. Moreover, this step has not created any $\mathcal{C}_x$-isolated vertices in R. Indeed, the vertices which are added, did not belong to $\mathcal{C}_x^*$, and hence their neighbour in R did. This completes the proof of the validity of both constructions $\mathcal{C}_a$ and $\mathcal{C}_b$. ☆

A.4 Proof of Theorem 4.44

In order to prove Theorem 4.44, we first need the following lemma.

Lemma A.5. *Let G be a connected triangle-free graph on $n \geq 4$ vertices and of maximum degree $\Delta \geq 3$ such that each subgraph H of G has an independent set of size at least $f(\Delta)|V(H)|$. There exists an independent set S in G such that $S = S_1 \cup S_2$, S_1 and S_2 are disjoint, and the following properties hold:*

1. *For each vertex u of S, u has a false twin in G if and only if $u \in S_1$.*

2. S does not contain any pair of false twins.

3. For each vertex $u \in V(G)$ of degree 1, there exists a vertex at distance 2 of u which does not belong to S.

4. There exists an $\alpha \in [0,1]$ such that:

- $|S_1| \geq \frac{\alpha}{2\Delta} n$ *and* $|S_2| \geq (1-\alpha)\min\left\{\frac{1}{3}, f(\Delta)\right\} n$*;*
- *if G has no false twins,* $|S_1| \geq \frac{\alpha}{\Delta+1} n$*;*
- *if G has minimum degree at least 2,* $|S_2| \geq (1-\alpha) f(\Delta) n$*.*

Proof. The proof of this lemma is algorithmic, in the sense that we propose an algorithm (Algorithm A.1) which builds S, S_1 and S_2. This algorithm uses the construction from Lemma 4.34.

Algorithm A.1 Greedy construction of the special independent set $S = S_1 \cup S_2$

Input: a connected graph $G = (V, E)$ on at least four vertices
1: $X \leftarrow V$, $S_1 \leftarrow \emptyset$, $S_2 \leftarrow \emptyset$, $P \leftarrow \emptyset$, $Q \leftarrow \emptyset$
2: **while** there exists a vertex $s \in X$ having a false twin in G **do**
3: $\quad S_1 \leftarrow S_1 \cup \{s\}$
4: $\quad$ **if** s has degree 1 in G and has only one false twin s' in G **then**
5: $\quad\quad$ Find a vertex x at distance 2 of both s and s', $x \notin S_1$ {we will prove that such a vertex exists}
6: $\quad\quad$ $P \leftarrow \Big(N[s] \cup \{s', x\}\Big) \cap X$
7: $\quad$ **else**
8: $\quad\quad$ $P \leftarrow \Big(N[s] \cup \{t \in V \mid N(s) = N(t)\}\Big) \cap X$
9: $\quad$ **end if**
10: $\quad Q \leftarrow Q \cup P$
11: $\quad X \leftarrow X \setminus P$
12: **end while**
13: Compute independent set S_2 of $G[X]$ using the construction of Lemma 4.34
14: **return** $S = S_1 \cup S_2$

Let us describe Algorithm A.1 in more detail. The set X is the set of candidate vertices, i.e. the potential vertices to be put into either S_1 or S_2. In the beginning of the algorithm, $X = V(G)$. The algorithm contains two main parts. In the first part, we build independent set S_1 by picking only vertices having a false twin in G. For each such picked vertex s, we remove from X the ball of s together with all false twins of s (plus one additional vertex in a very special case). We denote by Q, the set of vertices which have been removed from X during this first part of the algorithm.

In the second part of Algorithm A.1, we have $X = V(G) \setminus Q$. We build independent set S_2 by applying the construction of Lemma 4.34 to $G[X]$.

Let us prove that vertex x always exists at line 5 of the algorithm. Vertex s has degree 1 in G, and a unique false twin s'. Let t be the unique common neighbour of s and s'. Since $n \geq 4$ and G is connected, t has a neighbour u. If $u \notin S_1$, we set $x = u$. Otherwise, u has a false twin u' which is also a neighbour of t, but does not belong to S_1. Hence we can set $x = u'$, and we are done.

It can be first noticed that S is an independent set: in the first part of Algorithm A.1, when picking a candidate vertex s from X to put into S_1, we remove (at least) $N[s]$ from X. Moreover, since we pick a vertex at most once and add it either to S_2 or to S_1 but not to both, these two sets are disjoint. Let us prove that the claimed properties of S hold.

The first and the second properties are ensured by the first step of Algorithm A.1: each time a vertex having some false twins is added to S, all its false twins are removed from the set X of candidates. Moreover after this step no vertices having a false twin remain in X.

Let us show that the third property holds. Let u be a vertex of degree 1 in G. If $u \in S_1$, its false twin is not in S_1 and we are done. If $u \in Q \setminus S_1$ (i.e. it has been removed from the set of candidates while computing S_1), either it has a false twin u' in S_1, or it is a neighbour of a vertex in S_1. In the former case, if u has at least two false twins, we are done since only one of them can belong to S_1. Otherwise, this means u' has only one false twin (u) and we have ensured that some vertex at distance 2 of both u, u' has neither been put into S_1 nor S_2 (line 5 of Algorithm A.1). Finally, if $u \in V(G) \setminus Q$, by Lemma 4.34 we know u has a vertex x at distance 2 which does not belong to S_2. Suppose x belongs to S_1. Then x has a false twin x' which does not belong to S, and x' is also at distance 2 from u, so we are done.

It remains to prove the last property of S. Set $Q \subseteq V(G)$ is the set of vertices removed from X when adding a vertex to S_1. The sets Q and $V(G) \setminus Q$ form a partition of $V(G)$ and there exists some $\alpha \in [0,1]$ such that $|Q| = \alpha n$ and $|V(G) \setminus Q| = (1-\alpha)n$.

Now, we claim that $|Q| \leq 2\Delta|S_1|$. Indeed, let s be a vertex which is put into S_1 and consider the step where s has been added to S_1. If s is of degree 1 and has only one false twin, at most four vertices are removed from X and added to Q. Otherwise, at most the vertices of the closed neighbourhood of s and the set of its false twins are removed from X and added to Q. This set of vertices has at most $\Delta + 1 + \Delta - 1 = 2\Delta$ elements. Since G is connected and $n \geq 4$, $\Delta \geq 2$ and $\max\{4, 2\Delta\} = 2\Delta$. Hence the claim follows, and we have $|S_1| \geq \frac{\alpha}{2\Delta}n$. Moreover, if G has no false twins, a similar argument shows that $|Q| \leq (\Delta+1)|S_1|$.

Similarly, $|S_2| \geq \frac{(1-\alpha)(\ln\Delta - 1)}{2\Delta}n$. Indeed, we build S_2 by applying Lemma 4.34 on $G[V(G)\setminus Q]$, which has $(1-\alpha)n$ vertices. Since $G[V(G) \setminus Q]$ has no false twins, we obtain the two bounds (the general case and the case where G has minimum degree at least 2) by the second property of S_2 in Lemma 4.34. ☆

We can now use Lemma A.5 in order to prove Theorem 4.44. The proof is very similar to the proof of Theorem 4.37, except that we do not make a case distinction using the number of vertices having a false twin, and that we compute an independent set of G using Lemma A.5. Therefore we only sketch the main steps of the proof.

Theorem A.6 (Theorem 4.44). *Let G be a nontrivial connected identifiable triangle-free graph on n vertices with maximum degree $\Delta \geq 3$ such that each subgraph H of G has an independent set of size at least $f(\Delta)|V(H)|$. Then $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, 9, \frac{3}{f(\Delta)}\right\}}$.*
If G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, 9, \frac{3}{f(\Delta)}\right\}}$.
If G has minimum degree at least 3, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{2\Delta, \frac{2}{f(\Delta)}\right\}}$; if moreover G has no false twins, $\gamma^{ID}(G) \leq n - \frac{n}{\max\left\{\Delta+1, \frac{2}{f(\Delta)}\right\}}$.

Proof. Let $S = S_1 \cup S_2$ the independent set of G computed using Algorithm A.1 of Lemma A.5. Like in the proof of Theorem 4.37, we compute the set of pairs u, v forming an isolated edge in $G[V(G) \setminus S]$. We observe that this set forms a strong induced matching M. Let $L = L(M)$ and $R = R(M)$.

We now partition $V(G)$ into $L \cup R$ and its complement. Like in the proof of Theorem 4.37, we build an (L,R)-quasi-identifying code C_1 using Lemma 4.36, such that $|L'| \geq \frac{|L|}{3}$, where $L' = (L \cup R) \setminus C_1$ (if G has minimum degree at least 3, $|L'| \geq \frac{|L|}{2}$). We also set $C_2 = (V(G) \setminus (L \cup R)) \setminus S$. Note that set S restricted to $V(G) \setminus (L \cup R)$ fulfills the properties needed in order to apply Proposition 4.29. Hence it follows that C_2 is a $\big(V(G) \setminus (L \cup R)\big)$-identifying code of G.

Since S does not contain any pair of false twins, we can apply Proposition 4.33: $C = C_1 \cup C_2$ is an identifying code of G.

We now claim that $L \subseteq S_2$. Indeed, if some vertex l of L has a false twin l', by construction of S, $l' \notin S$. Let r be a neighbour of l in R, and let r' be the neighbour of r' in R. If $l' \neq r'$, r, r' are not an isolated edge in $G[V(G) \setminus S]$, a contradiction. But then $l' = r'$ and l' has at least one additional neighbour l'' in L. But since l, l' are false twins, l' and l'' are adjacent, a contradiction.

By our construction, we have $V(G)\setminus C = S_1 \cup (S_2 \setminus L) \cup L'$. Hence by Lemma A.5 and since in the general case $|L'| \geq \frac{|L|}{3}$ and $L \subseteq S_2$, for some $\alpha \in [0,1]$ we have:

$$\begin{aligned}
|V(G)\setminus C| &\geq |S_1| + \frac{|S_2|}{3} \\
&\geq \frac{\alpha}{2\Delta}n + (1-\alpha)\min\left\{\frac{1}{9}, \frac{f(\Delta)}{3}\right\}n \\
&\geq \min\left\{\frac{n}{2\Delta}, \frac{n}{\max\left\{9, \frac{3}{f(\Delta)}\right\}}\right\}
\end{aligned}$$

Hence $|C| \leq n - \frac{n}{\max\left\{2\Delta, 3, \frac{3}{f(\Delta)}\right\}}$.

Similar computations yield the other cases, taking into account that by Lemma A.5, when G has no false twins, $|S_1| \geq \frac{\alpha}{\Delta+1}n$, and when G has minimum degree at least 3, by Lemmas 4.36 and A.5, $|L'| \geq \frac{|L|}{2}$ and $|S_2| \geq (1-\alpha)f(\Delta)n$. ☆

A.5 Proof of Theorem 5.5

Theorem A.7 (Theorem 5.5)**.** *Let $G \in \mathcal{G}(n,d)$, then w.h.p. all the dominating sets of G have size at least $\frac{\ln d - 2\ln\ln d}{d}n$.*

Proof. We will proceed by contradiction. Given a set of vertices $\mathcal{D}$ of size m, we will compute the probability that $\mathcal{D}$ dominates $Y = V(G)\setminus\mathcal{D}$. Recall that G has been obtained from the configuration model by selecting a random perfect matching of K_{nd}. Let $y \in Y$ fixed, then let $A_y = \{N(\mathcal{D}) \cap \{y\} \neq \emptyset\}$ be the event that y is dominated by $\mathcal{D}$. Its complementary event corresponds to the situation where none of the edges of the perfect matching of K_{nd} connects the points corresponding to y to the ones corresponding to any vertex of $\mathcal{D}$. Define $W_{\mathcal{D}} = \cup_{v\in\mathcal{D}} W_v$ as the set of cells corresponding to $\mathcal{D}$ in K_{nd}. Then for any $v \in W_{\mathcal{D}}$, the event B_v corresponds to the fact that v is not connected to any point in W_y. If $W_{\mathcal{D}} = \{v_1, \ldots, v_{md}\}$,

$$\begin{aligned}
\Pr(\overline{A_y}) &= \Pr(\cap_{v\in W_{\mathcal{D}}} B_v) \\
&= \Pr(B_{v_1})\Pr(B_{v_2} \mid B_{v_1})\ldots\Pr(B_{v_{md}} \mid \cap_{i=1}^{md-1} B_{v_i}) \\
&= \left(1 - \frac{d}{nd-1}\right)\left(1 - \frac{d}{nd-3}\right)\cdots\left(1 - \frac{d}{nd-(2md-1)}\right) \\
&= \prod_{i=1}^{md}\left(1 - \frac{d}{nd-(2i-1)}\right) \\
&\geq \prod_{i=1}^{md}\left(1 - \frac{1}{n-2m}\right)
\end{aligned}$$

Since $1 - x = e^{-x+(\ln(1-x)+x)}$ (here we take $x = \frac{1}{n-2m}$) and $\ln(1-x)+x = O(x^2)$ (by the Taylor expansion of the logarithm in $x=0$), we obtain:

$$\begin{aligned}
\Pr(\overline{A_y}) &\geq \exp\left\{-\sum_{i=1}^{md}\frac{1}{n-2m} + O\left(\frac{1}{(n-2m)^2}\right)\right\} \\
&= \exp\left\{-(1+o(1))\frac{md}{n-2m}\right\}
\end{aligned}$$

The probability that $\mathcal{D}$ is dominating all vertices of $Y = \{y_1, \ldots, y_{n-m}\}$ is:

$$\Pr\left(\cap_{y\in Y} A_y\right) = \Pr\left(A_{y_1}\right)\Pr\left(A_{y_2} \mid A_{y_1}\right)\ldots\Pr\left(A_{y_{n-m}} \mid \cap_{j=1}^{n-m-1} A_{y_j}\right).$$

We claim that $\Pr\left(A_{y_i} \mid \cap_{j=1}^{i-1} A_{y_j}\right) \leq \Pr\left(A_{y_i}\right)$. Suppose that $y_1, \ldots, y_{i-1}$ are dominated. This means that the corresponding perfect matching of K_{nd} has an edge between one of the points corresponding to y_j $(1 \leq j \leq i-1)$ and one of the points corresponding to the vertices of $\mathcal{D}$. The probability that y_i is not dominated by $\mathcal{D}$ is now the probability that none of the remaining edges of the perfect matching connect any vertex of $\mathcal{D}$ with y_i. Hence:

$$\begin{aligned}
\Pr\left(\overline{A_{y_i}} \mid \cap_{j=1}^{i-1} A_{y_j}\right) &= \left(1 - \frac{d}{nd-2i+1}\right)\left(1 - \frac{d}{nd-2i-1}\right)\cdots\left(1 - \frac{d}{nd-2md+1}\right)\\
&\geq \left(1 - \frac{d}{nd-1}\right)\left(1 - \frac{d}{nd-3}\right)\cdots\left(1 - \frac{d}{nd-2md+1}\right)\\
&= \Pr(\overline{A_{y_i}})
\end{aligned}$$

By considering the complementary events, $\Pr\left(A_{y_i} \mid \cap_{j=0}^{i-1} A_{y_j}\right) \leq \Pr\left(A_{y_i}\right)$. Hence these events are negatively correlated, and:

$$\Pr\left(\cap_{y \in Y} A_y\right) \leq \prod_{i=1}^{n-m} \Pr(A_{y_i}) \leq \left(1 - e^{-\frac{md}{n-2m}}\right)^{n-m} \leq \exp\left\{-(n-m)e^{-\frac{md}{n-2m}}\right\}.$$

For the sake of contradiction, let $m \leq \frac{\ln d - c \ln\ln d}{d} n$ for some $c > 2$. Then:

$$\begin{aligned}
\Pr\left(\cap_{y \in Y} A_y\right) &\leq \exp\left\{-\left(1 - \frac{\ln d - c\ln\ln d}{d}\right) n \exp\left\{-\frac{\ln d - c\ln\ln d}{1 - 2\frac{\ln d - c\ln\ln d}{d}}\right\}\right\}\\
&= \exp\left\{-\left(1 + o_d(1)\right) n \exp\left\{-\frac{\ln d - c\ln\ln d}{1 + o_d(1)}\right\}\right\}\\
&= (1 + o_d(1)) e^{-\frac{(\ln d)^c}{d} n}
\end{aligned}$$

Note that if no set of size m dominates Y, neither will do a smaller one. So we have to look just at the sets of size m. The number of these sets can be bounded by

$$\begin{aligned}
\binom{n}{m} &\leq \frac{n^m}{m!} \leq \left(\frac{en}{m}\right)^m = \left(\frac{de}{\ln d - c\ln\ln d}\right)^{\frac{\ln d - c\ln\ln d}{d} n}\\
&= (1 + o_d(1))\left(\frac{de}{\ln d}\right)^{\frac{\ln d - c\ln\ln d}{d} n}
\end{aligned}$$

where we have used $m! \geq \left(\frac{m}{e}\right)^m$.

Let E_{DS} be the event that G has a dominating set of size m. Applying the union bound, we obtain:

$$\begin{aligned}
\Pr(E_{DS}) &\leq (1 + o_d(1))\left(\frac{de}{\ln d}\right)^{\frac{\ln d - c\ln\ln d}{d} n} e^{-\frac{(\ln d)^c}{d} n}\\
&= (1 + o_d(1)) \exp\left\{\frac{\ln d - c\ln\ln d}{d}(\ln d + 1 - \ln\ln d) n - \frac{(\ln d)^c}{d} n\right\}\\
&= (1 + o_d(1)) \exp\left\{\left(\frac{(\ln d)^2}{d} - \frac{(\ln d)^c}{d} + o_d\left(\frac{(\ln d)^2}{d}\right)\right) n\right\},
\end{aligned}$$

which tends to 0 when n tends to infinity since $c > 2$. This shows that w.h.p. no set of size less than $\frac{\ln d - 2\ln\ln d}{d} n$ can dominate the whole graph and completes the proof. ☆

A.6 Proof of Theorem 5.17

Theorem A.8 (Theorem 5.17). *We have* $\gamma^{EID}(K_n) = \begin{cases} 5, & \text{if } n = 4 \text{ or } 5 \\ n-1, & \text{if } n \geq 6 \end{cases}$. *Furthermore, let $\mathcal{C}_E$ be an edge-identifying code of K_n of size $n-1$ ($n \geq 6$) and let $G_1, G_2, \ldots, G_k$ be the connected components of $(V(K_n), \mathcal{C}_E)$. Then exactly one component, say G_i, is isomorphic to K_1 and every other component G_j ($j \neq i$) is isomorphic to a cycle of length at least 5.*

Proof. We note that $\mathcal{L}(K_4)$ is isomorphic to $K_6 \setminus M$, where M is a perfect matching of K_6. One can check that this graph has identifying code number 5. By a case analysis, we can show that K_5 does not admit an edge-identifying code of size 4. Indeed, since an edge-identifying code must be edge-identifiable, there are only two graphs possible for an edge-identifying code of this size: a path P_5 or a cycle C_4. In both cases, there are edges which are not separated. The edges of a C_5 form an edge-identifying code of size 5 of K_5, hence $\gamma^{\text{EID}}(K_5) = 5$. Furthermore, it is not difficult to check that the set of edges of a cycle of length $n-1$ ($n \geq 6$) identifies all edges of K_n. Thus we have $\gamma^{\text{EID}}(K_n) \leq n-1$. The fact that $\gamma^{\text{EID}}(K_n) \geq n-1$ follows from the second part of the theorem which is proved as follows.

Let $\mathcal{C}_E$ be an edge-identifying code of K_n of size $n-1$ or less ($n \geq 6$). Let $G' = (V(K_n), \mathcal{C}_E)$. Let $G_1, G_2, \ldots, G_k$ be the connected components of G'. Since G' has n vertices but at most $n-1$ edges, at least one component of G' is a tree. On the other hand we claim that at most one of these components can be a tree and that such tree would be isomorphic to K_1. Let G_i be a tree. First we show that $|V(G_i)| \leq 2$. If not, by Lemma 5.16 there is a vertex x of degree 1 in G_i with a neighbour u of degree 2. Let v be the other neighbour of u. Then the edges xv and uv are not identified. If $V(G_i) = \{x, y\}$ then for any other vertex u, the edges ux and uy are not separated. Finally, if there are G_i and G_j with $V(G_i) = \{x\}$ and $V(G_j) = \{y\}$, then the edge xy is not dominated by $\mathcal{C}_E$. Thus exactly one component of G', say G_1, is a tree and $G_1 \cong K_1$. This implies that $\gamma^{\text{EID}}(K_n) \geq n-1$. Therefore, $\gamma^{\text{EID}}(K_n) = n-1$ and, furthermore, each G_i, ($i \geq 2$), is a graph with a unique cycle.

It remains to prove that each G_i, $i \geq 2$ is isomorphic to a cycle of length at least 5. By contradiction suppose one of these graphs, say G_2, is not isomorphic to a cycle. Since G_2 has a unique cycle, it must contain a vertex v of degree 1. Let t be the neighbour of v in G_2 and let u be the vertex of G_1. Then the edges tv and tu are not separated by $\mathcal{C}_E$. Finally we note that such cycle cannot be of length 3 or 4, because C_3 is not edge-identifiable and in C_4, the two chords (which are edges of K_n) would not be separated. ☆

A.7 Proof of Corollary 5.28

Corollary A.9 (Corollary 5.28). *If G is an edge-identifiable graph on n vertices not isomorphic to K_4 or K_4^-, then $\gamma^{EID}(G) \leq 2n-5$.*

Proof. We first prove that if G is an edge-identifiable graph on n vertices not isomorphic to K_4, then $\gamma^{\text{EID}}(G) \leq 2n-4$. Let $\mathcal{C}_E$ be a minimal edge-identifying code and let G' be the subgraph induced by $\mathcal{C}_E$. Then, by Theorem 5.27, G' is 2-degenerate. Let $v_n, v_{n-1}, \ldots, v_1$ be a sequence of vertices of G' obtained by a process of eliminating vertices of degree at most 2. Since v_1 and v_2 can induce at most a K_2, we notice that there could only be at most $2n-3$ edges in G'. Furthermore, if there are exactly $2n-3$ edges in G', then $v_1v_2 \in \mathcal{C}_E$ and each vertex v_i, $3 \leq i \leq n$, has exactly two neighbours in $\{v_1, \ldots, v_{i-1}\}$. Hence, the subgraph induced by $\{v_1, v_2, v_3, v_4\}$ is isomorphic to K_4^-. Considering symmetries, there are three possibilities for the subgraph induced by $\{v_1, \ldots, v_5\}$ (recall that v_5 is of degree 2 in this subgraph): see Figure A.2 . In each of these three cases, the edge uv has both ends of degree at least 3. Thus, we can apply the argument used in the proof of Theorem 5.27 on G' and uv, showing that we have one of the four configurations of Figure 5.9. But none of them matches with the configurations of Figure A.2, a contradiction.

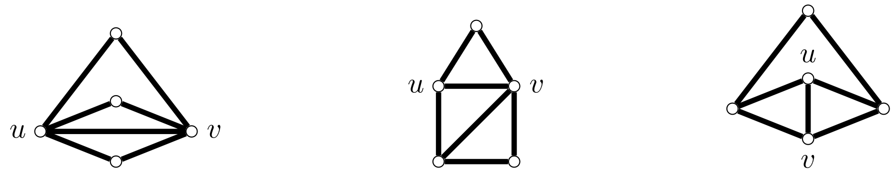


Figure A.2: The three maximal 2-degenerate graphs on five vertices.

Now we show that if $\gamma^{\text{EID}}(G) = 2n-4$, then $G \cong K_4^-$. This can be easily checked if G has at most four vertices, so we may assume $n \geq 5$. Let G'' be the subgraph of G' induced by $\{v_1, v_2, v_3, v_4, v_5\}$. If G'' has seven edges, then it is isomorphic to one of the graphs of Figure A.2, and we are done just like in the last case. Therefore, we can assume that G'' has exactly six edges and, since it is 2-degenerate, by an easy case analysis, it must be isomorphic to one of the graphs of Figure A.3.

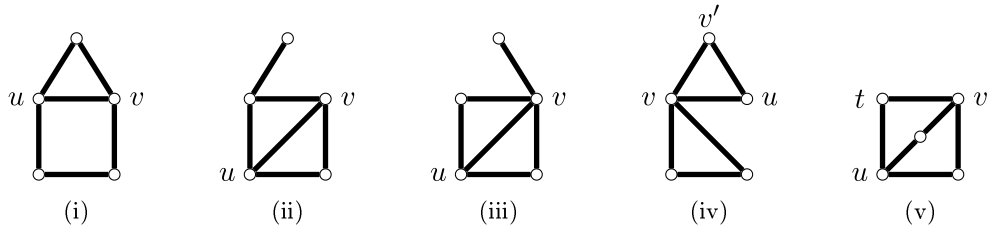


Figure A.3: The five possibilities of 2-degenerate graphs on five vertices with six edges.

If G'' is a graph in part (i), (ii) or (iii) of Figure A.3, then again one could repeat the arguments of the proof of Theorem 5.27 with G' and the edge uv of the corresponding figure, to obtain a contradiction.

Suppose G'' is isomorphic to the graph of Figure A.3(iv). Since G'' is not edge-identifiable, there must be at least one more vertex in G'. Let v_6 be as in the sequence obtained by the 2-degeneracy of G'. Since G' has exactly $2n-4$ edges, v_6 must have exactly two neighbours in G''. By the symmetry of the four vertices of degree 2 in G'', we may assume $uv_6 \in \mathcal{C}_E$. Then u and v are both of degree at least 3 in G'. Therefore, we could again repeat the argument of Theorem 5.27 with G' and uv, where only one of the configurations of this theorem, namely 5.9(d), matches G''. Furthermore, if this happens then $v'v_6$ should also be an edge of G'. Now u and v' are both of degree at least 3 and we apply the argument of Theorem 5.27 with G' and uv' to obtain a contradiction.

Finally, let G'' be isomorphic to the graph of Figure A.3(v). We claim that every other vertex v_i $(i \geq 6)$ is adjacent, in G', only to u and v. By contradiction suppose v_6 is adjacent to t. Then using the technique of Theorem 5.27 applied on G' and tu (respectively tv), we conclude that v_6 is adjacent to u (respectively v).

Since $|E(G')| = |\mathcal{C}_E| = 2n-4$, G' is a spanning subgraph of G. But then it is easy to verify that $\mathcal{C}_E \setminus \{xu, xv\}$ is an edge-identifying code of G — a contradiction. ☆

A.8 Proof of Theorem 7.10

We first need a few preliminary claims. For these claims, let G be a graph and G', the graph obtained from G using Reduction 7.9.

Claim A.10. *Let $\mathcal{D}$ be a dominating set of G. Using $\mathcal{D}$, one can build an identifying code of G' of size at most $|\mathcal{D}| + 3|V(G)|$.*

Proof. 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 apir

x, y of original vertices of G are separated by a_x and a_y, and x is seperated 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. ☆

Claim A.11. *Let $\mathcal{C}$ be an identifying code of G'. For each $x \in V(G)$, we have $|\mathcal{C} \cap \{b_x, c_x, d_x\}| \geq 2$ and $|\mathcal{C} \cap \{a_x, e_x\}| \geq 1$.*

Proof. For the first part, observe that vertices b_x, c_x and d_x are false twins, so $|\mathcal{C} \cap \{b_x, c_x, d_x\}| \geq 2$. Similarly, a_x and e_x are false twins, so $|\mathcal{C} \cap \{a_x, e_x\}| \geq 1$. ☆

Claim A.12. *Let $\mathcal{C}$ be an identifying code of G'. One can use $\mathcal{C}$ to build a dominating set of G of size at most $|\mathcal{C}| - 3|V(G)|$.*

Proof. By Claim A.11, for each $x \in V(G)$, we have $|\mathcal{C} \cap \{b_x, c_x, d_x\}| \geq 2$ and $|\mathcal{C} \cap \{a_x, e_x\}| \geq 1$. 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 $|\mathcal{D}| \leq |\mathcal{C}| - 3|V(G)|$. ☆

These claims are enough to give a proof of Theorem 7.10:

Theorem A.13 (Theorem 7.10)**.** Identifying Code *is NP-complete, even when restricted to chordal bipartite graphs.*

Proof. We apply Reduction 7.9 to the class of chordal bipartite graphs, for which Dominating Set is known to be NP-complete [157]. 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. Claims A.10 and A.12 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 $k + 3|V(G)|$, completing the proof. ☆

We can strengthen this result by showing that Reduction 7.9 applied to Min Dom Set restricted to graphs of maximum degree 3 is an L-reduction:

Theorem A.14. *Reduction 7.9 applied to graphs of maximum degree 3 is an L-reduction with parameters $\alpha = 13$ and $\beta = 1$. Therefore* Min Id Code *is APX-complete, even for bipartite graphs of maximum degree at most 5.*

Proof. Let G be a graph of maximum degree 3 and G' the graph constructed from G using Reduction 7.9. We have to prove Properties 1 and 2 from Definition 2.4.

First of all, observe that by Claim A.10, given an optimal dominating set $\mathcal{D}^*$ of G, we can construct an identifying code $\mathcal{C}$ with $\gamma^{\text{ID}}(G') \leq |\mathcal{C}| \leq |\mathcal{D}^*| + 3|V(G)| = \gamma(G) + 3|V(G)|$. Similarly, by Claim A.12, given an optimal identifying code $\mathcal{C}^*$ of G', we can construct a dominating set $\mathcal{D}$ of G such that $\gamma(G) \leq |\mathcal{D}| \leq |\mathcal{C}^*| - 3|V(G)| = \gamma^{\text{ID}}(G) - 3|V(G)|$. Hence we obtain:

$$\gamma^{\text{ID}}(G') = \gamma(G) + 3|V(G)|. \tag{A.1}$$

Property 1.

Since G has maximum degree 3, each vertex can dominate at most four vertices, hence we have $\gamma(G) \geq \frac{|V(G)|}{4}$, so $|V(G)| \leq 4\gamma(G)$. Using Equality (A.1), we get:

$$\gamma^{\text{ID}}(G') = \gamma(G) + 3|V(G)| \leq 13\gamma(G),$$

which proves Property 1 of Definition 2.4.

Property 2.

Let $\mathcal{C}$ be an identifying code of G'. Using Claim A.12 applied to $\mathcal{C}$, we obtain a dominating set

$\mathcal{D}$ with $|\mathcal{D}| \leq |\mathcal{C}| - 3|V(G)|$. By Equality (A.1), we have $-\gamma(G) = 3|V(G)| - \gamma^{\text{ID}}(G')$. So we obtain:

$$\begin{aligned} |\mathcal{D}| - \gamma(G) \leq& |\mathcal{C}| - 3|V(G)| + 3|V(G)| - \gamma^{\text{ID}}(G') \\ |\gamma(G) - |\mathcal{D}|| \leq& |\gamma^{\text{ID}}(G') - |\mathcal{C}||, \end{aligned}$$

which proves Property 2 of Definition 2.4.

For the second part of the statement, note that MIN DOM SET is known to be APX-complete, even for bipartite graphs of maximum degree 3 [53]. By construction, the graphs built from bipartite graphs of maximum degree 3 in Reduction 7.9 are bipartite and of maximum degree 5. ☆

A.9 Proofs from Section 7.3

Claim A.15 (Claim 7.29). *Let $\mathcal{C}_E$ be an edge-identifying code of G. One gets an identifying code $\mathcal{C}'_E$ with $|\mathcal{C}'_E| \leq |\mathcal{C}_E|$ by replacing $\mathcal{C}_E \cap E(P_G)$ by the three edges $\{\{b,c\},\{b,d\},\{d,e\}\}$.*

Proof. Edges $\{b,d\},\{d,e\}$ are forced. Once taking these, we need to separate $\{b,d\}$ from $\{d,e\}$, but it is sufficient to take edge $\{a,b\}$. Now one can check that all edges from $E(P_G)$ are correctly identified between each other and from all other edges of G. Moreover, edge $\{a,b\}$ is the only one from $E(P_G)$ which could possibly separate a pair of edges containing some edge from $E(G) \setminus E(P_G)$, hence $\mathcal{C}'_E$ is still an identifying code. Moreover by Claim 7.27 we have $|\mathcal{C}_E \cap E(P_G)| \geq 3$ hence $|\mathcal{C}'_E| \leq |\mathcal{C}_E|$. ☆

Claim A.16. *The edge-identifying code $\mathcal{C}(s)$ constructed in the proof of Claim 7.30 is valid.*

Proof. Let us show that $\mathcal{C}(s)$ is a valid identifying code of $G(X,\mathcal{Q},\lambda,\mu)$. In fact, a proof would be similar to many other proofs of this type (such as the one of Claim 7.14 for the reduction to split graphs of bounded CS-degree). It is however easy to check that all edges are dominated by $\mathcal{C}(s)$. Moreover, all edges are separated from each other; this can easily be seen for all pairs, besides the pair $\{\{q_0,q_1\},\{q_0,q_2\}\}$ in each clause gadget $G(Q_i,\lambda)$, whose proof of sepraration is detailed next. Note that this pair can only be separated by one of the edges $\{q_1,l_{i_1}^{2\lambda}\},\{q_2,l_{i_2}^{2\lambda}\},\{q_2,l_{i_3}^{2\lambda}\}$ of $G(Q_i,\lambda)$. If Q_i is satisified in s, there exists a true literal in it and hence, at least one edge among $\{q_1,l_{i_1}^{2\lambda}\},\{q_2,l_{i_2}^{2\lambda}\},\{q_2,l_{i_3}^{2\lambda}\}$ belongs to $\mathcal{C}(s)$. Otherwise, one of these edges belongs to the code due to the last step of the construction, completing the proof. ☆

Claim A.17 (Claim 7.31). *We have $|\mathcal{C}_E \cap (V(G(x_j,\mu)) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| \geq (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$.*

Proof. For each clause $Q_i \in \mathcal{Q}$, by Claim 7.27, counting each P-gadget of $G(Q_i,\lambda)$, we have $18\lambda + 3$ vertices inside all P-gadgets. Moreover, by Claim 7.28, edge $\{q_0,q_3\}$ is forced by the P-gadget attached at vertex q_3. Finally, by Claim 7.28 again, we need an edge incident to each of the vertices $l_{i_k}^{\ell}$, for $1 \leq k \leq 3$ and $1 \leq \ell \leq 2\lambda$. For this we need at least 3λ edges.

Similarly, for each variable $x_j \in X$, for each of the $4\mu - 3$ P-gadgets of $G(x_j,\mu)$, at least three edges of the gadget belong to the code. Moreover, each gadget forces a distinct edge to be in the code. ☆

Claim A.18 (Claim 7.32). *Let $x_j \in X$. We have $|\mathcal{C}_E \cap (E_j \cup A_j)| \geq \mu$. Moreover if $|\mathcal{C}_E \cap (E_j \cup A_j)| = \mu$, then either $|\mathcal{C}_E \cap (E_j \cup A_j)| = E_j^+$, or $|\mathcal{C}_E \cap (E_j \cup A_j)| = E_j^-$.*

Proof. The first part of the claim follows from the fact that for each k with $1 \leq k \leq 2\mu$, the pair of edges $\{a_k,b_k\},\{b_k,a_{(k \bmod 2\mu)+1}\}$ needs to be separated by some edge of $A_j \cup E_j$, and each such edge can separate at most two pairs. For the second part, observe that only edges of E_j can separate two pairs, and two consecutive edges of E_j separate only three pairs together. ☆

Claim A.19 (Claim 7.33). *Using $\mathcal{C}_E$, one can construct an edge-identifying code $\mathcal{C}'_E$ with $|\mathcal{C}'_E| \leq |\mathcal{C}_E|$ and such that for each variable $x_j \in X$, $|\mathcal{C}_E \cap (E_j \cup A_j)| \leq \mu + 1$.*

Proof. As noted in Claim 7.32, $|\mathcal{C}_E \cap (E_j \cup A_j)| \geq \mu$. Suppose $|\mathcal{C}_E \cap (E_j \cup A_j)| \geq \mu + 1$. Then we replace $\mathcal{C}_E \cap (E_j \cup A_j)$ by the set $\{\{a_1, x_j^1\}, \{a_2, \overline{x_j}^2\}, \{a_3, x_j^3\}, \{a_5, y_5\}, \ldots, \{a_{2\mu-1}, y_{2\mu-1}\}\}$. One can easily check that all edges from $E_j \cup A_j$ are separated, and since all three edges $\{a_1, x_j^1\}, \{a_2, \overline{x_j}^2\}, \{a_3, x_j^3\}$ (which are those which dominate edges from outside the variable gadget) belong to the code, $\mathcal{C}'_E$ is still an edge-identifying code. ☆

Claim A.20 (Claim 7.34). *Using $\mathcal{C}_E$, one can construct an edge-identifying code $\mathcal{C}'_E$ with $|\mathcal{C}'_E| \leq |\mathcal{C}_E|$ and $|\mathcal{C}_E \cap (V(G) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| = (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$.*

Proof. First of all, use Claim 7.29 on all P-gadgets of the graph. All edges that are forced by a P-gadget have to remain in $\mathcal{C}'_E$. Note that the remaining edges are edges $\{q_0, q_1\}$,$\{q_0, q_2\}$ and the ones incident to vertices $l_{i_k}^{\ell}$, for $1 \leq k \leq 3$ and $1 \leq \ell \leq 2\lambda$ (but not belonging to any P-gadget). We note that $\{q_0, q_1\}$ and $\{q_0, q_2\}$ need not to be part of $\mathcal{C}'_E$, so we remove them from the code. However, $\{q_0, q_1\}$ and $\{q_0, q_2\}$ are separated in $\mathcal{C}_E$ by one of the edges $\{q_1, l_{i_1}^{2\lambda}\}, \{q_2, l_{i_2}^{2\lambda}\}, \{q_2, l_{i_3}^{2\lambda}\}$. For each of these edges, say $\{q_1, l_{i_k}^{2\lambda}\}$, if it does not belong to $\mathcal{C}_E$, we replace $\mathcal{C}_E \cap \{\{q_1, l_{i_k}^{2\lambda}\}, \{l_{i_k}^{2\lambda}, l_{i_k}^{2\lambda-1}\}, \ldots, \{l_{i_k}^2, l_{i_k}^1\}\}$ by $\{\{l_{i_k}^{2\lambda}, l_{i_k}^{2\lambda-1}\}, \{l_{i_k}^{2\lambda-2}, l_{i_k}^{2\lambda-3}\}, \ldots, \{l_{i_k}^2, l_{i_k}^1\}\}$. If it does, we replace it by $\{\{q_1, l_{i_k}^{2\lambda}\}, \{l_{i_k}^{2\lambda-1}, l_{i_k}^{2\lambda-2}\}, \ldots, \{l_{i_k}^3, l_{i_k}^2\}\}$. In this last case, note that we may loose the property of being an identifying code since there might no longer be an edge of the code incident to the P-gadget P_k attached at vertex $l_{i_k}^1$. But if this was the case, then observe that we necessarily had $|\mathcal{C}_E \cap \{\{q_1, l_{i_k}^{2\lambda}\}, \{l_{i_k}^{2\lambda}, l_{i_k}^{2\lambda-1}\}, \ldots, \{l_{i_k}^2, l_{i_k}^1\}\}| \geq \lambda + 1$. Hence we still have room for at least one extra edge, and we add the edge incident to $l_{i_k}^1$ in the neighbouring variable gadget to $\mathcal{C}'_E$, solving the problem without making the code larger. ☆

Theorem A.21 (Theorem 7.38). *For any $\lambda \geq 1$ and $\mu \geq 2$, Reduction 7.25 is an L-reduction with parameters $\alpha = 51\mu + 201\lambda + 8$ and $\beta = 1$. Hence* Min Edge-Id Code *is APX-complete when restricted to bipartite graphs of maximum degree 3 and arbitrarily large girth, and* Min Id Code *is APX-complete when restricted to perfect line graphs of maximum degree 4.*

Proof. Having proved all the previous claims of this chapter, the proof is almost the same than the one of Theorem 7.19 for split graphs. Let $(X, \mathcal{Q})$ be an instance of Max (≤ 3,≤ 3)-SAT. First of all, we may assume that each variable x_i appears at least once as a positive literal, and at least once as a negative literal ($\overline{x_i}$). Indeed, otherwise it is easy to satisfy the clauses containing x_i and one may remove these clauses and x_i to get a smaller equivalent instance.

We have to prove Properties 1 and 2 from Definition 2.4.

Property 1.

Since each variable appears in at most three clauses, we have:

$$|\mathcal{Q}| \leq 3|X|. \tag{A.2}$$

Consider the truth assignment s with all variables "true". Since each variable x_i appears at least once as a positive literal, at least one clause is satisfied thanks to variable x_i. Since each clause contains at most three literals, we get that $OPT(X, \mathcal{Q}) \geq cost(s) \geq \frac{|X|}{3}$, that is:

$$|X| \leq 3 \cdot OPT(X, \mathcal{Q}). \tag{A.3}$$

Using Inequalities (A.2) and (A.3) together with Claim 7.30 with an optimal assignment of $(X, \mathcal{Q})$ having size $OPT(X, \mathcal{Q})$, we obtain:

$$\begin{aligned}
\gamma^{\text{ID}}(G(X, \mathcal{Q}, \lambda, \mu)) &\leq (17\mu - 12)|X| + (21\lambda + 5)|\mathcal{Q}| - OPT(X, \mathcal{Q}) \\
&\leq 3(17\mu - 12) \cdot OPT(X, \mathcal{Q}) + 9(21\lambda + 5) \cdot OPT(X, \mathcal{Q}) - OPT(X, \mathcal{Q}) \\
&= (51\mu + 201\lambda + 8) \cdot OPT(X, \mathcal{Q}),
\end{aligned}$$

which proves Property 1 of Definition 2.4.

Property 2.
Let $\mathcal{C}_E$ be an edge-identifying code of $G(X, \mathcal{Q}, \lambda, \mu)$ and $\mathcal{C}^*_E$, a minimum edge-identifying code of $G(X, \mathcal{Q}, \lambda, \mu)$, that is $|\mathcal{C}^*_E| = \gamma^{\text{ID}}(G(X, \mathcal{Q}, \lambda, \mu))$. We consider the code $\mathcal{C}'_E$ built using $\mathcal{C}_E$ and Claims 7.33 and 7.34. We also assume that $|\mathcal{C}^*_E \cap (V(G) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| = (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$ using Claim 7.34.

By Claims 7.32 and 7.33, for each variable $x_j \in X$, we have $\mu \leq |\mathcal{C}'_E \cap (E_j \cup A_j)| \leq \mu + 1$ and $\mu \leq |\mathcal{C}^*_E \cap (E_j \cup A_j)| \leq \mu + 1$. Hence $|\mathcal{C}'_E \cap \bigcup_{x_j \in X}(E_j \cup A_j)| = (\mu + \gamma)|X|$ and $|\mathcal{C}^*_E \cap \bigcup_{x_j \in X}(E_j \cup A_j)| = (\mu + \rho)|X|$ for some $\gamma, \rho \in [0, 1]$.

By Claim 7.31 and since $|\mathcal{C}'_E \cap (V(G) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| = |\mathcal{C}^*_E \cap (V(G) \setminus \bigcup_{x_j \in X}(E_j \cup A_j))| = (16\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}|$, we have $\gamma \geq \rho$ and:

$$|\mathcal{C}'_E| - |\mathcal{C}^*_E| = (\gamma - \rho)|X|. \tag{A.4}$$

Using Claim 7.35 with $\mathcal{C}'_E$, which has size $(17\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}| + \gamma|X|$, we can construct the truth assignment $s(\mathcal{C}'_E)$ of the variables of X such that:

$$cost(s(\mathcal{C}'_E)) \geq |\mathcal{Q}| - \gamma|X|. \tag{A.5}$$

Furthermore, we claim that the following holds:

$$OPT(X, \mathcal{Q}) \leq |\mathcal{Q}| - \rho|X|. \tag{A.6}$$

Indeed, suppose not. Then, there would be a truth assignment s^* of the variables of X satisfying strictly more than $|\mathcal{Q}| - \rho|X|$ clauses. But then by Claim 7.30 there would be an edge-identifying code of size at most $(17\mu - 12)|X| + (21\lambda + 5)|\mathcal{Q}| - cost(s^*) < (17\mu - 12)|X| + (21\lambda + 4)|\mathcal{Q}| + \rho|X| = |\mathcal{C}^*_E|$, a contradiction since $\mathcal{C}^*_E$ is a minimum edge-identifying code.

By combining Inequalities (A.5) and (A.6) and Equality (A.4), we get:

$$OPT(X, \mathcal{Q}) - cost(s(\mathcal{C}'_E)) \leq |\mathcal{Q}| - \rho|X| - (|\mathcal{Q}| - \gamma|X|) = |\mathcal{C}'_E| - |\mathcal{C}^*_E|,$$

which proves Property 2 of Definition 2.4.

Now, since Max (≤ 3,≤ 3)-SAT is APX-complete [162], the L-reduction and Corollary 7.20 show that Min Edge-Id Code is APX-complete, and equivalently, Min Id Code for line graphs is APX-complete. ☆

Bibliography

General references

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

[2] N. Alon and J. H. Spencer. *The probabilistic method*, 3rd edition, Wiley-Interscience, 2008.

[3] S. Arnborg, J. Lagergren and D. Seese. Easy problems for tree decomposable graphs. *Journal of Algorithms* 12:308–340, 1991.

[4] S. Arora and B. Barak. *Computational complexity: a modern approach*, Cambridge University Press, 2009.

[5] D. Auger. Induced paths in twin-free graphs. *The Electronic Journal of Combinatorics* 15:N17, 2008.

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

[7] D. Auger. *Problèmes d'identification combinatoire et puissances de graphes*. PhD thesis, ENST-Télécom ParisTech, France, 2010.

[8] D. Auger, I. Charon, I. Honkala, O. Hudry and A. Lobstein. Edge number, minimum degree, maximum independent set, radius and diameter in twin-free graphs. *Advances in Mathematics of Communications* 3(1):97–114, 2009.

[9] 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.

[10] D. Auger, I. Charon, O. Hudry and A. Lobstein. On the existence of a cycle of length at least 7 in a $(1, \leq 2)$-twin-free graph. *Discussiones Mathematicae Graph Theory* 30(4):591–609, 2010.

[11] D. Auger, I. Charon, O. Hudry and A. Lobstein. Watching systems in graphs: an extension of identifying codes. To appear in *Discrete Applied Mathematics*.

[12] D. Auger, I. Charon, O. Hudry and A. Lobstein. Maximum size of a minimum watching system and the graphs achieving the bound. To appear in *Discrete Applied Mathematics*.

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

[14] L. Babai and P. Frankl. *Linear algebra methods in combinatorics, preliminary version 2*, Department of Computer Science, The University of Chicago, 1992.

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

[16] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*, 2nd edition, Springer, 2009.

[17] L. W. Beineke. Characterizations of derived graphs. *Journal of Combinatorial Theory* 9(2)2:129–135, 1970.

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

[19] C. Berge. Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind. *Wissenschaftliche Zeitschrift der Martin-Luther-Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe* 10:114, 1961.

[20] 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.

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

[22] N. Bertrand. *Codes identifiants et codes localisateurs-dominateurs sur certains graphes.* Master thesis, ENST, France, June 2001.

[23] N. Bertrand, I. Charon, O. Hudry and A. Lobstein. Identifying and locating-dominating codes on chains and cycles. *European Journal of Combinatorics* 25(7):969–987, 2004.

[24] N. Bertrand, I. Charon, O. Hudry and A. Lobstein. 1-identifying codes on trees. *Australasian Journal of Combinatorics* 31:21–35, 2005.

[25] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors and Combinatorial Probability*, Cambridge University Press, 1986.

[26] B. Bollobás. *Random graphs*, 2nd edition, Cambridge University Press, 2001.

[27] J. A. Bondy. Induced subsets. *Journal of Combinatorial Theory, Series B* 12(2):201–202, 1972.

[28] J. A. Bondy and U. S. R. Murty. *Graph Theory*, 1st edition, Springer, 2008.

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

[30] M. Bouznif. *Complexité de problèmes combinatoires dans des graphes de type "grille" : modèles, algoritmes, saut de complexité.* PhD thesis, Université Joseph Fourier Grenoble I, France, 2012.

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

[32] R. L. Brooks. On colouring the nodes of a network. *Proceedings of the Cambridge Philosophical Society* 37:194–197, 1941.

[33] M. Burley and J. P. Uhry. Parity graphs. *Annals of Discrete Mathematics* 16:1–26, 1982.

[34] J. Cáceres, C. Hernando, M. Mora, I. M. Pelayo and M. L. Puertas. Locating dominating codes: bounds and extremal cardinalities. Manuscript, 2012 (`arXiv:1205.2177`).

[35] G. Chang and G. L. Nemhauser. The k-Domination and k-Stability Problems on Sun-Free Chordal Graphs. *SIAM Journal of Algebraic Discrete Methods* 5:332–345, 1984.

[36] 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.

[37] I. Charon, G. Cohen, O. Hudry and A. Lobstein. Discriminating codes in (bipartite) planar graphs. *European Journal of Combinatorics* 29(5):1353–1364, 2008.

[38] I. Charon, G. Cohen, O. Hudry and A. Lobstein. New identifying codes in the binary Hamming space. *European Journal of Combinatorics* 31(2):491–501, 2010.

[39] R. Crowston, G. Gutin, M. Jones, S. Saurabh and A. Yeo. Parameterized study of the test cover problem. *Proceedings of Mathematical Foundations of Computer Science, MFCS 2012*, *Lecture Notes in Computer Science* 7464:283–295, 2012.

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

[41] I. Charon, I. Honkala, O. Hudry and A. Lobstein. General bounds for identifying codes in some infinite regular graphs. *The Electronic Journal of Combinatorics* 8(1):R39, 2001.

[42] I. Charon, I. Honkala, O. Hudry and A. Lobstein. Structural properties of twin-free graphs. *The Electronic Journal of Combinatorics* 14(1):R16, 2007.

[43] I. Charon, O. Hudry and A. Lobstein. Identifying and locating-dominating codes: NP-completeness results for directed graphs. *IEEE Transactions on Information Theory* 48(8):2192–2200, 2002.

[44] 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.

[45] I. Charon, O. Hudry and A. Lobstein. Extremal cardinalities for identifying and locating-dominating codes in graphs. *Discrete Mathematics* 307(3-5):356–366, 2007.

[46] I. Charon, O. Hudry and A. Lobstein. Extremal values for the maximum degree in a twin-free graph. To appear in *Ars Combinatoria.*

[47] G. Chartrand, L. Eroh, M. Johnson and O. Oellermann. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics* 105(1–3):99–113, 2000.

[48] G. Chartrand, D. Erwin, M. A. Henning, P. J. Slater and P. Zhang. The locating-chromatic number of a graph. *Bulletin of the Institute of Combinatorics and its Applications* 36:89–101, 2002.

[49] C. Chen, C. Lu and Z. Miao. Identifying codes and locating-dominating sets on paths and cycles. *Discrete Applied Mathematics* 159(15):1540–1547, 2011.

[50] N. Chiba, T. Nishizeki, S. Abe and T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *Journal of Computer and System Sciences* 30(1):54–76, 1985.

[51] M. Chlebík and J. Chlebíková. Approximation hardness for small occurrence instances of NP-Hard problems. *Proceedings of the 5th Italian Conference on Algorithms and Complexity, CIAC 2003*, *Lecture Notes in Computer Science* 2653:152–164, 2003.

[52] M. Chlebík and J. Chlebíková. Approximation hardness of edge dominating set problems. *Journal of Combinatorial Optimization* 11:279–290, 2006.

[53] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation* 206:1264–1275, 2008.

[54] M. Chudnovsky, N. Robertson, P. Seymour and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics* 164:51–229, 2006.

[55] M. Chudnovsky and P. Seymour. The structure of claw-free graphs. *Surveys in combinatorics, London Mathematical Society Lecture Note Series* 327:153–171, 2005.

[56] M. Chudnovsky and P. Seymour. Excluding induced subgraphs. *Surveys in Combinatorics, London Mathematical Society Lecture Note Series* 346:99–119, 2007.

[57] B. N. Clark, C. J. Colbourn and D. S. Johnson. Unit disk graphs. *Discrete Mathematics* 86(1-3):165–177, 1990.

[58] A. Cobham. The intrinsic computational difficulty of functions. *Proceedings of the second International Congress for Logic, Methodology and Philosophy of Science*, Jerusalem, 1964.

[59] G. Cohen, I. Honkala, A. Lobstein and G. Zémor. On codes identifying vertices in the two-dimensional square lattice with diagonals. *IEEE Transactions on Computers* 50(2):174–176, 2001.

[60] G. Cohen, I. Honkala, A. Lobstein and G. Zémor. On identifying codes. *Proceedings of the DIMACS Workshop on Codes and Association Schemes '99*, pages 97-109, 2001.

[61] C. Colbourn, P. J. Slater and L. K. Stewart. Locating-dominating sets in series-parallel networks. *Congressus Numerantium* 56:135–162, 1987.

[62] S. Cook. The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71*, pages 151–158, 1971.

[63] D. G. Corneil and L. K. Stewart. Dominating sets in perfect graphs. *Discrete Mathematics* 86(1–3):145–164, 1990.

[64] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation* 85(1):12–75, 1990.

[65] B. Courcelle, J. Engelfriet and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computing and System Sciences* 46:218–270, 1993.

[66] B. Courcelle, J. Makowski and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique width. *Theory of Computing Systems* 33(2):125–150, 2000.

[67] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science* 109:49–82, 1993.

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

[69] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal of Computing* 23(4):864–894, 1994.

[70] P. Damaschke, H. Müller and D. Kratsch. Domination in convex and chordal bipartite graphs. *Information Processing Letters* 36:52:31–236, 1990.

[71] 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.

[72] E. D. Demaine and M. Hajiaghayi. Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. *Proceedings of the sixteenth annual ACM-SIAM symposium on discrete algorithms, SODA'05*, pages 682–689, 2005.

[73] H. N. De Ridder and others. Information System on Graph Classes and their Inclusions (ISGCI). `http://www.graphclasses.org`

[74] J. Díaz, O. Pottonen, M. Serna and E. J. Van Leeuwen. On the complexity of metric dimension. *Proceedings of the 20th annual European Symposium on Algorithms, ESA 2012*, *Lecture Notes in Computer Science* 7501:419–430, 2012.

[75] R. Diestel. *Graph Theory*, 4th edition, Springer, 2010.

[76] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics* 17:449–467, 1965.

[77] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal, R. Rado and V. T. Sós, editors. *Infinite and Finite Sets (to Paul Erdős on his 60th birthday). II.*, North-Holland, pages 609–627, 1975.

[78] P. Erdős and A. Rényi. On random graphs. I. *Publicationes Mathematicae* 6:290–297, 1959.

[79] P. Erdős and H. Sachs. Reguläre Graphen gegebener Taillenweite mit minimaler Knotenzahl. *Wissenschaftliche Zeitschrift der Martin-Luther-Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe* 12:251–258, 1963.

[80] L. Esperet, S. Gravier, M. Montassier, P. Ochem and A. Parreau. Locally identifying coloring of graphs. *The Electronic Journal of Combinatorics* 19(2):P40, 2012.

[81] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation*, ed. R. Karp, *SIAM-AMS Proceedings* 7:27–41, 1974.

[82] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics* 7:115–130, 1984.

[83] M. Farber and J. M. Keil. Domination in permutation graphs. *Journal of Algorithms* 6:309–321, 1985.

[84] J.F. Fink and M.S. Jacobson, n-domination in graphs. *Graph Theory with Applications to Algorithms and Computer Science*, John Wiley and Sons, 282–300, 1985.

[85] A. Frieze, R. Martin, J. Moncel, M. Ruszinkó and C. Smyth. Codes identifying sets of vertices in random networks. *Discrete Mathematics* 307(9-10):1094–1107, 2007.

[86] P. Gambette and S. Vialette. On restrictions of balanced 2-interval graphs. *Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG'07*, *Lecture Notes in Computer Science* 4769:55–65, 2007.

[87] 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.

[88] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[89] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory Series B* 16:47–56, 1974.

[90] J. Gimbel, B. D. Van Gorden, M. Nicolescu, C. Umstead and N. Vaiana. Location with dominating sets. *Congressus Numerantium* 151:129–144, 2001.

[91] O. Goldreich. *Computational Complexity: a conceptual perspective*, Cambridge University Press, 2008.

[92] M. C. Golumbic and U. Rotics. On the Clique-Width of Some Perfect Graph Classes. *International Journal of Foundations of Computer Science* 11(3):423–443, 2000.

[93] D. Gonçalves, A. Pinlou, M. Rao and S. Thomassé. The domination number of grids. *SIAM Journal of Discrete Mathematics* 25(3):1443–1453, 2011.

[94] 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.

[95] S. Gravier, M. Kovše, M. Mollard, J. Moncel and A. Parreau. New results on variants of covering codes in Sierpiński graphs. To appear in *Designs, Codes and Cryptography.*

[96] S. Gravier and J. Moncel. On graphs having a $V \setminus \{x\}$ set as an identifying code. *Discrete Mathematics* 307(3-5):432–434, 2007.

[97] S. Gravier, J. Moncel and A. Semri. Identifying codes of cycles. *European Journal of Combinatorics* 27(5):767–776, 2006.

[98] S. Gravier, J. Moncel and A. Semri. Identifying codes of Cartesian product of two cliques of the same size. *The Electronic Journal of Combinatorics* 15:N4, 2008.

[99] H. Grötzsch. Zur Theorie der diskreten Gebilde. VII. Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel (German). *Wissenschaftliche Zeitschrift der Martin-Luther-Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe* 8:109–120, 1958/1959.

[100] M. Grötschel, L. Lovász, A. Schrijver. Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics* 21:325–256, 1984.

[101] H. Hadwiger. Über eine Klassifikation der Streckenkomplexe. *Vierteljahrschriften der Naturforschungsgesellschaft Zürich* 88:133–143, 1943.

[102] R. Halin. S-functions for graphs. *Journal of Geometry* 8:171–186, 1976.

[103] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society* 10(1):26–30, 1935.

[104] F. Harary et R. A. Melter. On the metric dimension of a graph. *Ars Combinatoria* 2:191–195, 1976.

[105] A. Harutyunyan, P. Horn and J. Verstraete. Independent dominating sets in graphs of girth five. Submitted manuscript, 2009. Available online at `http://www.mathcs.emory.edu/~phorn/`.

[106] M. Hauptmann, R. Schmied and C. Viehmann. Approximation complexity of metric dimension problem. *Journal of Discrete Algorithms* 14:214–222, 2012.

[107] F. Havet. Minimum-density identifying codes in grids. Manuscript, 2010.

[108] T. W. Haynes, S. T. Hedetniemi and P. J. Slater. *Fundamentals of Domination in Graphs*, Marcel Dekker, 1998.

[109] T.W. Haynes, S. T. Hedetniemi and P. J. Slater, editors. *Domination in Graphs: Advanced Topics*, Marcel Dekker, 1998.

[110] 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.

[111] P. J. Heawood. Map-colour theorem. *Quarterly Journal of Pure and Applied Mathematics* 24:332–338, 1890.

[112] M. A. H. Henning and A. Yeo. Identifying open codes in cubic graphs. Submitted manuscript, 2012.

[113] M. A. H. Henning and A. Yeo. Identifying vertex covers in graphs. Submitted manuscript, 2012.

[114] D. Hochbaum (editor). *Approximation algorithms for NP-hard problems*, PWS Publishing Co. Boston, USA, 1997.

[115] H. Hocquard and M. Montassier. Adjacent vertex-distinguishing edge coloring of graphs with maximum degree Δ. To appear in *Journal of Combinatorial Optimization.*

[116] I. Honkala. On r-locating-dominating sets in paths. *European Journal of Combinatorics* 30(4):1022–1025, 2009.

[117] I. Honkala, M. Karpovsky and S. Litsyn. Cycles identifying vertices and edges in binary hypercubes and 2-dimensional tori. *Discrete Applied Mathematics* 129(2-3):409–419, 2003.

[118] I. Honkala, T. Laihonen and S. Ranto. On codes identifying sets of vertices in Hamming spaces. *Designs, Codes and Cryptography* 24, 193–204, 2001.

[119] I. Honkala, T. Laihonen and S. Ranto. On strongly identifying codes. *Discrete Mathematics* 254:191–205, 2002.

[120] I. Honkala and A. Lobstein. On identifying codes in binary Hamming spaces. *Journal of Combinatorial Theory Series A* 99(2):232–243, 2002.

[121] I. Honkala and A. Lobstein. On the complexity of the identification problem in Hamming spaces. *Acta Informatica* 38(11-12):839–845, 2002.

[122] J. Hopcroft and R. Tarjan. Efficient Planarity Testing. *Journal of the ACM* 21(4):549–568, 1974.

[123] J. Hromkovič. *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*, 2nd edition, Springer, 2002.

[124] 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.

[125] R. Impagliazzo, R. Paturi and F. Zane. Which problems have strongly exponential complexity?, *Journal of Computer and System Sciences* 63:512–530, 2001.

[126] S. Janson and T. Laihonen. Note: On the size of identifying codes in binary hypercubes. *Journal of Combinatorial Theory Series A* 116(5):1087–1096, 2009.

[127] D. S. Johnson, Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* 9:256–278, 1974.

[128] V. Junnila. *On identifying and locating-dominating codes*. PhD thesis, Turku Centre for Computer Science, Finland, 2011.

[129] V. Junnila and T. Laihonen. Optimal identifying codes in cycles and paths. *Graphs and Combinatorics* 28(4):469–481, 2012.

[130] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[131] 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.

[132] S. Khanna, R. Motwani, M. Sudan, U. V. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal of Computing* 28(1):164–191, 1998.

[133] S. Khuller, B. Raghavachari and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics* 70(3):217–229, 1996.

[134] S. Klavžar and U. Milutinović. Graphs $S(n,k)$ and a variant of the Tower of Hanoi problem. *Czechoslovak Mathematical Journal* 47(122):95–104, 1997.

[135] A. V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica* 4:307–316, 1984.

[136] D. Kratsch. Domination and total domination on asteroidal triple-free graphs. *Discrete Applied Mathematics* 99(1-3):111–123, 2000.

[137] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM* 22(1):155–171, 1975.

[138] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[139] M. Laifenfeld. *Coding for network applications: Robust identification and distributed resource allocation*. PhD thesis, Boston University, USA, 2008.

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

[141] 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.

[142] A. Lobstein. Watching systems, identifying, locating-dominating and discriminating codes in graphs: a bibliography. `http://www.infres.enst.fr/~lobstein/debutBIBidetlocdom.pdf`

[143] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics* 2:253–267, 1972.

[144] L. Lovász. *Combinatorial problems and exercises*, 2nd edition, Elsevier, 1993.

[145] L. Lovász. Graph minor theory. *Bulletin of the American Mathematical Society* 43:75–86, 2006.

[146] W. McCuaig and B. Shepherd. Domination in graphs with minimum degree two. *Journal of Graph Theory* 13(6):, 749–762, 1989.

[147] B. McKay and N. Wormald. Asymptotic enumeration by degree sequence of graphs with degrees $o(n^{1/2})$. *Combinatorica* 11(4):369–382, 1991.

[148] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|}|E|)$-algorithm for finding maximum matching in general graphs. *Proceedings of the 21st Annual Symposium on Foundations of Computer Science, FOCS 1980)*:17–27, 1980.

[149] M. Milanič, R. Rizzi and A. I. Tomescu. Set graphs. II. Complexity of set graph recognition and similar problems. Submitted manuscript, 2012.

[150] M. Milanič and A. I. Tomescu. Set graphs. I. Hereditarily finite sets and extensional acyclic orientations. To appear in *Discrete Applied Mathematics.*

[151] M. Milanič and A. I. Tomescu. Set graphs. IV. Further connections to claw-free graphs. In preparation.

[152] M. Molloy and B. Reed. *Graph Colouring and the Probabilistic Method*, 1st edition, Springer, December 2001.

[153] J. Moncel. *Codes Identifiants dans les graphes.* PhD thesis, Université Joseph Fourier Grenoble I, France, 2005.

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

[155] J. Moncel. Monotonicity of the minimum cardinality of an identifying code in the hypercube. *Discrete Applied Mathematics* 154(6):898–899, 2006.

[156] 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.

[157] 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.

[158] T. Müller and J.-S. Sereni. Identifying and locating-dominating codes in (random) geometric networks. *Combinatorics, Probability and Computing* 18(6):925–952, 2009.

[159] 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.

[160] O. M. Omodeo and A. I. Tomescu. Set graphs. III. Provisionnally accepted in *Journal of Automated Reasoning*, 2012.

[161] C. Papadimitriou. *Computational Complexity*, Addison-Wesley, 1994.

[162] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences* 43(3):425–440, 1991.

[163] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *Journal of Computer and System Sciences* 53(2):161–170, 1996.

[164] A. Parreau. *Problèmes d'identification dans les graphes.* PhD thesis, Université de Grenoble, France, 2012.

[165] M. Penrose. *Random Geometric Graphs*, Oxford University Press, 2003.

[166] S. Ranto. *Identifying and locating-dominating codes in binary Hamming spaces*. PhD thesis, Turku Centre for Computer Science, Finland, 2007.

[167] S. Ranto, I. Honkala and T. Laihonen. Two families of optimal identifying codes in binary hamming spaces. *IEEE Transactions on Information Theory* 48(5):1200–1203, 2002.

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

[169] 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.

[170] B. Reed. Paths, stars and the number three. *Combinatroics, Probabilities and Computing* 5(3):277–295, 1996.

[171] D. L. Roberts and F. S. Roberts. Locating sensors in paths and cycles: The case of 2-identifying codes. *European Journal of Combinatorics* 29(1):72–82, 2008.

[172] N. Robertson and P. D. Seymour. Graph Minors. I. Excluding a forest. *Journal of Combinatorial Theory Series B* 35(1):39–61, 1983.

[173] N. Robertson and P. D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory Series B* 92(2):325–357, 2004.

[174] P. Rosendahl. On the identification of vertices using cycles. *The Electronic Journal of Combinatorics* 10:P1, 2003.

[175] E. R. Scheinerman and D. B. West. The interval number of a planar graph: three intervals suffice. *Journal of Combinatorial Theory Series B* 35:224–239, 1983.

[176] S. J. Seo and P. J. Slater. Open neighborhood locating-dominating sets. *The Australasian Journal of Combinatorics* 46:109–120, 2010.

[177] S. J. Seo and P. J. Slater. Open neighborhood locating-domination in trees. *Discrete Applied Mathematics* 159(6):484–489, 2011.

[178] J. Shearer. A note on the independence number of triangle-free graphs. *Discrete Mathematics* 46(1):83–87, 1983.

[179] R. D. Skaggs. *Identifying vertices in graphs and digraphs*. PhD thesis, University of South Africa, South Africa, February 2007, available online at `http://hdl.handle.net/10500/2226`.

[180] P. J. Slater. Leaves of trees. *Congressus Numerantium* 14:549–559, 1975.

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

[182] P. J. Slater. Dominating and reference sets in a graph. *Journal of Mathematical and Physical Sciences* 22(4):445–455, 1988.

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

[184] B. M. Stanton. *On vertex identifying codes for infinite lattices* . PhD thesis, Iowa State University, USA, 2011.

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

[186] 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`.

[187] D. Sumner. Point determination in graphs. *Discrete Mathematics* 5:179–187, 1973.

[188] A. Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory Series B* 81(2):318–338, 2001.

[189] S. Thomassé and A. Yeo. Total domination of graphs and small transversals of hypergraphs. *Combinatorica* 27(4):473–487, 2007.

[190] K. Thulasiraman, M. Xu, Y. Xiao and H. Xiao-Dong. Vertex identifying codes for fault isolation in communication networks. *Proceedings of the International Conference on Discrete Mathematics and Applications, ICDM 2006*, pages 115–124, 2006.

[191] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics* 8(1):85–89, 1984.

[192] L. Trevisan. Inapproximability of combinatorial optimization problems. *Electronic Colloquium on Computational Complexity (ECCC)* TR04-065:2010. Available online at `http://eccc.hpi-web.de/report/2004/065/`.

[193] L. E. Trotter. Line perfect graphs. *Mathematical Programming* 12(1):255–259, 1977.

[194] 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.

[195] P. Valicov. *Problèmes de placement, de coloration et d'identification.* PhD Thesis, Université Bordeaux 1, France, 2012.

[196] K. Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen* 114(1):570–590, 1937.

[197] A. Winter. Another algebraic proof of Bondy's theorem on induced subsets. *Journal of Combinatorial Theory Series A* 89(1):145–147, 2000.

[198] N. Wormald. The asymptotic distribution of short cycles in random regular graphs. *Journal of Combinatorial Theory Series B* 31(2):168–182, 1981.

[199] M. Xua, K. Thulasiraman and X. Hu. Identifying codes of cycles with odd orders. *European Journal of Combinatorics* 29(7):1717–1720, 2008.

[200] M. Yannakakis. Edge-deletion problems. *SIAM Journal of Computing* 10(2):297–309, 1982.

[201] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics* 38:364–372, 1980.

[202] N. E. Young (`http://cstheory.stackexchange.com/users/8237`). Answer to the question "What is the complexity of this covering problem?", `http://cstheory.stackexchange.com/a/11886/1930`.

[203] Z. Zhang, L. Liu and J. Wang. Adjacent strong edge coloring of graphs. *Applied Mathematics Letters* 15(5):623–626, 2002.

Author's publications

[BBFP12] O. Baudon, J. Bensmail, F. Foucaud and M. Pilśniak. On the length of the longest path in partitionable balloons. Submitted manuscript, 2012.

[BFPW12] O. Baudon, F. Foucaud, J. Przybyło and M. Woźniak. On the structure of arbitrarily partitionable graphs with given connectivity. Submitted manuscript, 2012.

[Fo09] F. Foucaud. *Identifying codes in special graph classes.* Master thesis, Université Bordeaux 1, France, June 2009. Available online at `http://www.labri.fr/perso/foucaud/Research/MastersThesis/`.

[FGN+12] F. Foucaud, S. Gravier, R. Naserasr, A. Parreau and P. Valicov. Identifying codes in line graphs. To appear in *Journal of Graph Theory.*

[FGK+11] F. Foucaud, E. Guerrini, M. Kovše, R. Naserasr, A. Parreau and P. Valicov. Extremal graphs for the identifying code problem. *European Journal of Combinatorics* 32(4):628–638, 2011.

[FHL+12] F. Foucaud, I. Honkala, T. Laihonen, A. Parreau and G. Perarnau. Locally identifying colourings of graphs with given maximum degree. *Discrete Mathematics* 312(10):1832–1837, 2012.

[FKKR12] F. Foucaud, R. Klasing, A. Kosowski and A. Raspaud. On the size of identifying codes in triangle-free graphs. *Discrete Applied Mathematics* 160(10-11):1532–1546, 2012.

[FKM+12] F. Foucaud, A. Kosowski, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov. Identifying codes in subclasses of perfect graphs. In preparation.

[FK12] F. Foucaud and M. Kovše. On graph identification problems and the special case of identifying vertices using paths. *Proceedings of the International Workshop on Combinatorial Algorithms, IWOCA 2012.* To appear.

[FLP12] F. Foucaud, T. Laihonen and A. Parreau. An improved lower bound for $(1, \leq 2)$-identifying codes in the king grid. Submitted manuscript, 2012.

[FN12] F. Foucaud and R. Naserasr. On the complexity of signed graph homomorphism problems. In preparation.

[FNP12] F. Foucaud, R. Naserasr and A. Parreau. Characterizing extremal digraphs for identifying codes and extremal cases of Bondy's theorem on induced subsets. To appear in *Graphs and Combinatorics.*

[FP12] F. Foucaud and G. Perarnau. Bounds for identifying codes in terms of degree parameters. *The Electronic Journal of Combinatorics* 19:P32, 2012.

Index of definitions

List of notations

(u, v)	Arc from u to v in a digraph, page 16
$\alpha(G)$	Independence number of graph G, page 20
$\chi(G)$	Chromatic number of graph G, page 21
$\Delta(G)$	Maximum degree of graph G, page 17
$\delta(G)$	Minimum degree of graph G, page 17
$\gamma^{\text{EID}}(G)$	Edge-identifying code number of graph G, page 106
$\gamma(G)$	Domination number of graph G, page 2
$\gamma^{\text{S}}(G)$	Separating code number of graph G, page 2
$\overrightarrow{\gamma}^{\text{ID}}(D)$	Identifying code number of digraph G, page 2
$\gamma^{\text{LD}}(G)$	Location-domination number of graph G, page 21
$\gamma^{\text{ID}}(G)$	Identifying code number of graph G, page 2
$\mathbb{E}(X)$	Expectance of random variable X, page 15
$\mathcal{A} \bowtie K_1$	Set of all graphs of $\mathcal{A}$ with an additional universal vertex, page 57
$\mathcal{A}$	Closure of the set of all graphs A_k with respect to $\bowtie$, page 57
$\mathcal{B}(\mathcal{I}, \mathcal{A}))$	Bipartite incidence graph of set system $(\mathcal{I}, \mathcal{A})$, page 16
$\mathcal{H}_d$	Hypercube of dimension d, page 23
$\mathcal{L}(G)$	Line graph of graph G, page 25
$\mathcal{LOG}(\mathcal{A}, \mathcal{L})$	Bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \mathcal{L})$, page 120
$\mathcal{LOG}^*(\mathcal{A}, \mathcal{L})$	Non-singleton bipartite logarithmic identification of $\mathcal{A}$ over $(\mathcal{A}, \mathcal{L})$, page 120
$\Omega(g(x))$	Big-omega asymptotic notation for $g(x)$, page 15
$\omega(g(x))$	Little-omega asymptotic notation for $g(x)$, page 15
$\omega(G)$	Clique number of graph G, page 20
Ω_n	Big-omega asymptotic notation with respect to variable n, page 15
ω_n	Little-omega asymptotic notation with respect to variable n, page 15
$\overline{d}(G)$	Average degree of graph G, page 17
$\overline{G}$	Complement of graph G, page 18
$\overrightarrow{uv}$	Arc from u to v in a digraph, page 16
$\overrightarrow{\gamma}^{\text{S}}(G)$	Separating code number of digraph D, page 2
$\tau(G)$	Vertex cover number of graph G, page 21
$\Theta(g(x))$	Theta asymptotic notation for $g(x)$, page 15
Θ_n	Theta asymptotic notation with respect to variable n, page 15
$\{u, v\}$	Edge between vertices u, v in an undirected graph, page 16
$A(D)$	Arc set of digraph D, page 16
$A \ominus B$	Symmetric difference between two sets A and B, page 15
A_∞	Basic infinite graph with all its vertices as only identifing code, page 59
A_k	Basic graph on $2k$ vertices with identifying code number $2k-1$, page 56
C_n	Cycle graph on n vertices, page 18
$d(u, v)$	Distance between vertices u and v, page 17
$d^+(v)$	Out-degree of vertex v, page 17
$d^-(v)$	In-degree of vertex v, page 17
$deg(v)$	degree of vertex v, page 17
$E(G)$	Edge set of graph G, page 16
$g(G)$	Girth of graph G, page 18
$G - X$	Graph G with vertices of X removed, page 19
$G - x$	Graph G with vertex x removed, page 19
$G[X]$	Subgraph of G induced by vertex or edge set X, page 19
$G \cong H$	Graphs G and H are isomorphic, page 17

G^r	r^{th} power of graph G, page 18
$G_1 \bowtie G_2$	Complete join of graphs G_1 and G_2, page 19
$G_1 \oplus G_2$	Disjoint union of graphs G_1 and G_2, page 18
K_n	Complete graph on n vertices, page 20
K_n^-	K_n minus one edge, page 20
$K_{n,m}$	Complete bipartite graph with parts of sizes n and m, page 22
$N(v)$	Open neighbourhood of vertex v, page 17
$N(X)$	Union of open neighbourhoods of vertices of X, page 17
$N[v]$	Closed neighbourhood of vertex v, page 17
$N[X]$	Union of closed neighbourhoods of vertices of X, page 17
$N^+(v)$	Out-neighbourhood of vertex v, page 17
$N^+[v]$	Closed out-neighbourhood of vertex v, page 17
$N^-(v)$	In-neighbourhood of vertex v, page 17
$N^-[v]$	Closed in-neighbourhood of vertex v, page 17
$N_k[v]$	Distance-k-closed neighbourhood of vertex v, page 17
$NF(G)$	Proportion of non-forced vertices in graph G, page 71
$o(g(x))$	Little-o asymptotic notation for $g(x)$, page 15
O_n	Big-o asymptotic notation with respect to variable n, page 15
o_n	Little-o asymptotic notation with respect to variable n, page 15
P_n	Path graph on n vertices, page 18
$Pr(A)$	Probability of event A, page 15
$u \not\sim v$	Non-adjacency between vertices u and v, page 16
$u \sim v$	Adjacency between vertices u and v, page 16
uv	Edge between vertices u, v in an undirected graph, page 16
$V(G)$	Vertex set of (di)graph G, page 15
$x \overrightarrow{\triangleleft}(D)$	Digraph D with an extra universal source vertex x, page 49