# Identification, location-domination and metric dimension on interval and permutation graphs. II. Algorithms and complexity

**Florent Foucaud · George B. Mertzios · Reza Naserasr · Aline Parreau · Petru Valicov**

**Abstract** We consider the problems of finding optimal identifying codes, (open) locating-dominating sets and resolving sets (denoted IDENTIFYING CODE, (OPEN) LOCATING-DOMINATING SET and METRIC DIMENSION) of an interval or a permutation graph. In these problems, one asks to distinguish all vertices of a graph by a subset of the vertices, using either the neighbourhood within the solution set or the distances to the solution vertices. Using a general reduction for this class of problems, we prove that the decision problems associated to these four notions are NP-complete, even for interval graphs of diameter 2 and permutation graphs of diameter 2. While IDENTIFYING CODE and (OPEN) LOCATING-DOMINATING SET are trivially fixed-parameter-tractable when parameterized by solution size, it is known that in the same setting METRIC DIMENSION is $W[2]$-hard. We show that for interval graphs, this parameterization of METRIC DIMENSION is fixed-parameter-tractable.

**Keywords** Metric dimension · Resolving set · Identifying code · Locating-dominating set · Interval graph · Permutation graph · Complexity

## 1 Introduction

Combinatorial identification problems have been widely studied in various contexts. The common characteristic of these problems is that we are given a combinatorial structure (graph or hypergraph), and we wish to distinguish (i.e. uniquely identify) its vertices by the means of a small set of selected elements. In this paper, we study several such related identification problems where the instances are graphs. In the problem *metric dimension*, we wish to select a set $S$ of vertices of a graph $G$ such that every vertex of $G$ is uniquely identified by its distances to the vertices of $S$. The notions of *identifying codes* and *(open)*

F. Foucaud
LIMOS - CNRS UMR 6158 Université Blaise Pascal, Clermont-Ferrand, France
E-mail: florent.foucaud@gmail.com

G. B. Mertzios
School of Engineering and Computing Sciences, Durham University, United Kingdom
Partially supported by the EPSRC Grant EP/K022660/1
E-mail: george.mertzios@durham.ac.uk

R. Naserasr
CNRS, IRIF UMR 8243, Université Paris Diderot - Paris 7, France
E-mail: reza@irif.fr

A. Parreau
Université de Lyon,CNRS
Université Lyon 1, LIRIS, UMR 5205, F-69622, France
E-mail: aline.parreau@univ-lyon1.fr

P. Valicov
LIF - CNRS UMR 7279, Université d'Aix-Marseille, France
E-mail: petru.valicov@lif.univ-mrs.fr

*locating-dominating sets* are similar. Roughly speaking, instead of the distances to $S$, we ask for the vertices to be distinguished by their neighbourhood within $S$. These problems have been widely studied since their introduction in the 1970s and 1980s. They have been applied in various areas such as network verification [4,5], fault-detection in networks [41,56], graph isomorphism testing [3] or the logical definability of graphs [43]. We note that the similar problem of finding a *test cover* of a hypergraph (where hyperedges distinguish the vertices) has been studied under several names by various authors, see e.g. [9, 10,15,30,46,49].

**Important concepts and definitions.** All considered graphs are finite and simple. We will denote by $N[v]$, the *closed neighbourhood* of vertex $v$, and by $N(v)$ its *open neighbourhood*, i.e. $N[v] \setminus \{v\}$. A vertex is *universal* if it is adjacent to all the vertices of the graph. A set $S$ of vertices of $G$ is a *dominating set* if for every vertex $v$, there is a vertex $x$ in $S \cap N[v]$. It is a *total dominating set* if instead, $x \in S \cap N(v)$. In the context of (total) dominating sets we say that a vertex $x$ *(totally) separates* two distinct vertices $u, v$ if it (totally) dominates exactly one of them. Set $S$ (totally) separates the vertices of a set $X$ if every pair in $X$ has a vertex in $S$ (totally) separating it. We have the three key definitions, that merge the concepts of (total) domination and (total) separation:

**Definition 1 (Slater [52,53], Babai [3])** *A set $S$ of vertices of a graph $G$ is a* locating-dominating set *if it is a dominating set and it separates the vertices of $V(G) \setminus S$.*

The smallest size of a locating-dominating set of $G$ is the *location-domination number* of $G$, denoted $\gamma^{\mathrm{LD}}(G)$. This concept has also been used under the name *distinguishing set* in [3] and *sieve* in [43].

**Definition 2 (Karpovsky, Chakrabarty and Levitin [41])** *A set $S$ of vertices of a graph $G$ is an* identifying code *if it is a dominating set and it separates all vertices of $V(G)$.*

The smallest size of an identifying code of $G$ is the *identifying code number* of $G$, denoted $\gamma^{\mathrm{ID}}(G)$.

**Definition 3 (Seo and Slater [50])** *A set $S$ of vertices of a graph $G$ is an* open locating-dominating set *if it is a total dominating set and it totally separates all vertices of $V(G)$.*

The smallest size of an open locating-dominating set of $G$ is the *open location-domination number* of $G$, denoted $\gamma^{\mathrm{OLD}}(G)$. This concept has also been called *identifying open code* in [38].

Another kind of separation based on distances is used in the following concept:

**Definition 4 (Harary and Melter [34], Slater [51])** *A set $R$ of vertices of a graph $G$ is a* resolving set *if for each pair $u, v$ of distinct vertices, there is a vertex $x$ of $R$ with $d(x, u) \neq d(x, v)$.*[1]

The smallest size of a resolving set of $G$ is the *metric dimension* of $G$, denoted $dim(G)$.

It is easy to check that the inequalities $dim(G) \leq \gamma^{\mathrm{LD}}(G) \leq \gamma^{\mathrm{ID}}(G)$ and $\gamma^{\mathrm{LD}}(G) \leq \gamma^{\mathrm{OLD}}(G)$ hold, indeed every locating-dominating set of $G$ is a resolving set, and every identifying code (or open locating-dominating set) is a locating-dominating set. Moreover it is proved that $\gamma^{\mathrm{ID}}(G) \leq 2\gamma^{\mathrm{LD}}(G)$ [32] (using the same proof idea one would get a similar relation between $\gamma^{\mathrm{LD}}(G)$ and $\gamma^{\mathrm{OLD}}(G)$ and between $\gamma^{\mathrm{ID}}(G)$ and $\gamma^{\mathrm{OLD}}(G)$, perhaps with a different constant). There is no strict relation between $\gamma^{\mathrm{ID}}(G)$ and $\gamma^{\mathrm{OLD}}(G)$.

In a graph $G$ of diameter 2, one can easily see that the concepts of resolving set and locating-dominating set are almost the same, as $\gamma^{\mathrm{LD}}(G) \leq dim(G) + 1$. Indeed, let $S$ be a resolving set of $G$. Then all vertices in $V(G) \setminus S$ have a distinct neighbourhood within $S$. There might be (at most) one vertex that is not dominated by $S$, in which case adding it to $S$ yields a locating-dominating set.

While a resolving set and a locating-dominating set exist in every graph $G$ (for example the whole vertex set), an identifying code may not exist in $G$ if it contains *twins*, that is, two vertices with the same closed neighbourhood. However, if the graph is *twin-free* the set $V(G)$ is an identifying code of $G$. Similarly, a graph admits an open locating-dominating set if and only if it has no *open twins*, i.e. vertices sharing the same open neighbourhood. We say that such a graph is *open twin-free*.

The focus of this paper is the following set of four decision problems:

LOCATING-DOMINATING-SET
*Instance:* A graph $G$, an integer $k$.
*Question:* Is it true that $\gamma^{\mathrm{LD}}(G) \leq k$?

IDENTIFYING CODE
*Instance:* A graph $G$, an integer $k$.
*Question:* Is it true that $\gamma^{\mathrm{ID}}(G) \leq k$?

---

[1] Resolving sets are also known under the name of *locating sets* [51]. Optimal resolving sets have sometimes been called *metric bases* in the literature; to avoid an inflation in the terminology we will only use the term *resolving set*.

OPEN LOCATING-DOMINATING SET
*Instance:* A graph $G$, an integer $k$.
*Question:* Is it true that $\gamma^{\mathrm{OLD}}(G) \leq k$?

METRIC DIMENSION
*Instance:* A graph $G$, an integer $k$.
*Question:* Is it true that $dim(G) \leq k$?

We will study these four concepts and decision problems on graphs belonging to specific subclasses of perfect graphs (i.e. graphs whose induced subgraphs all have equal clique and chromatic numbers). Many standard graph classes are perfect, for example bipartite graphs, split graphs, interval graphs. For precise definitions, we refer to the books of Brandstädt, Le and Spinrad and of Golumbic [13,31]. Some of these classes are classes defined using a geometric intersection model, that is, the vertices are associated to the elements of a set $S$ of (geometric) objects, and two vertices are adjacent if and only if the corresponding objects intersect. The graph defined by the intersection model $S$ is its *intersection graph*. An *interval graph* is the intersection graph of intervals of the real line, and a *unit interval graph* is an interval graph whose intersection model contains only (open) intervals of unit length. Given two parallel lines $B$ and $T$, a *permutation graph* is the intersection graph of segments of the plane which have one endpoint on $B$ and the other endpoint on $T$.

Interval graphs and permutation graphs are classic graph classes that have many applications and are widely studied. They can be recognized efficiently, and many combinatorial problems have simple and efficient algorithms for these classes.

**Previous work.** The complexity of distinguishing problems has been studied by many authors. IDENTIFYING CODE was first proved to be NP-complete by Charon, Hudry, Lobstein and Zémor [18], and LOCATING-DOMINATING-SET, by Colbourn, Slater and Stewart [19]. Regarding their instance restriction to specific graph classes, IDENTIFYING CODE and LOCATING-DOMINATING-SET were shown to be NP-complete for bipartite graphs by Charon, Hudry and Lobstein [14]. This was improved by Müller and Sereni to planar bipartite unit disk graphs [47], by Auger to planar graphs with arbitrarily large girth [2], and by Foucaud to planar bipartite subcubic graphs [26]. Foucaud, Gravier, Naserasr, Parreau and Valicov proved that IDENTIFYING CODE is NP-complete for graphs that are both planar and line graphs of subcubic bipartite graphs [27]. Berger-Wolf, Laifenfeld, Trachtenberg [7] and Suomela [55] independently showed that both IDENTIFYING CODE and LOCATING-DOMINATING-SET are hard to approximate within factor $\alpha$ for any $\alpha = o(\log n)$ (where $n$ denotes the order of the graph), with no restriction on the input graph. This result was recently extended to bipartite graphs, split graphs and co-bipartite graphs by Foucaud [26]. Moreover, Bousquet, Lagoutte, Li, Parreau and Thomassé [12] proved the same non-approximability result for bipartite graphs with no 4-cycles. On the positive side, IDENTIFYING CODE and LOCATING-DOMINATING-SET are constant-factor approximable for bounded degree graphs (showed by Gravier, Klasing and Moncel in [32]), line graphs [26,27], interval graphs [12] and are linear-time solvable for graphs of bounded clique-width (using Courcelle's theorem [20]). Furthermore, Slater [52] and Auger [2] gave explicit linear-time algorithms solving LOCATING-DOMINATING-SET and IDENTIFYING CODE, respectively, in trees.

The complexity of OPEN LOCATING-DOMINATING SET was not studied much; Seo and Slater showed that it is NP-complete [50], and the inapproximability results of Foucaud [26] for bipartite, co-bipartite and split graphs transfer to it.

The problem METRIC DIMENSION is widely studied. It was shown to be NP-complete by Garey and Johnson [30, Problem GT61]. This result has recently been extended to bipartite graphs, co-bipartite graphs, split graphs and line graphs of bipartite graphs by Epstein, Levin and Woeginger [24], to a special subclass of unit disk graphs by Hoffmann and Wanke [39], and to planar graphs by Diaz, Pottonen, Serna and van Leeuwen [21].

Epstein, Levin and Woeginger [24] also gave polynomial-time algorithms for the weighted version of METRIC DIMENSION for paths, cycles, trees, graphs of bounded cyclomatic number, cographs and partial wheels. Diaz, Pottonen, Serna, van Leeuwen [21] gave a polynomial-time algorithm for outerplanar graphs, and Fernau, Heggernes, van't Hof, Meister and Saei gave one for chain graphs [25]. METRIC DIMENSION can most likely not be expressed in MSOL and it is an open problem to determine its complexity for bounded treewidth (even treewidth 2).

METRIC DIMENSION is hard to approximate within any $o(\log n)$ factor for general graphs, as shown by Beerliova, Eberhard, Erlebach, Hall, Hoffmann, Mihalák and Ram [5]. This is even true for subcubic graphs, as shown by Hartung and Nichterlein [36] (a result extended to *bipartite* subcubic graphs in Hartung's thesis [35]).

In light of these results, the complexity of Locating-Dominating-Set, Open Locating-Dominating Set, Identifying Code and Metric Dimension for interval and permutation graphs is a natural open question (as asked by Manuel, Rajan, Rajasingh, Chris Monica M. [45] and by Epstein, Levin, Woeginger [24] for Metric Dimension on interval graphs), since these classes are standard candidates for designing efficient algorithms to solve otherwise hard problems.

Let us say a few words about the parameterized complexity of these problems. A decision problem is said to be fixed-parameter tractable (FPT) with respect to a parameter $k$ of the instance, if it can be solved in time $f(k)n^{O(1)}$ for an instance of size $n$, where $f$ is a computable function (for definitions and concepts in parameterized complexity, we refer to the books [22, 48]). It is known that for the problems Locating-Dominating-Set, Open Locating-Dominating Set and Identifying Code, for a graph of order $n$ and solution size $k$, the bound $n \le 2^k$ holds (see e.g. [41, 53]). Therefore, when parameterized by $k$, these problems are (trivially) FPT: one can first check whether $n \le 2^k$ holds (if not, return "no"), and if yes, use a brute-force algorithm checking all possible subsets of vertices. This is an FPT algorithm. However, Metric Dimension (parameterized by solution size) is W[2]-hard even for bipartite subcubic graphs [35, 36] (and hence unlikely to be FPT). Remarkably, the bound $n \le D^k + k$ holds [16] (where $n$ is the graph's order, $D$ its diameter, and $k$ is the solution size of Metric Dimension), and therefore for graphs of diameter bounded by a function of $k$, the same arguments as the previous ones yield an FPT algorithm. This holds, for example, for the class of (connected) split graphs, which have diameter at most 3. Also, it was recently proved that Metric Dimension is FPT when parameterized by the largest possible number of leaves in a spanning tree of a graph [23]. Besides this, as remarked in [36], no non-trivial FPT algorithm for Metric Dimension was previously known.

Finally, we also mention a companion paper [29], in which we study problems Identifying Code, Locating-Dominating-Set, Open Locating-Dominating Set and Metric Dimension on interval and permutation graphs from a combinatorial point of view, proving several bounds involving the order, the diameter and the solution size of a graph.

**Our results.** We continue the classification of the complexity of problems Identifying Code, Locating-Dominating-Set, Open Locating-Dominating Set and Metric Dimension by giving a unified reduction showing that all four problems are NP-complete even for graphs that are interval graphs and have diameter 2 or permutation graphs of diameter 2. The reductions are presented in Section 2. Then, in Section 3, we use dynamic programming on a path-decomposition to show that Metric Dimension is FPT on interval graphs, when the parameter is the solution size. Up to our knowledge, this is the first non-trivial FPT algorithm for this problem when parameterized by solution size. We then conclude the paper with some remarks in Section 4.

## 2 Hardness results

We will provide a general framework to prove NP-hardness for distinguishing problems in interval graphs and permutation graphs. We just need to assume few generic properties about the problems, and then provide specific gadgets for each problem.

We will reduce our problems from 3-Dimensional Matching which is known to be NP-complete [40].

3-Dimensional Matching
*Instance:* Three disjoint sets $A$, $B$ and $C$ each of size $n$, and a set $\mathcal{T}$ of $m$ triples of $A \times B \times C$.
*Question:* Is there a perfect 3-dimensional matching $\mathcal{M} \subseteq \mathcal{T}$ of the hypergraph $(A, B, C, \mathcal{T})$, i.e. a set of disjoint triples of $\mathcal{T}$ such that each element of $A \cup B \cup C$ belongs to exactly one of the triples?

We give the general framework and the gadgets we will use in Section 2.1, then prove the general reduction using this framework in Section 2.2 and apply it to obtain the NP-hardness for Identifying Code, Locating-Dominating-Set and Open Locating-Dominating Set in Section 2.3. We finally deduce from the results for graphs of diameter 2 the hardness of Metric Dimension in Section 2.4. We give the reduction using interval graphs and prove subsequently that it can be built as a permutation graph too.

2.1 Preliminaries and gadgets

In the three distinguishing problems LOCATING-DOMINATING-SET, IDENTIFYING CODE and OPEN LOCATING-DOMINATING SET, one asks for a set of vertices that dominates all vertices and separates all pairs of vertices (for suitable definitions of domination and separation). Since we give a reduction which applies to all three problems (and others that share certain properties with them), we will generally speak of a *solution* as a vertex set satisfying the two properties.

For two vertices $u, v$ let us denote by $I_{u,v}$ the set $N[u] \setminus N[v]$. In the reduction, we will only make use of the following properties (that are common to LOCATING-DOMINATING-SET, IDENTIFYING CODE and OPEN LOCATING-DOMINATING SET):

**Property 5** *Let $G$ be a graph with a solution $S$ to* LOCATING-DOMINATING-SET, IDENTIFYING CODE *or* OPEN LOCATING-DOMINATING SET.
*(1) For each vertex $v$, any vertex from $N(v)$ dominates $v$;*
*(2) For each vertex $v$, at least one element of $N[v]$ belongs to $S$;*
*(3) For every pair $u, v$ of adjacent vertices, any vertex of $I_{u,v} \cup I_{v,u}$ separates $u, v$;*
*(4) For every pair $u, v$ of adjacent vertices, $S$ contains a vertex of $I_{u,v} \cup I_{v,u} \cup \{u, v\}$.*

The problems IDENTIFYING CODE and OPEN LOCATING-DOMINATING SET clearly satisfy these properties. For LOCATING-DOMINATING-SET, the vertices of a solution set $S$ do not need to be separated from any other vertex. However one can say equivalently that two vertices $u, v$ are separated if either $u$ or $v$ belongs to $S$, or if there is a vertex of $S$ in $I_{u,v} \cup I_{v,u}$. Therefore, LOCATING-DOMINATING-SET also satisfies the above properties.

Before describing the reduction, we define the following *dominating gadget* independently of the considered problem (we describe the specific gadgets for LOCATING-DOMINATING-SET, OPEN LOCATING-DOMINATING SET and IDENTIFYING CODE in Section 2.3). The idea behind this gadget is to ensure that specific vertices are dominated locally — and are therefore separated from the rest of the graph. We will use it extensively in our construction.

**Definition 6 (Dominating gadget)** *A dominating gadget $D$ is an interval graph such that there exists an integer $d \geq 1$ and a subset $S_D$ of $V(D)$ of size $d$ (called* standard solution *for $D$) with the following properties:*

- *$S_D$ is an optimal solution for $D$ with the property that no vertex of $D$ is dominated by all the vertices of $S_D$;[2]*
- *if $D$ is an induced subgraph of an interval graph $G$ such that each interval of $V(G) \setminus V(D)$ either contains all intervals of $V(D)$ or does not intersect any of them, then for any solution $S$ for $G$, $|S \cap V(D)| \geq d$.[3]*

In the following, a dominating gadget will be represented graphically as shown in Figure 1, where $D$ is an induced subgraph of an interval graph $G$. In our constructions, we will build a graph $G$ with many isomorphic copies of $D$ as its induced subgraphs, where $D$ will be a fixed graph. Denote by $S$ an optimal solution for $G$: the size of each local optimal solution $S \cap V(D)$ for $D$ will always be $d$. Moreover, the conditions of the second property in Definition 6 (that each interval of $V(G) \setminus V(D)$ either contains all intervals of $V(D)$ or does not intersect any of them) will always be satisfied.
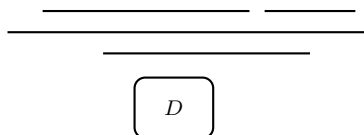


**Fig. 1** Representation of dominating gadget $D$.

---

[2] Note that this implies $d \geq 2$.
[3] By this property, an interval of $V(G) \setminus V(D)$ may only be useful to dominate a vertex in $D$ (but not to separate a pair in $D$). Hence the property holds if any optimal solution for the separation property only, has the same size as an optimal solution for both separation and domination.

**Claim 7** *Let $G$ be an interval graph containing a dominating gadget $D$ as an induced subgraph, such that each interval of $V(G) \setminus V(D)$ either contains all intervals of $V(D)$ or does not intersect any of them. Then, for any optimal solution $S$ of $G$, $|S \cap V(D)| = d$ and we can obtain an optimal solution $S'$ with $|S'| = |S|$ by replacing $S \cap V(D)$ by the standard solution $S_D$.*

*Proof* By the second property of a dominating gadget, we have $|S \cap V(D)| \geq d \geq 1$. Since each interval of $V(G) \setminus V(D)$ either contains all intervals of $V(D)$ or does not intersect any of them, a pair of intervals of $V(G) \setminus V(D)$ either cannot be separated by any interval in $V(D)$, or is separated equally by any interval in $V(D)$. Since $d \geq 1$, there is at least one interval in $S \cap V(D)$ but the structure of $S \cap D$ does not influence the rest of the graph. Hence, $S \cap V(D)$ can be replaced by $S_D$ and we have $|S \cap V(D)| \leq d$ (otherwise the solution with $S_D$ would be better and $S$ would not be optimal). □

**Definition 8 (Choice pair)** *A pair $\{u, v\}$ of intervals is called* choice pair *if $u, v$ both contain the intervals of a common dominating gadget (denoted $D(uv)$), and such that none of $u, v$ contains the other.*

See Figure 2 for an illustration of a choice pair. Intuitively, a choice pair gives us the choice of separating it from the left or from the right: since none of $u, v$ is included in the other, the intervals intersecting $u$ but not $v$ (the set $I_{u,v}$) can only be located at one side of $u$; the same holds for $v$. In our construction, we will make sure that all pairs of intervals will be easily separated using domination gadgets. It will remain to separate the choice pairs.

We have the following claim:

**Claim 9** *Let $S$ be a solution of a graph $G$ and $\{u, v\}$ be a choice pair in $G$. If the solution $S \cap V(D(uv))$ for the dominating gadget $D(uv)$ is the standard solution $S_D$, both vertices $u$ and $v$ are dominated, separated from all vertices in $D(uv)$ and from all vertices not intersecting $D(uv)$.*

*Proof* If $S$ is such a solution, by the definition of a dominating gadget, $|S \cap D(uv)| \geq d \geq 1$. Since all vertices of $D(uv)$ are in the open neighbourhood of $u$ and $v$, by Property 5(1)-(3), $u$ and $v$ are dominated and separated from the vertices not intersecting $D(uv)$. Moreover, both $u, v$ are adjacent to all vertices of $D(uv) \cap S$. By Definition 6, no vertex of $D(uv)$ is dominated by the whole set $S_D$, hence $u, v$ are separated from all vertices in $D(uv)$. □
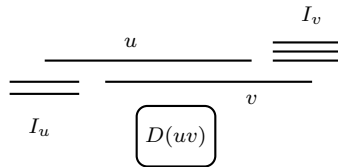


**Fig. 2** Choice pair $u, v$.

We now define the central gadget of the reduction, the *transmitter gadget*. Roughly speaking, it allows to transmit information across an interval graph using the separation property.

**Definition 10 (Transmitter gadget)** *Let $P$ be a set of two or three choice pairs in an interval graph $G$. A* transmitter gadget $Tr(P)$ *is an induced subgraph of $G$ consisting of a path on seven vertices $\{u, uv^1, uv^2, v, vw^1, vw^2, w\}$ and five dominating gadgets $D(u), D(uv), D(v), D(vw), D(w)$ such that the following properties are satisfied:*

- *$u$ and $w$ are the only vertices of $Tr(P)$ that separate the pairs of $P$;*
- *the intervals of the dominating gadget $D(u)$ (resp. $D(v)$, $D(w)$) are included in interval $u$ (resp. $v$, $w$) and no interval of $Tr(P)$ other than $u$ (resp. $v$, $w$) intersects $D(u)$ (resp. $D(v)$, $D(w)$);*
- *pair $\{uv^1, uv^2\}$ is a choice pair and no interval of $V(Tr(P)) \setminus (D(uv^1, uv^2) \cup \{uv^1, uv^2\})$ intersects both intervals of the pair. The same holds for pair $\{vw^1, vw^2\}$.*
- *the choice pairs $\{uv^1, uv^2\}$ and $\{vw^1, vw^2\}$ cannot be separated by intervals of $G$ other than $u$, $v$ and $w$.*
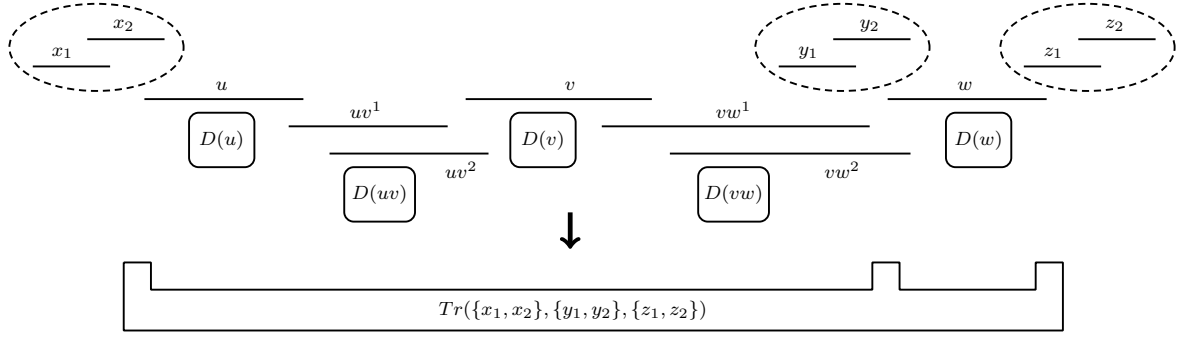
**Fig. 3** Transmitter gadget $Tr(\{x_1, x_2\}, \{y_1, y_2\}, \{z_1, z_2\})$ and its "box" representation.

Figure 3 illustrates a transmitter gadget and shows the succinct graphical representation that we will use. As shown in the figure, we may use a "box" to denote $T_r(P)$. This box does not include the choice pairs of $P$ but indicates where they are situated. Note that the middle pair $\{y_1, y_2\}$ could also be separated (from the left) by $u$ instead of $w$, or it may not exist at all.

The following claim shows how transmitter gadgets will be used in the main reduction.

**Claim 11** *Let $G$ be an interval graph with a transmitter gadget $Tr(P)$ and let $S$ be a solution. We have $|S \cap Tr(P)| \geq 5d + 1$ and if $|S \cap Tr(P)| = 5d + 1$, no pair of $P$ is separated by a vertex in $S \cap Tr(P)$.*

*Moreover, there exist two sets of vertices of $Tr(P)$, $S^-_{Tr(P)}$ and $S^+_{Tr(P)}$ of size $5d + 1$ and $5d + 2$ respectively, such that the following holds.*

- *The set $S^-_{Tr(P)}$ dominates all the vertices of $Tr(P)$ and separates all the pairs of $Tr(P)$ but no pairs in $P$.*
- *The set $S^+_{Tr(P)}$ dominates all the vertices of $Tr(P)$, separates all the pairs of $Tr(P)$ and all the pairs in $P$.*

*Proof* By the definition of the dominating gadget, we must have $|S \cap Tr(P)| \geq 5d$ with $5d$ vertices of $S$ belonging to the dominating gadgets. By Property 5(4) on the choice pair $\{uv^1, uv^2\}$, at least one vertex of $\{u, uv^1, uv^2, v\}$ belongs to $S$ (recall that the intervals not in $Tr(P)$ cannot separate the choice pairs in $Tr(P)$), and similarly, for the choice pair $\{vw^1, vw^2\}$, at least one vertex of $\{v, vw^1, vw^2, w\}$ belongs to $S$. Hence $|S \cap Tr(P)| \geq 5d + 1$ and if $|S \cap Tr(P)| = 5d + 1$, vertex $v$ must be in $S$ and neither $u$ nor $w$ are in $S$. Therefore, no pair of $P$ is separated by a vertex in $S \cap Tr(P)$.

We now prove the second part of the claim. Let $S_{dom}$ be the union of the five standard solutions $S_D$ of the dominating gadgets of $Tr(P)$. Let $S^-_{Tr(P)} = S_{dom} \cup \{v\}$ and $S^+_{Tr(P)} = S_{dom} \cup \{u, w\}$. The set $S_{dom}$ has $5d$ vertices and so $S^-_{Tr(P)}$ and $S^+_{Tr(P)}$ have respectively $5d + 1$ and $5d + 2$ vertices. Each interval of $Tr(P)$ either contains a dominating gadget or is part of a dominating gadget and is therefore dominated by a vertex in $S_{dom}$. Hence, pairs of vertices that are not intersecting the same dominating gadget are clearly separated. By the first property in Definition 6, a vertex adjacent to a whole dominating gadget is separated from all the vertices of the dominating gadget. Similarly, by definition, pairs of vertices inside a dominating gadget are separated by $S_{dom}$. Therefore, the only remaining pairs to consider are the choice pairs. By Property 5(3), they are separated both at the same time either by $v$ or by $\{u, w\}$. Hence the two sets $S^-_{Tr(P)}$ and $S^+_{Tr(P)}$ are both dominating and separating the vertices of $Tr(P)$. Moreover, since $S^+_{Tr(P)}$ contains $u$ and $w$, it also separates the pairs of $P$. □

A transmitter gadget with a solution set of size $5d + 1$ (resp. $5d + 2$ vertices) is said to be *tight* (resp. *non-tight*). We will call the sets $S^-_{Tr(P)}$ and $S^+_{Tr(P)}$ the *tight* and *non-tight standard solutions* of $Tr(P)$.

### 2.2 The main reduction

We are now ready to describe the reduction from 3-DIMENSIONAL MATCHING. Each element $x \in A \cup B \cup C$ is modelled by a choice pair $\{f_x, g_x\}$. Each triple of $\mathcal{T}$ is modelled by a triple gadget defined as follows.

**Definition 12 (Triple gadget)** *Let $T = \{a, b, c\}$ be a triple of $\mathcal{T}$. The triple gadget $G_t(T)$ is an interval graph consisting of four choice pairs $p = \{p_1, p_2\}$, $q = \{q_1, q_2\}$, $r = \{r_1, r_2\}$, $s = \{s_1, s_2\}$ together with their associated dominating gadgets $D(p), D(q), D(r), D(s)$ and five transmitter gadgets $Tr(p, q), Tr(r, s)$, $Tr(s, a), Tr(p, r, b)$ and $Tr(q, r, c)$, where:*

- *$a = \{f_a, g_a\}$, $b = \{f_b, g_b\}$ and $c = \{f_c, g_c\}$;*
- *Except for the choice pairs $p$, $q$, $r$, $s$, $a$, $b$, $c$, for each pair of intervals of $G_t(T)$, its two intervals intersect different subsets of dominating gadgets of $G_t(T)$.*
- *In each transmitter gadget $Tr(P)$ and for each choice pair $\pi \in P$, the intervals of $\pi$ intersect the same intervals except for the vertices $u, v, w$ of $Tr(P)$;*
- *The intervals of $V(G) \setminus V(G_t(T))$ that are intersecting only a part of the gadget intersect accordingly to the transmitter gadget definition and do not separate the choice pairs $p$, $q$, $r$ and $s$.*

Note that there are several ways to obtain a triple gadget that is an interval graph and that satisfies the properties in Definition 12. The one in Figure 4 represents one of these possibilities. We remark that $p$, $q$, $r$ and $s$ in $G_t(\{a, b, c\})$, are all functions of $\{a, b, c\}$ but to simplify the notations we simply write $p$, $q$, $r$ and $s$.
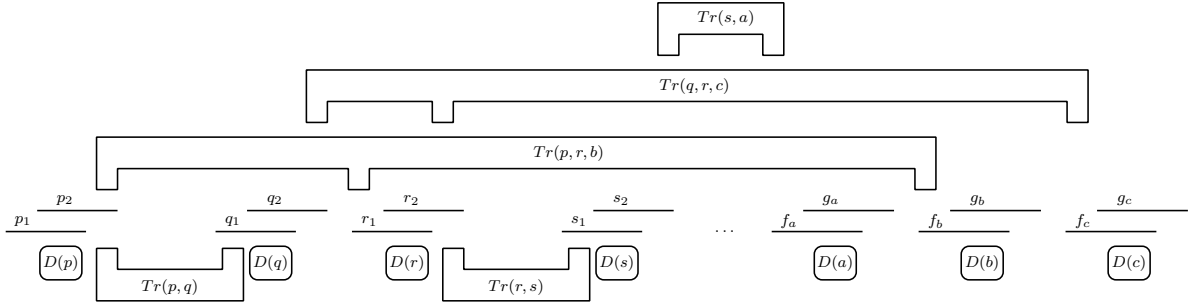


**Fig. 4** Triple gadget $G_t(\{a, b, c\})$ together with choice pairs of elements $a$, $b$ and $c$.

**Claim 13** *Let $G$ be a graph with a triple gadget $G_t(T)$ and $S$ be a solution. We have $|S \cap G_t(T)| \geq 29d + 7$ and if $|S \cap G_t(T)| = 29d + 7$, no choice pair corresponding to $a$, $b$ or $c$ is separated by a vertex in $S \cap G_t(T)$.*

*Moreover, there exist two sets of vertices of $G_t(T)$, $S^-_{G_t(T)}$ and $S^+_{G_t(T)}$ of size $29d + 7$ and $29d + 8$ respectively, such that the following holds.*

- *The set $S^-_{G_t(T)}$ dominates all the vertices of $G_t(T)$ and separates all the pairs of $G_t(T)$ but does not separate any choice pairs corresponding to $\{a, b, c\}$.*
- *The set $S^+_{G_t(T)}$ dominates all the vertices of $G_t(T)$, separates all the pairs of $G_t(T)$ and separates the choice pairs corresponding to $\{a, b, c\}$.*

*Proof* The proof is similar of the proof of Claim 11. Each transmitter gadget must contain at least $5d + 1$ vertices, and each of the four dominating gadgets of the choice pairs $p$, $q$, $r$, $s$ must contain $d$ vertices. Hence there must be already $29d + 5$ vertices of $G_t(T)$ in the solution. Furthermore, to separate the choice pair $s$, $Tr(r, s)$ or $Tr(s, a)$ must be non-tight (since $s$ is not separated by other vertices of the graph). In the same way, to separate the choice pair $p$, $Tr(p, q)$ or $Tr(p, r, b)$ must be non-tight. Then at least two transmitter gadgets are non-tight and we have $|S \cap G_t(T)| \geq 29d + 7$. If $|S \cap G_t(T)| = 29d + 7$, exactly two transmitter gadgets are non-tight and they can only be $Tr(r, s)$ and $Tr(p, q)$ (otherwise some of the choice pairs $p, q, r, s$ would not be separated). Hence the choice pairs corresponding to $\{a, b, c\}$ are not separated by the vertices of $G_t(T) \cap S$.

For the second part of the claim, the set $S^-_{G_t(T)}$ is defined by taking the union of the tight standard solutions of $Tr(s, a), Tr(q, r, c)$ and $Tr(p, r, b)$), the non-tight standard solutions of $Tr(p, q)$ and $Tr(r, s)$ and the standard solutions of the dominating gadgets $D(p), D(q), D(r)$ and $D(s)$. The set $S^+_{G_t(T)}$ is defined by taking the union of the non-tight standard solutions of $Tr(s, a), Tr(q, r, c)$ and $Tr(p, r, b)$, the tight standard solutions of $Tr(p, q)$ and $Tr(r, s)$ and the standard solutions of the dominating gadgets $D(p), D(q), D(r)$ and $D(s)$. By Claim 11, the definition of a dominating gadget and the fact that the only intervals sharing the same sets of dominating gadgets are the choice pairs, all intervals of $G_t(T)$ are

dominated and all the pairs of intervals except the choice pairs are separated by both $S^-_{G_t(T)}$ and $S^+_{G_t(T)}$. The choice pairs $p$, $q$, $r$ and $s$ are separated by the non-tight solutions of the transmitter gadgets. Hence $S^-_{G_t(T)}$ and $S^+_{G_t(T)}$ are dominating and separating all the intervals of $G_t(T)$.

When the solution contains $S^-_{G_t(T)}$, the transmitter gadgets $Tr(s,a)$, $Tr(q,r,c)$ and $Tr(p,r,b)$ are tight. Hence $S^-_{G_t(T)}$ does not separate any choice pairs among $\{a,b,c\}$. On the other hand, since $S^+_{G_t(T)}$ contains the non-tight solution of $Tr(s,a)$, $Tr(q,r,c)$ and $Tr(p,r,b)$, the three choice pairs $\{a,b,c\}$ are separated by $S^+_{G_t(T)}$. □

As before, a triple gadget with $29d+7$ vertices (resp. $29d+8$) is said to be *tight* (resp. *non-tight*). We will call the sets $S^-_{G_t(T)}$ and $S^+_{G_t(T)}$ the *tight* and *non-tight standard solutions* of $G_t(T)$.

Given an instance $(A, B, C, \mathcal{T})$ of 3-Dimensional Matching with $|A| = |B| = |C| = n$ and $|\mathcal{T}| = m$, we construct the interval graph $G = G(A, B, C, \mathcal{T})$ as follows.

- As mentioned previously, to each element $x$ of $A \cup B \cup C$, we assign a distinct choice pair $\{f_x, g_x\}$ in $G$. The intervals of any two distinct choice pairs $\{f_x, g_x\}, \{f_y, g_y\}$ are disjoint and they are all in $\mathbb{R}^+$.
- For each triple $T = \{a, b, c\}$ of $\mathcal{T}$ we first associate an interval $I_T$ in $\mathbb{R}^-$ such that for any two triples $T_1$ and $T_2$, $I_{T_1}$ and $I_{T_2}$ do not intersect. Then inside $I_T$, we build the choice pairs $\{p_1, p_2\}$, $\{q_1, q_2\}$, $\{r_1, r_2\}$, $\{s_1, s_2\}$. Finally, using the choice pairs already associated to elements $a$, $b$ and $c$ we complete this to a triple gadget.
- When placing the remaining intervals of the triple gadgets, we must ensure that triple gadgets do not "interfere": for every dominating gadget $D$, no interval in $V(G) \setminus V(D)$ must have an endpoint inside $D$. Similarly, the choice pairs of every triple gadget or transmitter gadget must only be separated by intervals among $u$, $v$ and $w$ of its corresponding private transmitter gadget.

  For intervals of distinct triple gadgets, this is easily done by our placement of the triple gadgets. To ensure that the intervals of transmitter gadgets of the same triple gadget do not interfere, we proceed as follows. We place the whole gadget $Tr(p, q)$ inside interval $u$ of $Tr(p, r, b)$. Similarly, the whole $Tr(r, s)$ is placed inside interval $w$ of $Tr(q, r, c)$ and the whole $Tr(s, a)$ is placed inside interval $v$ of $Tr(p, r, b)$. One has to be more careful when placing the intervals of $Tr(p, r, b)$ and $Tr(q, r, c)$. In $Tr(p, r, b)$, we must have that interval $u$ separates $p$ from the right of $p$. We also place $u$ so that it separates $r$ from the left of $r$. Intervals $uv^1, uv^2$ both start in $r_1$, so that $u$ also separates $uv^1, uv^2$ and also to ensure that $uv^1, uv^2$ does not separate the choice pair $r$. Intervals $uv^1, uv^2$ continue until after pair $s$. In $Tr(q, r, c)$, we place $u$ so that it separates $q$ from the right, and we place $w$ so that it separates $r$ from the right; intervals $uv^1, uv^2, v$ lie strictly between $q$ and $r$; intervals $vw^1, vw^2$ intersect $r_1, r_2$ but stop before the end of $r_2$ (so that $w$ can separate both pairs $vw^1, vw^2$ and $r$ but without these pairs interfering). It is now easy to place $Tr(s, a)$ between $s$ and $a$. An example is given in Figure 5.
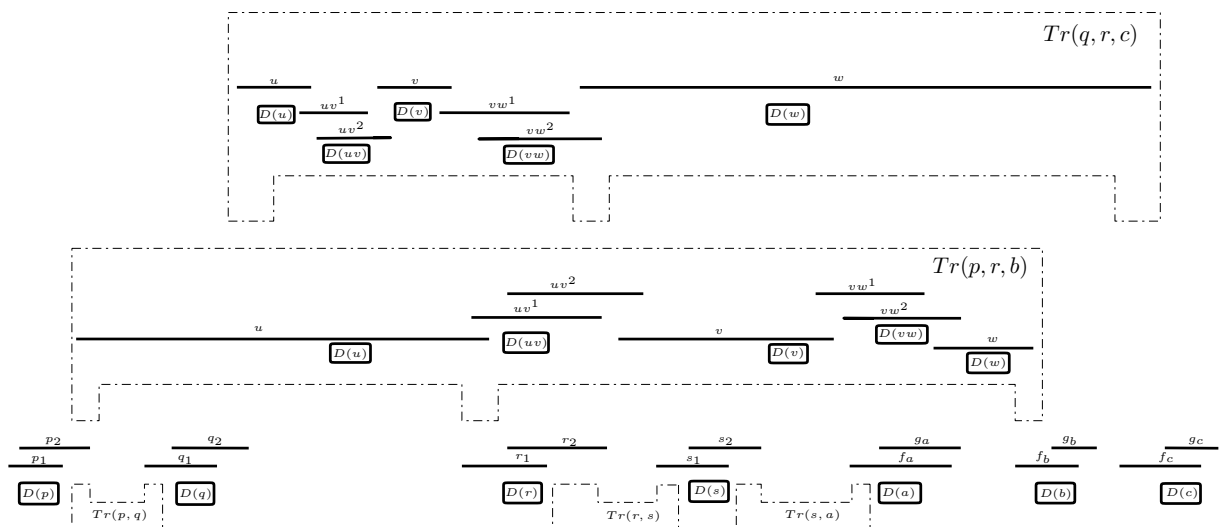


**Fig. 5** The detailed construction of a triple gadget.

The graph $G(A, B, C, \mathcal{T})$ has $(29v_D + 43)m + 3(v_D + 2)n$ vertices (where $v_D$ is the order of a dominating gadget) and the interval representation described by our procedure can be obtained in polynomial time. We are now ready to state the main result of this section.

**Theorem 14** $(A, B, C, \mathcal{T})$ *has a perfect 3-dimensional matching if and only if $G = G(A, B, C, \mathcal{T})$ has a solution with $(29d + 7)m + (3d + 1)n$ vertices.*

*Proof* Let $\mathcal{M}$ be a perfect 3-dimensional matching of $(A, B, C, \mathcal{T})$. Let $S^+$ (resp. $S^-$) be the union of all the non-tight (resp. tight) standard solutions $S^+_{G_t(T)}$ for $T \in \mathcal{M}$ (resp. $S^-_{G_t(T)}$ for $T \notin \mathcal{M}$). Let $S_d$ be the union of all the standard solutions of the dominating gadgets corresponding to the choice pairs of the elements.

Then $S = S^+ \cup S^- \cup S_d$ is a solution of size $(29d + 7)m + (3d + 1)n$. Indeed, by the definition of the dominating gadgets, all the intervals inside a dominating gadget are dominated and separated from all the other intervals. All the other intervals intersect at least one dominating gadget and thus are dominated. Furthermore, two intervals that are not a choice pair do not intersect the same set of dominating gadgets and thus are separated by one of the dominating gadgets. Finally, the choice pairs inside a triple gadget are separated by Claim 13 and the choice pairs corresponding to the elements of $A \cup B \cup C$ are separated by the non-tight standard solutions of the triple gadgets corresponding to the perfect matching.

Now, let $S$ be a solution of size $(29d + 7)m + (3d + 1)n$. We may assume that the solution is standard on all triple gadgets and on the dominating gadgets. Let $n_2$ be the number of non-tight triple gadgets in the solution $S$. By Claim 13, there must by at least $(29d + 7)m + n_2$ vertices of $S$ inside the $m$ triple gadgets and $3dn$ vertices of $S$ for the dominating gadgets of the $3n$ elements of $A \cup B \cup C$. Hence $(29d + 7)m + n_2 + 3dn \leq (29d + 7)m + (3d + 1)n$ and we have $n_2 \leq n$. Each non-tight triple gadget can separate three choice pairs corresponding to the elements of $A \cup B \cup C$. Hence, if $n_2 < n$, it means that at least $3(n - n_2)$ choice pairs corresponding to elements are not separated by a triple gadget. By the separation property, the only way to separate a choice pair $\{f_x, g_x\}$ without using a non-tight triple gadget is to have $f_x$ or $g_x$ in the solution. Hence we need $3(n - n_2)$ vertices to separate these $3(n - n_2)$ choice pairs, and these vertices are not in the triple gadgets nor in the dominating gadgets. Hence the solution has size at least $(29d + 7)m + n_2 + 3dn + 3(n - n_2) > (29d + 7)m + (3d + 1)n$, leading to a contradiction.

Therefore, $n_2 = n$ and there are exactly $n$ non-tight triple gadgets. Each of them separates three choice element pairs and since there are $3n$ elements, the non-tight triple gadgets separate distinct choice pairs. Hence, the set of triples $\mathcal{M}$ corresponding to the non-tight triple gadgets is a perfect 3-dimensional matching of $(A, B, C, \mathcal{T})$. □

**Corollary 15** *Any graph distinguishing problem $P$ based on domination and separation satisfying Property 5 and admitting a dominating gadget that is an interval graph, is NP-complete even for the class of interval graphs.*

A similar hardness result can be derived for the class of permutation graphs as follows.

**Corollary 16** *Any graph distinguishing problem $P$ based on domination and separation satisfying Property 5 and admitting a dominating gadget that is a permutation graph, is NP-complete even for the class of permutation graphs.*

*Proof* We can use the same reduction as the one that yields Theorem 14. We represent a permutation graph using its intersection model of segments as defined in the introduction. A dominating gadget will be represented as in Figure 6. The transmitter gadget of Definition 10 is also a permutation graph, see Figure 7 for an illustration. Using these gadgets, we can build a triple gadget that satisfies Definition 12 and is a permutation graph. A simplified permutation diagram (without dominating gadgets) of such a triple gadget is given in Figure 8. Now, similarly as for interval graphs, given an instance $(A, B, C, \mathcal{T})$ of 3-Dimensional Matching, one can define a graph $G = G(A, B, C, \mathcal{T})$ that is a permutation graph and for which $(A, B, C, \mathcal{T})$ has a perfect 3-dimensional matching if and only if $G = G(A, B, C, \mathcal{T})$ has a solution with $(29d + 7)m + (3d + 1)n$ vertices. The proof is the same as the one in Theorem 14.
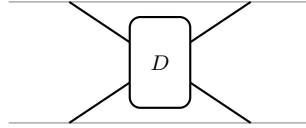
**Fig. 6** Representation of a dominating gadget as a permutation diagram.
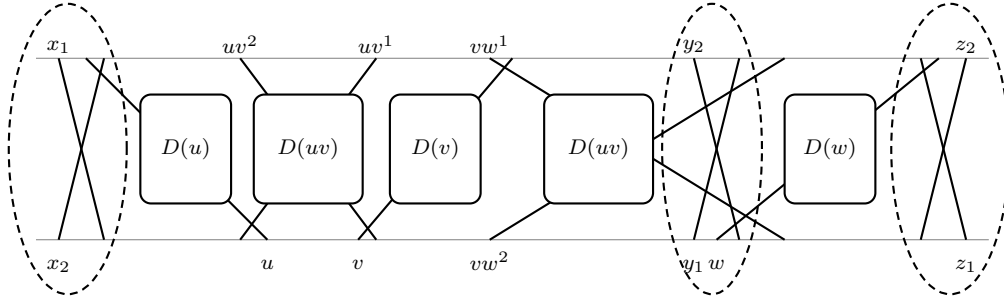


**Fig. 7** Representation of a the transmitter gadget as a permutation diagram.
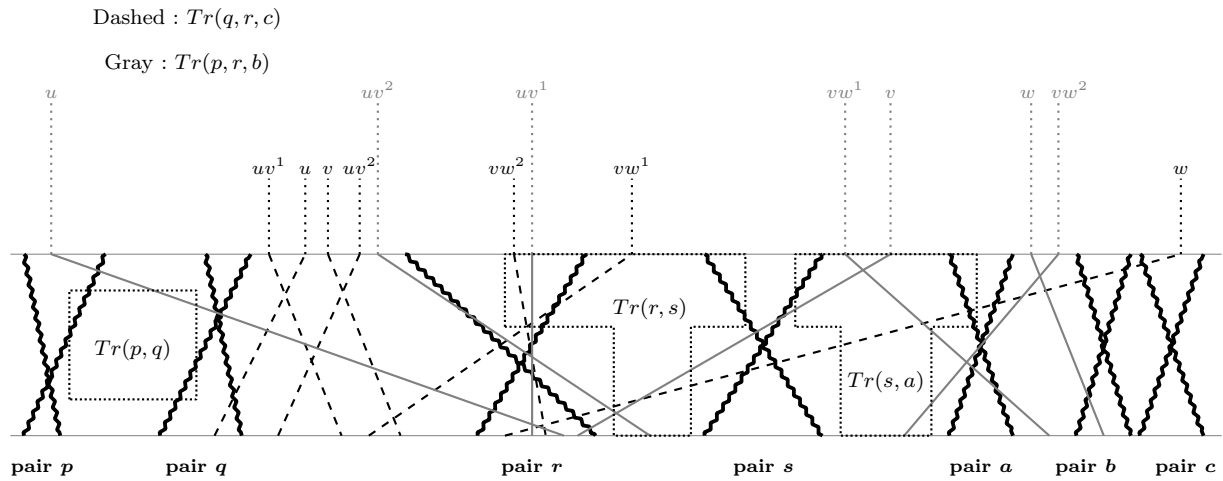


**Fig. 8** Representation of a the triple gadget as a permutation diagram.

2.3 Applications to the specific problems

We now apply Theorem 14 to LOCATING-DOMINATING-SET, IDENTIFYING CODE and OPEN LOCATING-DOMINATING SET by providing corresponding dominating gadgets.

**Corollary 17** LOCATING-DOMINATING-SET*,* IDENTIFYING CODE *and* OPEN LOCATING-DOMINATING SET *are NP-complete for interval graphs and permutation graphs.*

*Proof* We prove that the path graphs $P_4$, $P_5$ and $P_6$ are dominating gadgets for LOCATING-DOMINATING-SET, IDENTIFYING CODE and OPEN LOCATING-DOMINATING SET, respectively. These graphs are clearly interval and permutation graphs at the same time. To comply with Definition 6, we must prove that a dominating gadget $D$ (i) has an optimal solution $S_D$ of size $d$ such that no vertex of $D$ is dominated by all the vertices of $S_D$, and (ii) if $D$ is an induced subgraph of an interval graph $G$ such that each interval of $V(G) \setminus V(D)$ either contains all intervals of $V(D)$ or does not intersect any of them, then for any solution $S$ for $G$, $|S \cap V(D)| \geq d$.

– LOCATING-DOMINATING-SET. Let $V(P_4) = \{x_1, \ldots, x_4\}$ and $d = 2$. The set $S_D = \{x_1, x_4\}$ satisfies (i). For (ii), assume that $S$ is a locating-dominating set of a graph $G$ containing a copy $P$ of $P_4$ satisfying the conditions. If $S \cap P = \emptyset$ or $S \cap P = \{x_1\}$, then $x_3$ and $x_4$ are not separated. If $S \cap P = \{x_2\}$, then $x_1$ and $x_3$ are not separated. Hence, by symmetry, there at least two vertices of $P$ in $S$, and (ii) is satisfied.

- IDENTIFYING CODE. Let $V(P_5) = \{x_1, \ldots, x_5\}$ and $d = 3$. The set $S_D = \{x_1, x_3, x_5\}$ satisfies (i). For (ii), assume that $S$ is an identifying code of a graph $G$ containing a copy $P$ of $P_5$ satisfying the conditions. To separate the pair $\{x_1, x_2\}$, we must have $x_3 \in S$, since the other vertices cannot separate any pair inside $P$. To separate the pair $\{x_2, x_3\}$, we must have $\{x_1, x_4\} \cap S \neq \emptyset$ and to separate the pair $\{x_3, x_4\}$, we must have $\{x_2, x_5\} \cap S \neq \emptyset$. Hence there at least three vertices of $P$ in $S$, and (ii) is satisfied.
- OPEN LOCATING-DOMINATING SET. Let $V(P_6) = \{x_1, \ldots, x_6\}$ and $d = 4$. The set $S_D = \{x_1, x_3, x_4, x_6\}$ satisfies (i). For (ii), assume that $S$ is an open locating-dominating set of a graph $G$ containing a copy $P$ of $P_6$ satisfying the conditions. To separate the pair $\{x_1, x_3\}$, we must have $x_4 \in S$. Symmetrically, $x_3 \in S$. To separate the pair $\{x_2, x_4\}$, we might have $\{x_1, x_5\} \cap S \neq \emptyset$ and symmetrically, $\{x_2, x_6\} \cap S \neq \emptyset$. Hence there at least four vertices of $P$ in $S$, and (ii) is satisfied.

$\square$

### 2.4 Reductions for diameter 2 and consequence for METRIC DIMENSION

We now describe self-reductions for IDENTIFYING CODE, LOCATING-DOMINATING-SET and OPEN LOCATING-DOMINATING SET for graphs with a universal vertex (hence, graphs of diameter 2). We also give a similar reduction from LOCATING-DOMINATING-SET to METRIC DIMENSION.

Let $G$ be a graph. We define $f_1(G)$ to be the graph obtained from $G$ by adding a universal vertex $u$ and then, a neighbour $v$ of $u$ of degree 1. Similarly, $f_2(G)$ is the graph obtained from $f_1(G)$ by adding a twin $w$ of $v$. See Figures 9(a) and 9(b) for an illustration.



(a) Transformation $f_1(G)$        (b) Transformation $f_2(G)$        (c) Transformation $f_3(G)$
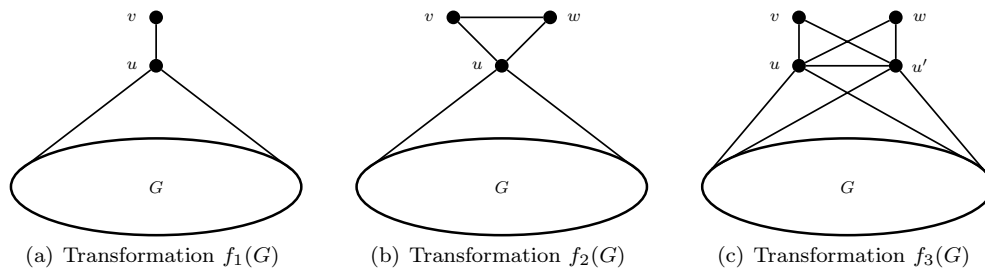
**Fig. 9** Three reductions for diameter 2.

**Lemma 18** *For any graph $G$, we have $\gamma^{LD}(f_1(G)) = \gamma^{LD}(G) + 1$. If $G$ is twin-free, $\gamma^{ID}(f_1(G)) = \gamma^{ID}(G) + 1$. If $G$ is open twin-free, $\gamma^{OLD}(f_2(G)) = \gamma^{OLD}(G) + 2$.*

*Proof* Let $S$ be an identifying code of $G$. Then $S \cup \{v\}$ is also an identifying code of $f_1(G)$: all vertices within $V(G)$ are distinguished by $S$ as they were in $G$; vertex $v$ is dominated only by itself; vertex $u$ is the only vertex dominated by the whole set $S \cup \{v\}$. The same argument works for a locating-dominating set. Hence, $\gamma^{LD}(f_1(G)) \leq \gamma^{LD}(G) + 1$ and $\gamma^{ID}(f_1(G)) \leq \gamma^{ID}(G) + 1$. If $S$ is an open locating-dominating set of $G$, then similarly, $S \cup \{v, w\}$ is one of $f_2(G)$, hence $\gamma^{OLD}(f_2(G)) \leq \gamma^{OLD}(G) + 2$.

It remains to prove the converse. Let $S_1$ be an identifying code (or locating-dominating set) of $f_1(G)$. Observe that $|S_1 \cap \{u, v\}| \geq 1$ since $v$ must be dominated. Hence if $S_1 \setminus \{u, v\}$ is an identifying code (or locating-dominating set) of $G$, we are done. Let us assume the contrary. Then, necessarily $u \in S_1$ since $v$ does not dominate any vertex of $V(G)$. But $u$ is a universal vertex, hence $u$ does not separate any pair of vertices of $V(G)$. Therefore, $S_1 \setminus \{u\}$ separates all pairs, but does not dominate some vertex $x \in V(G)$: we have $N[x] \cap S_1 = \{u\}$. Note that $x$ is the only such vertex of $G$. This implies that $v \in S_1$ (otherwise $x$ and $v$ are not separated by $S_1$). But then $(S_1 \setminus \{u, v\}) \cup \{x\}$ is an identifying code (or locating-dominating set) of $G$ of size $|S_1| - 1$. This completes the proof.

A similar proof works for open location-domination: if $S_2$ is an open locating-dominating set of $f_2(G)$, then $|S_2 \cap \{u, v, w\}| \geq 2$ since $v, w$ must be separated and totally dominated. Similarly, if $S_2 \setminus \{u, v, w\}$ is an open locating-dominating set of $G$, we are done. Otherwise, again $u$ must belong to $S_2$, and is needed only for domination. But then if there is a vertex among $v, w$ that is not in $S_2$, the other one would not be

separated from the vertex $x$ only dominated by $u$. But then $S_2 \setminus \{u, v, w\} \cup \{y\}$, for any vertex $y \in N(x)$, is an open locating-dominating set of size $|S_2| - 2$ and we are done.                    $\square$

Lemma 18 directly implies the following theorem:

**Theorem 19** *Let $\mathcal{C}$ be a class of graphs that is closed under the graph transformation $f_1$ ($f_2$, respectively). If* IDENTIFYING CODE *or* LOCATING-DOMINATING-SET *(*OPEN LOCATING-DOMINATING SET, *respectively) is NP-complete for graphs in $\mathcal{C}$, then it is also NP-complete for graphs in $\mathcal{C}$ that have diameter 2.*

Theorem 19 can be applied to the classes of split graphs (for $f_1$), interval graphs and permutation graphs (for both $f_1$ and $f_2$). By the results about split graphs from [26] and about interval graphs and permutation graphs of Corollary 17, we have:

**Corollary 20** IDENTIFYING CODE *and* LOCATING-DOMINATING-SET *are NP-complete for split graphs of diameter 2.* IDENTIFYING CODE, LOCATING-DOMINATING-SET *and* OPEN LOCATING-DOMINATING SET *are NP-complete for interval graphs of diameter 2 and for permutation graphs of diameter 2.*

We now a give a similar reduction from LOCATING-DOMINATING-SET to METRIC DIMENSION. Given a graph $G$, let $f_3(G)$ be the graph obtained from $G$ by adding two adjacent universal vertices $u, u'$ and then, two non-adjacent vertices $v$ and $w$ that are only adjacent to $u$ and $u'$ (see Figure 9(c) for an illustration).

**Lemma 21** *For any graph $G$, $dim(f_3(G)) = \gamma^{LD}(G) + 2$.*

*Proof* Let $S$ be a locating-dominating set of $G$. We claim that $S_3 = S \cup \{u, v\}$ is a resolving set of $f_3(G)$. Every vertex of $S_3$ is clearly distinguished. Every original vertex of $G$ is determined by a distinct set of vertices of $S$ that are at distance 1 of it. Vertex $u'$ is the only vertex to be at distance 1 of each vertex in $S_3$. Finally, vertex $w$ is the only vertex to be at distance 1 of $u$ and at distance 2 from all other vertices of $S_3$.

For the other direction, assume $B$ is a resolving set of $f_3(G)$. Then necessarily one of $u, u'$ (say $u$) belongs to $B$; similarly, one of $v, w$ (say $v$) belongs to $B$. Hence, if the restriction $B_G = B \cap V(G)$ is a locating-dominating set of $G$, we are done. Otherwise, since no vertex among $u, u', v, w$ may distinguish any pair of $G$ and since vertices of $G$ are at distance at most 2 in $f_3(G)$, all the sets $N[x] \cap B$ are distinct for $x \in V(G) \setminus B_G$. But $B_G$ is not a locating-dominating set, so there is a (unique) $x$ vertex of $G$ that is not dominated by $B_G$ in $G$. If $|B \cap \{u, u', v, w\}| \geq 3$, $B_G \cup \{x\}$ is a locating-dominating set of size at most $|B| - 2$ and we are done. Otherwise, note that in $f_3(G)$, $x$ is at distance 1 from $u$ and at distance 2 from all other vertices of $B$. But this is also the case for $w$, which is not separated from $x$ by $B$, which is a contradiction.                    $\square$

We obtain the following results:

**Theorem 22** *Let $\mathcal{C}$ be a class of graphs that is closed under the graph transformation $f_3$. If* LOCATING-DOMINATING-SET *is NP-complete for graphs in $\mathcal{C}$, then* METRIC DIMENSION *is also NP-complete for graphs in $\mathcal{C}$ that have diameter 2.*

Again, using the results about split graphs from [26] and about interval graphs and permutation graphs of Corollary 17, we have:

**Corollary 23** METRIC DIMENSION *is NP-complete for split graphs of diameter 2, for interval graphs of diameter 2 and for permutation graphs of diameter 2.*

## 3 METRIC DIMENSION parameterized by solution size is FPT on interval graphs

The purpose of this section is to prove that METRIC DIMENSION (parameterized by solution size) is FPT on interval graphs. We begin with preliminary results, before describing our algorithm and proving its correctness. The algorithm is based on dynamic programming over a path-decomposition.

3.1 Preliminaries

We start by stating a few properties and lemmas that are necessary for our algorithm.

*3.1.1 Interval graphs*

Given an interval graph $G$, we can assume that in its interval model, all endpoints are distinct, and that the intervals are closed intervals. Given an interval $I$, we will denote by $\ell(I)$ and by $r(I)$ its left and right endpoints, respectively. We define two natural total orderings of $V(G)$ based on this model: $x <_L y$ if and only if the left endpoint of $x$ is smaller then the left endpoint of $y$, and $x <_R y$ if and only if the right endpoint of $x$ is smaller than the right endpoint of $y$.

Given a graph $G$, its *distance-power* $G^d$ is the graph obtained from $G$ by adding an edge between each pair of vertices at distance at most $d$ in $G$. We will use the following result.

**Theorem 24 ([1])** *Let $G$ be an interval graph with an interval model inducing orders $<_L$ and $<_R$, and let $d \geq 2$ be an integer. Then the power graph $G^d$ is an interval graph with an interval model inducing the same orders $<_L$ and $<_R$ as $G$ (that can be computed in linear time).*

*3.1.2 Tree-decompositions*

**Definition 25** *A* tree-decomposition *of a graph $G$ is a pair $(\mathscr{T}, \mathcal{X})$, where $\mathscr{T}$ is a tree and $\mathcal{X} := \{X_t : t \in V(\mathscr{T})\}$ is a collection of subsets of $V(G)$ (called* bags*), such that they satisfy the following conditions:*
*(i) $\bigcup_{t \in V(\mathscr{T})} X_t = V(G)$;*
*(ii) for every edge $uv \in E(G)$, there is a bag of $\mathcal{X}$ that contains both $u$ and $v$;*
*(iii) for every vertex $v \in V(G)$, the set of bags containing $v$ induces a subtree of $\mathscr{T}$.*

Given a tree-decomposition of $(\mathscr{T}, \mathcal{X})$, the maximum size of a bag $X_t$ over all tree nodes $t$ of $\mathscr{T}$ minus one is called the *width* of $(\mathscr{T}, \mathcal{X})$. The minimum width of a tree-decomposition of $G$ is the *treewidth* of $G$. The notion of tree-decomposition has been used extensively in algorithm design, especially via dynamic programming over the tree-decomposition.

We consider a *rooted* tree-decomposition by fixing a root of $\mathscr{T}$ and orienting the tree edges from the root toward the leaves. A rooted tree-decomposition is *nice* (see Kloks [44]) if each node $t$ of $\mathscr{T}$ has at most two children and falls into one of the four types:
(i) *Join* node: $t$ has exactly two children $t_1$ and $t_2$, and $X_t = X_{t_1} = X_{t_2}$.
(ii) *Introduce* node: $t$ has a unique child $t'$, and $X_t = X_{t'} \cup \{v\}$.
(iii) *Forget* node: $t$ has a unique child $t'$, and $X_t = X_{t'} \setminus \{v\}$.
(iv) *Leaf* node: $t$ is a leaf node in $\mathscr{T}$.

Given a tree-decomposition, a nice tree-decomposition of the same width always exists and can be computed in linear time [44].

If $G$ is an interval graph, we can construct a tree-decomposition of $G$ (in fact, a path-decomposition) with special properties.

**Proposition 26** *Let $G$ be an interval graph with clique number $\omega$ and an interval model inducing orders $<_L$ and $<_R$. Then, $G$ has a nice tree-decomposition $(\mathscr{P}, \mathcal{X})$ of width $\omega - 1$ that can be computed in linear time, where moreover:*
*(a) $\mathscr{P}$ is a path (hence there are no join nodes);*
*(b) every bag is a clique;*
*(c) going through $\mathscr{P}$ from the leaf to the root, the order in which vertices are introduced in an introduce node corresponds to $<_L$;*
*(d) going through $\mathscr{P}$ from the leaf to the root, the order in which vertices are forgotten in a forget node corresponds to $<_R$;*
*(e) the root's bag is empty, and the leaf's bag contains only one vertex.*

*Proof* Given a graph $G$, one can decide if it is an interval graph and, if so, compute a representation of it in linear time [11]. This also gives us the ordered set of endpoints of intervals of $G$.

To obtain $(\mathscr{P}, \mathcal{X})$, we first create the leaf node $t$, whose bag $X_t$ contains the interval with smallest left endpoint. We then go through the set of all endpoints of intervals of $G$, from the second smallest to the

largest. Let $t$ be the last created node. If the new endpoint is a left endpoint $\ell(I)$, we create an introduce node $t'$ with $X_{t'} = X_t \cup \{I\}$. If the new endpoint is a right endpoint $r(I)$, we create a forget node $t'$ with $X_{t'} = X_t \setminus \{I\}$. In the end we create the root node as a forget node $t$ with $X_t = \emptyset$ that forgets the last interval of $G$.

Observe that one can associate to every node $t$ (except the root) a point $p$ of the real line, such that the bag $X_t$ contains precisely the set of intervals containing $p$: if $t$ is an introduce node, $p$ is the point $\ell(I)$ associated to the creation of $t$, and if $t$ is a forget node, it is the point $r(I) + \epsilon$, where $\epsilon$ is sufficiently small and $r(I)$ is the endpoint associated to the creation of $t$. This set forms a clique, proving Property (b). Furthermore this implies that the maximum size of a bag is $\omega$, hence the width is at most $\omega - 1$ (and at least $\omega - 1$ since every clique must be included in some bag).

Moreover it is clear that the procedure is linear-time, and by construction, Properties (a), (c), (d), (e) are fulfilled.

Let us now show that $(\mathscr{P}, \mathcal{X})$ is a tree-decomposition. It is clear that every vertex belongs to some bag, proving Property (i) of Definition 25. Moreover let $u, v$ be two adjacent vertices of $G$, and assume $u <_L v$. Then, consider the introduce node of $\mathscr{P}$ where $v$ is introduced. Since $u$ has started before $v$ but has not stopped before the start of $v$, both $u, v$ belong to $X_t$, proving Property (ii). Finally, note that a vertex $v$ appears exactly in all bags starting from the bag where $v$ is introduced, until the bag where $v$ is forgotten. Hence Property (iii) is fulfilled, and the proof is complete.   $\square$

The following lemma immediately follows from Proposition 24.

**Lemma 27** *Let $G$ be an interval graph with an interval model inducing orders $<_L$ and $<_R$, let $d \geq 1$ be an integer and let $(\mathscr{P}, \mathcal{X})$ be a tree-decomposition of $G^d$ obtained by Proposition 26 (recall that by Proposition 24, $G^d$ is an interval graph, and it has an intersection model inducing the same orders $<_L$ and $<_R$). Then the following holds.*
*(a) Let $t$ be an introduce node of $(\mathscr{P}, \mathcal{X})$ with child $t'$, with $X_t = X_{t'} \cup \{v\}$. Then, $X_t$ contains every vertex $w$ in $G$ such that $d_G(v, w) \leq d$ and $w <_L v$.*
*(b) Let $t'$ be the child of a forget node $t$ of $(\mathscr{P}, \mathcal{X})$, with $X_t = X_{t'} \setminus \{v\}$. Then, $X_{t'}$ contains every vertex $w$ in $G$ such that $d_G(v, w) \leq d$ and $v <_R w$.*

*Proof* We prove (a), the proof of (b) is the same. By Proposition 24, we may assume that $<_L$ is the same in $G$ and $G^d$. By construction of $(\mathscr{P}, \mathcal{X})$ the introduce node of $v$ contains all intervals $w$ of $G^d$ intersecting $v$ with $w <_L v$ in $G^d$. Hence $w <_L v$ in $G$ as well, and $d_G(v, w) \leq d$.   $\square$

*3.1.3 Lemmas for the algorithm*

We now prove a few preliminary results necessary for the argumentation. We first start with a definition and a series of lemmas based on the linear structure of an interval graph, that will enable us to defer the decision-taking (about which vertex should belong to the solution in order distinguish a specific vertex pair) to later steps of the dynamic programming.

**Definition 28** *Given a vertex $u$ of an interval graph $G$, the* rightmost path $P_R(u)$ *of $u$ is the path $u_0^R, \ldots, u_p^R$ where $u = u_0^R$, for every $u_i^R$ ($i \in \{0, \ldots, p-1\}$) $u_{i+1}^R$ is the neighbour of $u_i^R$ with the largest right endpoint, and thus $u_p^R$ is the interval in $G$ with largest right endpoint. Similarly, we define the* leftmost path $P_L(u) = u_0^L, \ldots, u_q^L$ *where for every $u_i^L$ ($i \in \{0, \ldots, q-1\}$) $u_{i+1}^L$ is the neighbour of $u_i^L$ with the* smallest left *endpoint.*

Note that $P_R(u)$ and $P_L(u)$ are two shortest paths from $u_0^R$ to $u_p^R$ and $u_q^L$, respectively.

**Lemma 29** *Let $u$ be an interval in an interval graph $G$ and $P_R(u) = u_0^R, \ldots, u_p^R$ be the rightmost path of $u$, and let $v$ be an interval starting after the end of $u_{i-1}^R$ ($i \in \{1, \ldots, p\}$), where $u_{i-1}^R \in P_R(u)$. Then $d(u, v) = d(u_i^R, v) + i$. Similarly, if $v$ ends before the start of an interval $u_{i-1}^L$ in $P_L(u) = u_0^L, \ldots, u_q^L$ ($i \in \{1, \ldots, q\}$), then $d(u, v) = d(u_i^L, v) + i$.*

*Proof* We prove the claim only for the first case, the second one is symmetric. Consider the shortest path from $u$ to $v$ by choosing the interval intersecting $u$ that has the largest right endpoint, and iterating. This path coincides with $P_R(u)$ until it contains some interval $u_j^R$ such that $u_j^R$ intersects $v$. Since $v$ starts after the end of $u_{i-1}^R$, we have $i \leq j$. Thus, the interval $u_i^R$ lies on a shortest path from $u$ to $v$, and hence $d(u, v) = d(u_i^R, v) + d(u, u_i^R) = d(u_i^R, v) + i$.   $\square$

**Lemma 30** *Let $u, v$ be a pair of intervals of an interval graph $G$ and $P_R(u) = u_0^R, \ldots, u_p^R$, $P_R(v) = v_0^R, \ldots, v_{p'}^R$ their corresponding rightmost paths (recall that $u_p^R = v_{p'}^R$). Assuming that $p \leq p'$, for every $u_i^R \in P_R(u)$ and $v_i^R \in P_R(v)$ such that $i \in \{0, \ldots, p\}$, we have $d(u_i^R, v_i^R) \leq d(u, v)$.*

*Proof* First note that, by letting $w = u_i^R$, we have $w_1^R = u_{i+1}^R$. Therefore, we only need to prove the claim for $i = 1$.

If $u$ and $v$ are adjacent, then either $v = u_1^R$ (then we are done) or $u_1^R$ must end after $v$. Then, either $u_1^R$ intersects $v_1^R$, or $u_1^R = v_1^R$. In both cases, $d(u_1^R, v_1^R) \leq 1$.

If $u$ and $v$ are not adjacent, we can assume that $u$ ends before $v$ starts. Then, by Lemma 29, $d(u_1^R, v) = d(u, v) - 1$ and $d(u_1^R, v_1^R) \leq d(u_1^R, v) + d(v, v_1^R) = d(u, v) - 1 + 1 = d(u, v)$. ☐

We say that a pair $u, v$ of intervals in an interval graph $G$ is separated by interval $x$ *strictly from the right* (*strictly from the left*, respectively) if $x$ starts after both right endpoints of $u, v$ (ends before both left endpoints of $u, v$ respectively). In other words, $x$ is not a neighbour of any of $u$ and $v$.

The next lemma is crucial for our algorithm.

**Lemma 31** *Let $u, v, x$ be three intervals in an interval graph $G$ and let $i$ be an integer such that $x$ starts after both right endpoints of $u_i^R \in P_R(u)$ and $v_i^R \in P_R(v)$. Then the three following facts are equivalent:*
*(1) $x$ separates $u_i^R$, $v_i^R$;*
*(2) for every $j$ with $0 \leq j \leq i$, $x$ separates $u_j^R$, $v_j^R$;*
*(3) for some $j$ with $0 \leq j \leq i$, $x$ separates $u_j^R$, $v_j^R$.*
*Similarly, assume that $x$ ends before both left endpoints of $u_i^L \in P_L(u)$ and $v_i^L \in P_L(v)$. Then the three following facts are equivalent:*
*(i) $x$ separates $u_i^L$, $v_i^L$;*
*(ii) for every $j$ with $0 \leq j \leq i$, $x$ separates $u_j^L$, $v_j^L$;*
*(iii) for some $j$ with $0 \leq j \leq i$, $x$ separates $u_j^L$, $v_j^L$.*

*Proof* We prove only (1)–(3), the proof of (i)–(iii) is symmetric. Let $0 \leq j \leq i$ and $u' = u_j^R$ and $v' = v_j^R$. Then $(u')_{i-j}^R = u_i^R$ and $(v')_{i-j}^R = v_i^R$. By Lemma 29, $d(u_j^R, x) = d(u_i^R, x) + (j - i)$ and similarly $d(v_j^R, x) = d(v_i^R, x) + (j - i)$. Hence $x$ separates $u_i^R$ and $v_i^R$ if and only if it separates $u_j^R$ and $v_j^R$ which implies the lemma. ☐

We now introduce a local version of resolving sets that will be used in our algorithm.

**Definition 32** *A* distance-2 resolving set *is a set $S$ of vertices where for each pair $u, v$ of vertices at distance at most 2, there is a vertex $x \in S$ such that $d(u, s) \neq d(v, s)$.*

Using the following lemma, we can manage to "localize" the dynamic programming, as we will only need to distinguish pairs of vertices that will be present together in one bag.

**Lemma 33** *Any distance-2 resolving set of an interval graph $G$ is a resolving set of $G$.*

*Proof* Assume to the contrary that $S$ is a distance-2 resolving set of an interval graph $G$ but not a resolving set. It means that there is a pair of vertices $u, v$ at distance at least 3 that are not separated by any vertex of $S$. Among all such pairs, we choose one, say $\{u, v\}$, such that $d(u, v)$ is minimized. Without loss of generality, we assume that $u$ ends before $v$ starts.

Consider $u_1^R$ ($v_1^L$, respectively), the interval intersecting $u$ ($v$, respectively) that has the largest right endpoint (smallest left endpoint, respectively). We have $u_1^R \neq v_1^L$ (since $d(u, v) \geq 3$) and $d(u_1^R, v_1^L) = d(u, v) - 2 < d(u, v)$. By minimality, $u_1^R$ and $v_1^L$ are separated by some vertex $s \in S$. But $s$ does not separate $u$ and $v$, thus $s \notin \{u_1^R, v_1^L\}$.

Without loss of generality, we can assume that $d(u_1^R, s) < d(v_1^L, s)$. In particular, $d(v_1^L, s) \geq 2$ and $s$ is ending before $v_1^L$ starts. Thus, by Lemma 29, $d(v, s) = d(v_1^L, s) + 1$. However, we also have $d(u, s) \leq d(u_1^R, s) + 1 \leq d(v_1^L, s) < d(v, s)$. Hence $s$ is separating $u$ and $v$, a contradiction. ☐

The next lemma, which is a slightly modified version of a result in our paper [29], enables us to upper-bound the size of the bags in our tree-decompositions, which will induce diameter 4-subgraphs of $G$.

**Lemma 34** *Let $G$ be an interval graph with a resolving set of size $k$, and let $B \subseteq V(G)$ be a subset of vertices such that for each pair $u, v \in B$, $d_G(u, v) \leq d$. Then $|B| \leq 4dk^2 + (2d + 3)k + 1$.*

*Proof* Let $s_1, \ldots, s_k$ be the elements of a resolving set $S$ of size $k$ in $G$. Consider an interval representation of $G$, and let $\mathcal{B}$ be the minimal segment of the real line containing all intervals corresponding to vertices of $B$.

For each $i$ in $\{1, \ldots, k\}$, consider the leftmost and rightmost paths $P_L(s_i)$ and $P_R(s_i)$, as defined in Definition 28. Let $L^i$ be the ordered set of left endpoints of intervals of $P_L(s_i)$, and let $R^i$ be the ordered set of right endpoints of intervals of $P_R(s_i)$. Note that intervals at distance $j$ of $s_i$ in $G$ are exactly the intervals finishing between $\ell(u_{j+1}^L)$ and $\ell(u_j^L)$, or starting between $r(u_j^R)$ and $r(u_{j+1}^R)$. Hence, for any interval of $G$, its distance to $s_i$ is uniquely determined by the position of its right endpoint in the ordered set $L^i$ and the position of its left endpoint in the ordered set $R^i$. Moreover, note that, since any two vertices in $B$ are at distance at most $d$, $\mathcal{B}$ may contain at most $d$ points of $L^i$ and at most $d$ points of $R^i$.

Therefore, $\mathcal{B}$ may contain at most $2kd$ points of $\bigcup_{1 \leq i \leq k}(L^i \cup R^i)$. This set of points defines a natural partition $\mathcal{P}$ of $\mathcal{B}$ into at most $2kd + 1$ sub-segments, and any interval of $B$ is uniquely determined by the positions of its two endpoints in $\mathcal{P}$ (if two intervals start and end in the same part of $\mathcal{P}$, they are not separated by $S$, a contradiction).

Let $I \in B \setminus S$. For a fixed $i \in \{1, \ldots, k\}$, by definition of the sets $L^i$, the interval $I$ cannot contain two points of $L^i$ and similarly, it cannot contain two points of $R^i$. Thus, $I$ contains at most $2k$ points of the union of all the sets $L^i$ and $R^i$. Therefore, if $P$ denotes a part of $\mathcal{P}$, there are at most $2k + 1$ intervals with left endpoints in $P$. In total, there are at most $(2kd + 1) \cdot (2k + 1)$ intervals in $B \setminus S$ and hence $|B| \leq (2kd + 1) \cdot (2k + 1) + k = 4dk^2 + (2d + 3)k + 1$. □

3.2 The algorithm

We are now ready to describe our algorithm.

**Theorem 35** METRIC DIMENSION *can be solved in time $2^{O(k^4)}n$ on interval graphs, i.e. it is FPT on this class when parameterized by the solution size $k$.*

*Proof* Let $(\mathscr{P}, \mathcal{X})$ be a path-decomposition of $G^4$ (which by Proposition 24 is an interval graph) obtained using Proposition 26.

The algorithm is a bottom-up dynamic programming on $(\mathscr{P}, \mathcal{X})$. By Proposition 26(b), every bag of $(\mathscr{P}, \mathcal{X})$ is a clique of $G^4$ (i.e. an induced subgraph of diameter at most 4 in $G$) and hence by Lemma 34, it has $O(k^2)$ vertices. Thanks to Lemma 31, we can "localize" the problem by considering for separation, only pairs of vertices present together in the current bag. Let us now be more precise.

For a node $t$ in $\mathscr{P}$, we denote by $\mathcal{P}(X_t)$ the pairs of intervals in $X_t$ that are at distance at most 2 (in $G$).

For each node $t$, we compute a set of *configurations* using the configurations of the child of $t$ in $\mathscr{P}$. A configuration contains full information about the local solution on $X_t$, but also stores necessary information about the vertex pairs that still need to be separated. More precisely, a configuration $C = (\mathsf{S}, \mathsf{sep}, \mathsf{toSepR}, \mathsf{cnt})$ of $t$ is a tuple where:

- $\mathsf{S} \subseteq X_t$ contains the vertices of the sought solution belonging to $X_t$;
- $\mathsf{sep} : \mathcal{P}(X_t) \to \{0, 1, 2\}$ assigns, to every pair in $\mathcal{P}(X_t)$, value 0 if the pair has not yet been separated, value 2 if it has been separated strictly from the left, and value 1 otherwise;
- $\mathsf{toSepR} : \mathcal{P}(X_t) \to \{0, 1\}$ assigns, to every pair in $\mathcal{P}(X_t)$, value 1 if the pair needs to be separated strictly from the right (and it is not yet separated), and value 0 otherwise;
- $\mathsf{cnt}$ is an integer counting the total number of vertices in the partial solution that has led to $C$.

Starting with the leaf of $\mathscr{P}$, for each node our algorithm goes through all possibilities of choosing $\mathsf{S}$; however, $\mathsf{sep}$, $\mathsf{toSepR}$ and $\mathsf{cnt}$ are computed along the way. At each new visited node $t$ of $\mathscr{P}$, a set of configurations is constructed from the configuration sets of the child of $t$. The algorithm makes sure that all the information is consistent, and that configurations that will not lead to a valid resolving set (or with $\mathsf{cnt} > k$) are discarded.

**Leaf node:** For the leaf node $t$, since by Proposition 26(e) $X_t = \{v\}$, we create two configurations $C_1 = (\emptyset, \mathsf{sep}, \mathsf{toSepR}, 0)$ and $C_2 = (\{v\}, \mathsf{sep}, \mathsf{toSepR}, 1)$ (where $\mathsf{sep}$ and $\mathsf{toSepR}$ are empty in both configurations).

**Introduce node:** Let $t$ be an introduce node with $t'$ its child, where $X_t = X_{t'} \cup \{v\}$. For every configuration $(\mathsf{S}', \mathsf{sep}', \mathsf{toSepR}', \mathsf{cnt}')$ of $t'$, we create two configurations $C_1 = (\mathsf{S}' \cup \{v\}, \mathsf{sep}_1, \mathsf{toSepR}_1, \mathsf{cnt}' + 1)$ (corresponding to the case where $v$ is in the partial solution) and $C_2 = (\mathsf{S}', \mathsf{sep}_2, \mathsf{toSepR}_2, \mathsf{cnt}')$ (where $v$ is not added to the partial solution).

The elements of $\mathsf{sep}_1$ and $\mathsf{toSepR}_1$ in $C_1$ are first copied from $\mathsf{sep}'$ and $\mathsf{toSepR}'$, and updated by checking, for every pair $x, y$ of $\mathcal{P}(X_t)$ whether $v$ separates $x, y$ (note that $v$ cannot separate any such pair strictly from the left). Also note that $v$ is separated from all other vertices since it belongs to the solution, but for $x = v$ we still need to check whether $v, y$ are strictly separated from the left (in which case we set $\mathsf{sep}_1(v, y) = 2$, otherwise $\mathsf{sep}_1(v, y) = 1$). To do this, we compute $v_1^L$ and $y_1^L$ (by Lemma 27(a) they both belong to $X_t$), and we first check if they are strictly separated from the left, which is true if and only if $\mathsf{sep}'(v_1^L, y_1^L) = 2$. If $v_1^L$ and $y_1^L$ are separated strictly from the left, then so are $v$ and $y$. Otherwise, if $v$ and $y$ are still strictly separated from the left, there must be an interval $z$ ending before the left endpoint of $y$ and separating $v, y$. Since $z$ does not separate $v_1^L$ and $y_1^L$ strictly from the left, $z$ must be adjacent to $y_1^L$ and thus $d_G(v, z) \leq 4$ (since $d_G(v, y) \leq 2$). Then, by Lemma 27, $z$ belongs to $X_t$, thus it is enough to test whether any vertex of $\mathsf{S}'$ separates $v, y$ strictly from the left. Moreover, we let $\mathsf{toSepR}_1(v, y) = 0$.

For $C_2$, we must compute $\mathsf{sep}_2(v, w)$ and $\mathsf{toSepR}_2(v, w)$ for every $w$ such that $(v, w) \in \mathcal{P}(X_t)$. To do so, we consider the first intervals of $P_L(v)$ and $P_L(w)$. We let $\mathsf{sep}_2(v, w) = 2$ if for the pair $v_1^L, w_1^L$ with $v_1^L \in P_L(v)$ and $w_1^L \in P_L(w)$, $\mathsf{sep}'(v_1^L, w_1^L) = 2$, or if some vertex of $\mathsf{S}'$ separates $v, w$ strictly from the left. Otherwise, if $v, w$ are separated by a neighbour of $w$, we set $\mathsf{sep}_2(v, w) = 1$. We also compute $\mathsf{toSepR}_2$ from $\mathsf{toSepR}'$ by letting $\mathsf{toSepR}_2(v, w) = 0$ and copying all other values.

If $\mathsf{cnt} + 1 > k$, $C_1$ is discarded. The remaining valid configurations among $C_1, C_2$ are added to the set of configurations of $t$. If in this set, there are two configurations that differ only on their value of $\mathsf{cnt}$, we only keep the one with the smallest value of $\mathsf{cnt}$.

**Forget node:** Let $t$ be a forget node and $t'$ be its child, with $X_t = X_{t'} \setminus \{v\}$. For every configuration $(\mathsf{S}', \mathsf{sep}', \mathsf{toSepR}', \mathsf{cnt}')$ of $t'$, we create the configuration $(\mathsf{S}' \setminus \{v\}, \mathsf{sep}, \mathsf{toSepR}, \mathsf{cnt}')$. We create $\mathsf{sep}$ and $\mathsf{toSepR}$ by copying all entries $\mathsf{sep}'(x, y)$ and $\mathsf{toSepR}'(x, y)$ such that $x, y \in \mathcal{P}(X_t)$.

For every vertex $w$ in $X_t$ such that $d_G(v, w) \leq 2$, if $\mathsf{sep}'(v, w) = 0$ or $\mathsf{toSepR}'(v, w) = 1$ (i.e. $v, w$ still need to be separated strictly from the right), we determine $v_1^R$ and $w_1^R$ and let $\mathsf{toSepR}(v_1^R, w_1^R) = 1$ (note that $d_G(v, v_1^R) = 1$, $d_G(v, w_1^R) \leq 3$, $v <_R v_1^R$ and $v <_R w_1^R$, hence by Lemma 27(b) $v_1^R, w_1^R \in X_{t'}$ and hence $v_1^R, w_1^R \in X_t$). However, if $v_1^R = w_1^R$, we discard the current configuration. Indeed, by Lemma 31, $v, w$ cannot be separated strictly from the right: any shortest path to any of $v, w$ from some vertex $x$ whose interval starts after both right endpoints of $v, w$ must go through $v_1^R = w_1^R$ and hence $d(x, v_1^R) = d(x, w_1^R)$. We also discard the configuration if $v_1^R$ or $w_1^R$ does not exist (i.e. $v$ or $w$ is the rightmost interval of $G$).

Finally, if there are two configurations that differ only on their value of $\mathsf{cnt}$, again we only keep the one with the smallest value of $\mathsf{cnt}$.

**Root node:** At root node $t$, since by Proposition 26(e) $X_t = \emptyset$, $t$ has at most one configuration. We output "yes" only if this configuration exists, and if $\mathsf{cnt} \leq k$. Otherwise, we output "no".

We now analyze the algorithm.

**Correctness.** We claim that $G$ has a resolving set of size at most $k$ if and only if the root node of $\mathscr{P}$ contains a valid configuration. By Lemma 33, this is equivalent to proving that $G$ has an optimal *distance-2 resolving set* of size at most $k$ if and only if the root node of $\mathscr{P}$ contains a valid configuration. First, assume that the dynamic programming has succeeded, i.e. the root bag contains a valid configuration $C$. Assume that $C$ has smallest value $\mathsf{cnt}$. We want to prove that the union of all partial solutions $\mathsf{S}$ of all configurations that have led to the computation of $C$ is a valid optimal solution $S$.

We first prove that for every pair $u, v$ of vertices with $d_G(u, v) \leq 2$ and $u <_R v$, $S$ separates $u, v$. By Lemma 27(b), $u, v$ are present together in the child $t'$ of forget node $t$ of $\mathscr{P}$ where $u$ is forgotten. Let $C_{t'} = (\mathsf{S}', \mathsf{sep}', \mathsf{toSepR}', \mathsf{cnt}')$ and $C_t = (\mathsf{S}, \mathsf{sep}, \mathsf{toSepR}, \mathsf{cnt})$ be the configurations of $t', t$ that have led to the end configuration $C$. In the computation of $C_t$, since $C_t$ was not discarded, we either had $\mathsf{sep}'(u, v) > 0$ in $C_{t'}$ or the algorithm has set $\mathsf{toSepR}(u_r^1, v_r^1) = 1$, in which case $u_1^R \neq v_1^R$. Assume we had $\mathsf{sep}'(u, v) = 1$. Then, in some configuration $C_{t''}$ that has led to computing $C_{t'}$ (possibly $t' = t''$), $u$ and $v$ were separated by some vertex in $S$ belonging to $C_{t''}$, and we are done. If $\mathsf{sep}'(u, v) = 2$, similarly either $u, v$ have been separated by some vertex of $S$ belonging to a (possibly earlier) configuration, or we had $\mathsf{sep}(u_i^L, v_i^L) = 2$, in which case by Lemma 31 we are also done. If however, the algorithm has set $\mathsf{toSepR}(u_1^R, v_1^R) = 1$, recall that unless in some bag $u_1^R, v_1^R$ is separated strictly from the right, when

we forget $u_1^R$ we set $\mathsf{toSepR}(u_2^R, v_2^R) = 1$. Hence, since $C$ was a valid configuration (and has not been discarded), at some step we have separated $u_i^R, v_i^R$ strictly from the right, which by Lemma 31 implies that $u, v$ are separated by $S$, and we are done.

Moreover $S$ is optimal because we have chosen $C$ so as to minimize the size $\mathsf{cnt}$ of the overall solution. At each step, the algorithm discards, among equivalent configurations, the ones with larger values of $\mathsf{cnt}$, ensuring that the size of the solution is minimized. This proves our claim.

For the converse, assume that $G$ has an optimal distance-2 resolving set $S$ of size at most $k$. We will need the following claim.

**Claim 36** *Let $u, v$ be a pair of vertices with $d_G(u, v) \leq 2$. Then, any vertex $x$ that could separate $u, v$ neither strictly from the right nor strictly from the left is present in some bag together with both $u, v$.*

*Proof of claim.* Necessarily, $x$ is a neighbour of one of $u, v$ in $G$. Hence $d_G(x, u) \leq 3$ and $d_G(x, v) \leq 3$. If $x <_L v$, by Lemma 27(a) $x, u, v$ are present in the bag where $v$ is introduced. If $v <_L x$, similarly $x, u, v$ are present in the bag where $x$ is introduced. $\hspace{1cm} (\diamond)$

We will prove that some configuration $C$ was computed using a series of configurations where for each node $t$ of $\mathscr{P}$, the right subset $S \cap X_t$ was guessed. By contradiction, if this was not the case, then at some step of the algorithm we would have discarded a configuration $C'$ although it arised from guessing the correct partial solution of $S$. Since $S$ is optimal, $C'$ was not discarded because there was a copy of $C'$ with different value of counter $\mathsf{cnt}$ (otherwise this copy would lead to a solution strictly smaller than $S$). Hence the discarding of $C'$ has happened at a node $t$ that is a forget node. Assume that $t$ is a forget node where vertex $v$ was forgotten (assume $t'$ is the child of $t$ in $\mathscr{P}$). This happens only if for some $w \in X_t$ with $d_G(v, w) \leq 2$, we had either (i) $\mathsf{sep}'(v, w) = 0$ and $v_1^R = w_1^R$, or (ii) $\mathsf{toSepR}(v, w) = 1$ and $v_1^R = w_1^R$. If (i) holds, then $v, w$ are considered not to be separated, although they are actually separated (by our assumption on $C'$). Since $v_1^R = w_1^R$, $v_1^R$ and $w_1^R$ cannot be separated strictly from the right, hence by Lemma 31 $v, w$ are not separated strictly from the right. If they are not separated strictly from the left, Claim 36 implies a contradiction because the vertex separating $v, w$ was present together in a bag with $v, w$ and hence we must have $\mathsf{sep}'(v, w) = 1$. Hence, $v, w$ are separated strictly from the left. But again by Lemma 31, this means that some vertices $v_i^L, w_i^L$ in $P_R(v) \times P_R(w)$ have been separated strictly from the left (assume that $i$ is maximal with this property). Since by Lemma 30, $d_G(v_i^L, w_i^L) \leq 2$, by Lemma 27 these two vertices were present in some bag simultaneously, together with the vertex that is strictly separating them from the left (and has distance at most 4 from $w_i^L$). Then in the configuration corresponding to this bag, $\mathsf{sep}(v_i^L, w_i^L) = 2$, and we had $\mathsf{sep}'(v, w) = 2$ in $C'$, a contradiction. If (ii) holds, there exists a pair $x, y$ such that in some earlier configuration, we had $\mathsf{sep}(x, y) = 0$, $v = x_i^R \in P_R(x)$ and $w = y_i^R \in P_R(y)$. By the same reasoning as for (i) we obtain a contradiction. This proves this side of the implication, and completes the proof of correctness.

**Running time.** At each step of the dynamic programming, we compute the configurations of a bag from the set of configurations of the child bag. The computation of each configuration is polynomial in the size of the current bag of $(\mathscr{P}, \mathcal{X})$. Since a configuration is precisely determined by a tuple $(S, \mathsf{sep}, \mathsf{toSepR})$ (if there are two configurations where only $\mathsf{cnt}$ differs, we only keep the one with smallest value), there are at most $2^{|X_t|} 3^{|X_t|^2} 2^{|X_t|^2} \leq 3^{2|X_t|^2}$ configurations for a bag $X_t$. Hence, in total the running time is upper-bounded by $2^{O(b^2)} n$, where $b$ is the maximum size of a bag in $(\mathscr{P}, \mathcal{X})$. Since any bag induces a subgraph of $G$ of diameter at most 4, by Lemma 34, $b = O(k^2)$. Therefore $2^{O(b^2)} n = 2^{O(k^4)} n$, as claimed. $\hspace{1cm} \square$

## 4 Conclusion

We proved that LOCATING-DOMINATING-SET, OPEN LOCATING-DOMINATING SET, IDENTIFYING CODE and METRIC DIMENSION are NP-complete even for interval graphs that have diameter 2 and for permutation graphs that have diameter 2. This is in contrast to related problems such as DOMINATING SET, which is linear-time solvable both on interval graphs and on permutation graphs. However, we do not know their complexity for unit interval graphs or bipartite permutation graphs. Note that both LOCATING-DOMINATING-SET and METRIC DIMENSION are polynomial-time solvable on chain graphs, a subclass of bipartite permutation graphs [25]. Probably the same approach as in [25] would also work for OPEN LOCATING-DOMINATING SET and IDENTIFYING CODE.

Contrary to what we claimed in the conference version of this paper [28], our reduction gadgets are not interval graphs and permutation graphs at the same time. Hence, we leave it as an open question to determine the complexity of the studied problems when restricted to graphs which are both interval and permutation graphs. Similarly, it could be interesting to determine their complexity for graphs that are both split graphs and interval graphs, or split graphs and permutation graphs.

We remark that our generic reduction would also apply to related problems that have been considered in the literature, such as LOCATING-TOTAL DOMINATING SET [37] or DIFFERENTIATING-TOTAL DOMINATING SET [17].

Regarding our positive result that METRIC DIMENSION parameterized by the solution size is FPT on interval graphs, an interesting question is whether it can be extended to other graph classes, such as permutation graphs. Another interesting class is the one of chordal graph, since it is a proper superclass of both interval graphs and split graphs, both of which admit an FPT algorithm for METRIC DIMENSION. During the revision of this paper, it was brought to our knowledge that in a recent paper, Belmonte, Fomin, Golovach and Ramanujan [6] have answered these questions by showing that for any class of graphs of bounded tree-length, METRIC DIMENSION is FPT when parameterized by the solution size. Examples of such classes are the ones of chordal graphs, asteroidal triple-free graphs and permutation graphs.

# References

1. G. Agnarsson, P. Damaschke and M. M. Halldórsson. Powers of geometric intersection graphs and dispersion algorithms. *Discrete Applied Mathematics* 132(1–3):3–16, 2003.
2. D. Auger. Minimal identifying codes in trees and planar graphs with large girth. *European Journal of Combinatorics* 31(5):1372–1384, 2010.
3. L. Babai. On the complexity of canonical labeling of strongly regular graphs. *SIAM Journal of Computing* 9(1):212–216, 1980.
4. E. Bampas, D. Bilò, G. Drovandi, L. Gualà, R. Klasing and G. Proietti. Network verification via routing table queries. *Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2011*, LNCS 6796:270–281, 2011.
5. Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihalák and L. S. Ram. Network discovery and verification. *IEEE Journal on Selected Areas in Communications* 24(12):2168–2181, 2006.
6. R. Belmonte, F. V. Fomin, P. A. Golovach, M. S. Ramanujan: Metric dimension of bounded width graphs. *Proceedings of 40th International Symposium of Mathematical Foundations of Computer Science, MFCS 2015*, LNCS 9235:115–126, 2015.
7. 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.
8. N. Bertrand, I. Charon, O. Hudry and A. Lobstein. 1-identifying codes on trees. *Australasian Journal of Combinatorics* 31:21–35, 2005.
9. B. Bollobás and A. D. Scott. On separating systems. *European Journal of Combinatorics* 28:1068–1071, 2007.
10. J. A. Bondy. Induced subsets. *Journal of Combinatorial Theory, Series B* 12(2):201–202, 1972.
11. K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences* 13(3):335–379, 1976.
12. N. Bousquet, A. Lagoutte, Z. Li, A. Parreau and S. Thomassé. Identifying codes in hereditary classes of graphs and VC-dimension. *SIAM Journal on Discrete Mathematics* 29(4):2047–2064, 2015.
13. A. Brandstädt, V. B. Le and J. Spinrad. *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications, 1999.
14. 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.
15. 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* 2(4):403–420, 2008.
16. 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.
17. M. Chellali. On locating and differetiating-total domination in trees. *Discussiones Mathematicae Graph Theory* 28(3): 383–392, 2008.
18. G. Cohen, I. Honkala, A. Lobstein and G. Zémor. On identifying codes. *Proceedings of the DIMACS Workshop on Codes and Association Schemes*, Series in Discrete Mathematics and Theoretical Computer Science 56:97–109, 2001.
19. C. Colbourn, P. J. Slater and L. K. Stewart. Locating-dominating sets in series-parallel networks. *Congressus Numerantium* 56:135–162, 1987.
20. B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation* 85(1):12–75, 1990.

21. J. Diaz, O. Pottonen, M. Serna and E. J. van Leeuwen. On the complexity of metric dimension. *Proceedings of the 20th European Symposium on Algorithms, ESA 2012*, LNCS 7501:419–430, 2012.
22. R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
23. D. Eppstein. Metric dimension parameterized by max leaf number. *Journal of Graph Algorithms and Applications* 19(1):313–323, 2015.
24. L. Epstein, A. Levin and G. J. Woeginger. The (weighted) metric dimension of graphs: hard and easy cases. *Algorithmica* 72(4):1130–1171, 2015.
25. H. Fernau, P. Heggernes, P. van't Hof, D. Meister and R. Saei. Computing the metric dimension for chain graphs. *Information Processing Letters* 115:671–676, 2015.
26. F. Foucaud. Decision and approximation complexity for identifying codes and locating-dominating sets in restricted graph classes. *Journal of Discrete Algorithms* 31:48–68, 2015.
27. F. Foucaud, S. Gravier, R. Naserasr, A. Parreau and P. Valicov. Identifying codes in line graphs. *Journal of Graph Theory* 73(4):425–448, 2013.
28. F. Foucaud, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov. Algorithms and complexity for metric dimension and location-domination on interval and permutation graphs. *Proceedings of the 41st International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2015*, LNCS, to appear.
29. F. Foucaud, G. Mertzios, R. Naserasr, A. Parreau and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. I. Bounds. `http://arxiv.org/abs/1507.08164`, 2015.
30. M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman, 1979.
31. M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, Elsevier, 2004.
32. 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.
33. M. Habib and C. Paul. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics* 145(2):183–197, 2005.
34. F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combinatoria* 2:191–195, 1976.
35. S. Hartung. *Exploring parameter spaces in coping with computational intractability*. PhD Thesis, TU Berlin, Germany, 2014.
36. S. Hartung and A. Nichterlein. On the parameterized and approximation hardness of metric dimension. *Proceedings of the IEEE Conference on Computational Complexity, CCC 2013*:266–276, 2013.
37. M. A. Henning and N. J. Rad. Locating-total domination in graphs. *Discrete Applied Mathematics* 160:1986–1993, 2012.
38. M. A. H. Henning and A. Yeo. Distinguishing-transversal in hypergraphs and identifying open codes in cubic graphs. *Graphs and Combinatorics* 30(4):909–932, 2014.
39. S. Hoffmann and E. Wanke. Metric dimension for Gabriel unit disk graphs is NP-Complete. *Proceedings of ALGOSEN-SORS 2012*:90-92, 2012.
40. 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.
41. 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.
42. S. Khuller, B. Raghavachari and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics* 70(3):217–229, 1996.
43. J. H. Kim, O. Pikhurko, J. Spencer and O. Verbitsky. How complex are random graphs in First Order logic? *Random Structures and Algorithms* 26(1-2):119–145, 2005.
44. T. Kloks. *Treewidth, Computations and Approximations.* Springer, 1994.
45. P. Manuel, B. Rajan, I. Rajasingh and Chris Monica M. On minimum metric dimension of honeycomb networks. *Journal of Discrete Algorithms* 6(1):20–27, 2008.
46. 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.
47. T. Müller and J.-S. Sereni. Identifying and locating-dominating codes in (random) geometric networks. *Combinatorics, Probability and Computing* 18(6):925–952, 2009.
48. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
49. A. Rényi. On random generating elements of a finite Boolean algebra. *Acta Scientiarum Mathematicarum Szeged* 22:75–81, 1961.
50. S. J. Seo and P. J. Slater. Open neighborhood locating-dominating sets. *The Australasian Journal of Combinatorics* 46:109–120, 2010.
51. P. J. Slater. Leaves of trees. *Congressus Numerantium* 14:549–559, 1975.
52. P. J. Slater. Domination and location in acyclic graphs. *Networks* 17(1):55–64, 1987.
53. P. J. Slater. Dominating and reference sets in a graph. *Journal of Mathematical and Physical Sciences* 22(4):445–455, 1988.
54. J. Spinrad. Bipartite permutation graphs. *Discrete Applied Mathematics* 18(3):279–292, 1987.
55. J. Suomela. Approximability of identifying codes and locating-dominating codes. *Information Processing Letters* 103(1):28–33, 2007.
56. 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*, LNCS 3283:175–189, 2004.