Received: 12 October 2022 | Revised: 16 October 2023 | Accepted: 23 November 2023

DOI: 10.1002/jgt.23065

ARTICLE

WILEY

Classes of intersection digraphs with good algorithmic properties

Lars Jaffke[1] | O-joung Kwon[2,3] | Jan Arne Telle[1]

[1]Department of Informatics, University of Bergen, Bergen, Norway

[2]Department of Mathematics, Hanyang University, Seoul, South Korea

[3]Discrete Mathematics Group, Institute for Basic Science, Daejeon, South Korea

Correspondence
Lars Jaffke, Department of Informatics, University of Bergen, Bergen, Norway.
Email: lars.jaffke@uib.no

Funding information
Norwegian Research Council (Norges Forskningsråd), Grant/Award Number: 274526; National Research Foundation of Korea (NRF), Grant/Award Numbers: NRF-2021K2A9A2A11101617, RS-2023-00211670; Institute for Basic Science, Grant/Award Number: IBS-R029-C1

Abstract

While intersection graphs play a central role in the algorithmic analysis of hard problems on undirected graphs, the role of intersection *digraphs* in algorithms is much less understood. We present several contributions towards a better understanding of the algorithmic treatment of intersection digraphs. First, we introduce natural classes of intersection digraphs that generalize several classes studied in the literature. Second, we define the directed locally checkable vertex (DLCV) problems, which capture many well-studied problems on digraphs, such as (Independent) Dominating Set, Kernel, and H-Homomorphism. Third, we give a new width measure of digraphs, *bi-mim-width*, and show that the DLCV problems are polynomial-time solvable when we are provided a decomposition of small bi-mim-width. Fourth, we show that several classes of intersection digraphs have bounded bi-mim-width, implying that we can solve all DLCV problems on these classes in polynomial time given an intersection representation of the input digraph. We identify reflexivity as a useful condition to obtain intersection digraph classes of bounded bi-mim-width, and therefore to obtain positive algorithmic results.

KEYWORDS

directed domination, directed H-homomorphism, H-digraphs, intersection digraphs, reflexive digraphs

1 | INTRODUCTION

The computational intractability of graph problems is often dealt with by restricting the input graph to be a member of some graph class and exploiting the structural properties of this class to design efficient algorithms. Intersection graph classes are an extensively studied family of classes of undirected graphs where vertices are represented by sets with two vertices being adjacent if and only if their corresponding sets intersect. For instance, a graph is an *interval graph* if it is an intersection graph of intervals on a line. The literature on algorithmic aspects of classes of intersection graphs is vast, and we refer to [13] for an overview. Even though the concept of intersection *digraphs* has already been introduced in the early 1980s [9], these classes of directed graphs have not received nearly as much attention in the algorithmic literature as their undirected counterparts. That is not to say that they have not been considered before; for instance, interval digraphs [44], circular-arc digraphs [43], and permutation digraphs [35] have been introduced quite early on.

Formally, a digraph G is an *intersection digraph* if there exists a family $\{(S_v, T_v) : v \in V(G)\}$ of ordered pairs of sets such that there is an edge from v to w in G if and only if S_v intersects T_w. Note that we add a loop on a vertex v if S_v and T_v intersect. Even for interval digraphs, a natural starting point for the investigation of algorithmic properties of intersection digraphs, no algorithmic applications are known besides a polynomial-time recognition algorithm of the class [35]. One possible explanation for this is that the class of interval digraphs appears to be much richer than their undirected counterparts. We observe that interval digraphs contain, for each integer n, some orientation of the $(n \times n)$-grid (see Proposition 3); in contrast, interval graphs do not contain an induced subgraph isomorphic to the 1-subdivision of the claw. This shows that the underlying undirected graphs of interval digraphs are very different from interval graphs.

The case of interval digraphs suggests that further structural restrictions are necessary to make classes of intersection digraphs amenable for algorithmic treatment. In this vein, restrictions of interval digraphs have been considered in the literature [20, 39] with applications to digraph problems, such as Independent Dominating Set, Kernel, and List Homomorphism. A common feature of the restrictions considered in [20, 39] is that the digraphs are *reflexive*, meaning that each vertex has a loop. Note that for a class of intersection digraphs, reflexivity gives much more additional structure than just added loops.

In this work, we give a host of algorithmic applications of intersection digraph classes, in the following manner:

- We give new and more general classes of intersection digraphs, namely, H-digraphs, rooted directed path digraphs, and H-convex digraphs. (See the discussion below Theorem 3 for definitions.)
- We introduce directed analogs of the locally checkable vertex problems [48], which include many well-studied digraph problems, such as (Independent) Dominating Set, Kernel, H-Homomorphism, and Oriented k-Coloring, see Tables 1 and 2.
- We define a new width measure of digraphs, called *bi-mim-width*, and prove that the directed locally checkable vertex (DLCV) problems can be solved in polynomial time when a decomposition of bounded bi-mim-width of the input graph is given.
- We prove that fairly general subclasses of these intersection digraph classes have bounded bi-mim-width, see Figure 1.

Note in particular that the last item implies that given a representation of the input digraph, all directed locally checkable problems are solvable in polynomial time on the classes of

TABLE 1 Examples of $(\sigma^+, \sigma^-, \rho^+, \rho^-)$-sets, represented by finite or cofinite sets.

σ^+	σ^-	ρ^+	ρ^-	**Standard name**
$\{0\}$	$\{0\}$	$\mathbb{N}\setminus\{0\}$	$\mathbb{N}$	Kernel [50]
$\{0, \ldots, k-1\}$	$\{0\}$	$\{i : i \geq l\}$	$\mathbb{N}$	(k, l)-out Kernel [40]
$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	Dominating set [25]
$\{0\}$	$\{0\}$	$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	Independent Dominating set [16]
$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	$\mathbb{N}$	In-Dominating set/Absorbing set [23]
$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	$\mathbb{N}\setminus\{0\}$	Twin Dominating set [17]
$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\{i : i \geq k\}$	k-Dominating set [36]
$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	$\mathbb{N}$	$\mathbb{N}\setminus\{0\}$	Total Dominating set [2]
$\{0\}$	$\{0\}$	$\mathbb{N}$	$\{1\}$	Efficient (Closed) Dominating set [8]
$\mathbb{N}$	$\{1\}$	$\mathbb{N}$	$\{1\}$	Efficient Total Dominating set [42]
$\{k\}$	$\{k\}$	$\mathbb{N}$	$\mathbb{N}$	k-Regular Induced Subgraph [15]

Note: For any row there is an associated NP-complete problem, usually maximizing or minimizing the cardinality of a set with the property. For example, some properties are known under different names. Efficient Total Dominating sets are also called Efficient Open Dominating sets, and here even the existence of such a set in a digraph G is NP-complete, as it corresponds to deciding if $V(G)$ can be partitioned by the open out-neighborhoods of some $S \subseteq V(G)$. If rows A and B have their in-restrictions and out-restrictions swapped for both σ and ρ (i.e., σ^+ of row A equals σ^- of row B and vice versa, and the same for ρ^+ and ρ^-), then a row-A set in G is always a row-B set in the digraph with all arcs of G reversed; this is the case for Dominating set versus in-Dominating set and for Kernel versus Independent Dominating set.

intersection digraphs in question. For H-digraphs, we identify reflexivity as the additional restriction that gives bounded bi-mim-width, and therefore algorithmic applications, while we prove that the bi-mim-width is unbounded when we drop this requirement. Recently, Francis, Hell, and Jacob [22] obtained polynomial-time algorithms for Kernel, Dominating Set, and Absorbing Set on reflexive interval digraphs. Our results are more general in two ways: we give algorithms for more problems, including the aforementioned ones (see Tables 1 and 2), and on much broader digraph classes (see Figure 1). Naturally, the specific algorithms presented in [22] are more efficient than the algorithm following from our general framework. In the following, we discuss the above items in more detail.

1.1 | Bi-mim-width

We introduce a new digraph width parameter, called *bi-mim-width*, which is a directed analog of the mim-width of an undirected graph introduced by Vatshelle [49]. Roughly speaking, the bi-mim-width of a digraph G is defined as a branch-width with a cut function that measures for a vertex partition (A, B) of G, the sum of the sizes of maximum induced matchings in two bipartite digraphs, one induced by edges from A to B, and the other induced by edges from B to A. This is similar to how rank-width is generalized to bi-rank-width for digraphs [31, 32]. We formally define bi-mim-width and linear bi-mim-width in Section 3. We compare bi-mim-width and other known width parameters. The mim-width of an undirected graph is exactly half of the bi-mim-width of the digraph obtained by replacing each edge with bi-directed edges, and

TABLE 2 Examples of DLCVP problems that are represented by finite or cofinite sets.

Problem name	q	DLCVP $(q \times q)$-matrix D
Directed H-Homomorphism [26]	$\lvert V(H)\rvert$	$\forall i, j D[i,j] = \left(\mu_{ij}^{+}, \mu_{ij}^{-}\right)$, where $\mu_{ij}^{+} = \begin{cases} \mathbb{N} & \text{if } ij \in E(H), \\ \{0\} & \text{otherwise} \end{cases}$ and $\mu_{ij}^{-} = \begin{cases} \mathbb{N} & \text{if } ji \in E(H), \\ \{0\} & \text{otherwise} \end{cases}$
Oriented k-Coloring [18, 47]	k	$\bigvee_{H:\overrightarrow{K_k}}$ Directed H-Homomorphism
Simple k-Coloring (*) [45]	k	$\bigvee_{H:\overrightarrow{K_k^{\circ}}}$ Directed H-Homomorphism
$\exists$ $(\sigma^+, \sigma^-, \rho^+, \rho^-)$-set (this paper)	2	$\begin{pmatrix} (\sigma^+, \sigma^-) & (\mathbb{N}, \mathbb{N}) \\ (\rho^+, \rho^-) & (\mathbb{N}, \mathbb{N}) \end{pmatrix}$
$(\delta^+ \geq k_1, \delta^- \geq k_2)$-Partition [4]	2	$\begin{pmatrix} (\{j : j \geq k_1\}, \mathbb{N}) & (\mathbb{N}, \mathbb{N}) \\ (\mathbb{N}, \mathbb{N}) & (\mathbb{N}, \{j : j \geq k_2\}) \end{pmatrix}$
$(\delta^+ \geq k_1, \delta^+ \geq k_2)$-Partition [3]	2	$\begin{pmatrix} (\{j : j \geq k_1\}, \mathbb{N}) & (\mathbb{N}, \mathbb{N}) \\ (\mathbb{N}, \mathbb{N}) & (\{j : j \geq k_2\}, \mathbb{N}) \end{pmatrix}$
$(\Delta^+ \leq k_1, \Delta^+ \leq k_2)$-Partition [6]	2	$\begin{pmatrix} (\{j : j \leq k_1\}, \mathbb{N}) & (\mathbb{N}, \mathbb{N}) \\ (\mathbb{N}, \mathbb{N}) & (\{j : j \leq k_2\}, \mathbb{N}) \end{pmatrix}$
$(\delta^+ \geq k_1, \delta^- \geq k_2)$-Bipartite-Partition [7]	2	$\begin{pmatrix} (\mathbb{N}, \mathbb{N}) & (\{j : j \geq k_1\}, \mathbb{N}) \\ (\mathbb{N}, \{j : j \geq k_2\}) & (\mathbb{N}, \mathbb{N}) \end{pmatrix}$
$(\delta^+ \geq k_1, \delta^+ \geq k_2)$-Bipartite-Partition [7]	2	$\begin{pmatrix} (\mathbb{N}, \mathbb{N}) & (\{j : j \geq k_1\}, \mathbb{N}) \\ (\{j : j \geq k_2\}, \mathbb{N}) & (\mathbb{N}, \mathbb{N}) \end{pmatrix}$
2-Out-Coloring [1]	2	$\begin{pmatrix} (\mathbb{N} \setminus \{0\}, \mathbb{N}) & (\mathbb{N} \setminus \{0\}, \mathbb{N}) \\ (\mathbb{N} \setminus \{0\}, \mathbb{N}) & (\mathbb{N} \setminus \{0\}, \mathbb{N}) \end{pmatrix}$

Note: For every row there are choices of values for which the problems are NP-complete. For Directed H-Homomorphism let $V(H) = \{1, \ldots, \lvert V(H)\rvert\}$ and denote by $H : \overrightarrow{K_k}$ that H is an orientation of a complete graph on k vertices, and by $H : \overrightarrow{K_k^{\circ}}$ that H is an orientation of a complete graph on k vertices, with loops. (*) For Simple k-Coloring, we require two nonempty color classes to avoid trivial solutions. The general algorithm can easily be modified to take this into account.

Abbreviation: DLCVP, directed locally checkable vertex partitioning.

this observation can be used to argue that a bound on the bi-mim-width of a class of digraphs implies a bound on the mim-width of a certain class of undirected graphs.

1.2 | DLCV problems

We introduce directed locally checkable vertex subset (DLCVS) and partitioning (DLCVP) problems, in analogy with [48]. We abbreviate the union of these two families of problems to "DLCV problems." A DLCVS problem is represented as a $(\sigma^+, \sigma^-, \rho^+, \rho^-)$-problem for some $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$, and it asks to find a maximum or minimum vertex set S in a digraph G such that for every vertex v in S, the numbers of out/in-neighbors in S are contained in σ^+ and σ^-, respectively, and for every vertex v in $V(G) \setminus S$, the numbers of out/in-neighbors in S are contained in ρ^+ and ρ^-, respectively. If each $\mu \in \{\sigma^+, \sigma^-, \rho^+, \rho^-\}$ is either finite or cofinite (i.e., $\mathbb{N} \setminus \mu$ is finite), then we say that the problem is *represented by finite or cofinite sets*. See Table 1

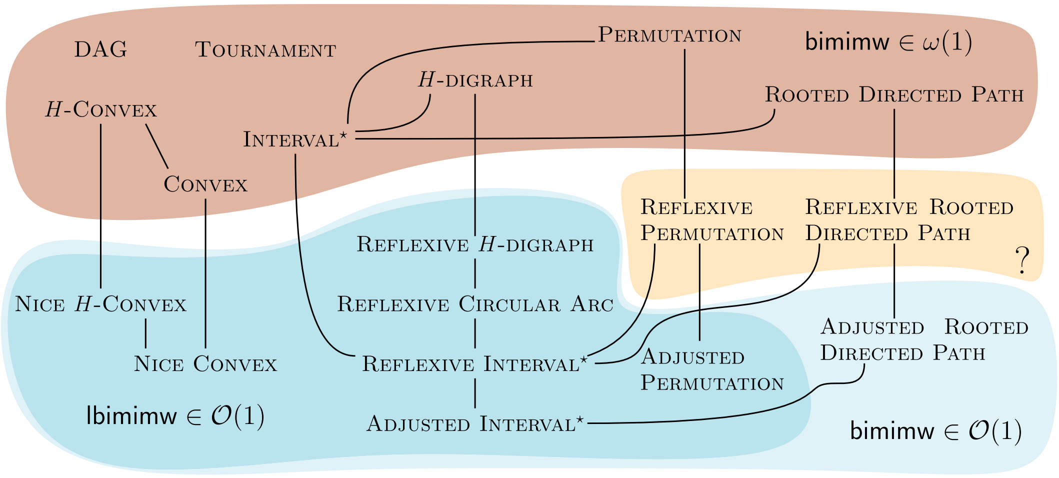

FIGURE 1 Digraph classes with bounds on their (linear) bi-mim-width. For graph classes marked with ⋆ there are polynomial-time algorithms to compute representations of their members. If digraph class A is depicted above B and there is an edge between A and B then $B \subseteq A$. The bubble on the top, marked with "bimimw $\in \omega(1)$," contains classes of unbounded bi-mim-width, while the bubble on the bottom, labeled "bimimw $\in \mathcal{O}(1)$," contains classes of constant bi-mim-width. The inner lower bubble, labeled "lbimimw $\in \mathcal{O}(1)$" contains classes whose *linear* bi-mim-width is constant. The boundedness of the bi-mim-width in the bubble labeled "?" remains an open question. DAG, directed acyclic graph. [Color figure can be viewed at wileyonlinelibrary.com]

for several examples of DLCVS problems that appear in the literature and note that they are all represented by finite or cofinite sets. In particular, it includes the Kernel problem, which was introduced by von Neumann and Morgenstern [50].

A DLCVP problem is represented by a $(q \times q)$-matrix D for some positive integer q, where for all $i, j \in \{1, \ldots, q\}$, $D[i, j] = (\mu^+_{i,j}, \mu^-_{i,j})$ for some $\mu^+_{i,j}, \mu^-_{i,j} \subseteq \mathbb{N}$. The problem asks to find a vertex partition of a given digraph into $X_1, X_2, \ldots, X_q$ such that for all $i, j \in [q]$, the numbers of out/in-neighbors of a vertex of X_i in X_j are contained in $\mu^+_{i,j}$ and $\mu^-_{i,j}$, respectively. In analogy with subset problems, we say that the problem is *represented by finite or cofinite sets* if each set appearing in a pair that is an entry of D is either finite or cofinite. Directed H-Homomorphism is a DLCVP problem represented by finite or cofinite sets: For a digraph H on vertices $\{1, \ldots, q\}$, we can view a homomorphism from a digraph G to H as a q-partition $(X_1, \ldots, X_q)$ of $V(G)$ such that we can only have an edge from X_i to X_j if the edge (i, j) is present in H. See Table 2. The Oriented k-Coloring problem, introduced by Sopena [46], asks whether there is a homomorphism to some orientation of a complete graph on at most k vertices, and can therefore be reduced to a series of DLCVP problems. Removing the requirement that the color classes have to be independent sets, Smolíková [45] introduced the notion of a *simple k-coloring*, requiring however that the number of colors is at least two, to avoid trivial solutions. Several works in the literature concern problems of 2-partitioning the vertex sets of digraphs into parts with degree constraints either inside or between the parts of the partition [1, 3, 4, 6, 7]. All of these problems can be observed to be LCVP problems as well, see Table 2. Note that in the DLCVP-framework, we can consider q-partitions for any fixed $q \geq 2$, for all problems apart from 2-Out-Coloring. This fails for q-Out-Coloring, since this problem asks for a q-coloring with no monochromatic out-neighborhood.

Theorem 1. *Directed LCVS and LCVP problems represented by finite or cofinite sets can be solved in* XP *time parameterized by the bi-mim-width of a given decomposition of the input digraph.*

Furthermore, we show that the distance variants of DLCVS problems, for instance, Distance-r Dominating Set can be solved in polynomial time on digraphs of bounded bi-mim-width. Another natural variant is the k-Kernel problem (see [5, Section 8.6.2]), which asks for a kernel in the $(k-1)$th power of a given digraph. To show this, we prove that the rth power of a digraph of bi-mim-width w has bi-mim-width at most rw (Lemma 6). For undirected graphs, there is a bound that does not depend on r [28], but we were not able to obtain such a bound for the directed case.

Theorem 2. *For any fixed r, each distance-r version of a DLCVS problem that is represented by finite or cofinite sets can be solved in* XP *time parameterized by the bi-mim-width of a given decomposition of the input digraph.*

1.3 | Classes of intersection digraphs and their bi-mim-width

We provide various classes of digraphs of bounded bi-mim-width. We first summarize our results in the following theorem and give the background below. We illustrate the bounds in Figure 1.

Theorem 3.

(i) *Given a reflexive interval digraph, one can output a linear branch decomposition of bi-mim-width at most* 2 *in polynomial time. On the other hand, interval digraphs have unbounded bi-mim-width.*
(ii) *Given a representation of an adjusted permutation digraph G, one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* 4*. Permutation digraphs have unbounded bi-mim-width.*
(iii) *Given a representation of an adjusted rooted directed path digraph G, one can construct in polynomial time a branch decomposition of G of bi-mim-width at most* 2*. Rooted directed path digraphs have unbounded bi-mim-width and adjusted rooted directed path digraphs have unbounded linear bi-mim-width.*
(iv) *Let H be an undirected graph. Given a representation of a reflexive H-digraph G, one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* $12|E(H)|$*. P_2-digraphs, which are interval digraphs, have unbounded bi-mim-width.*
(v) *Let H be an undirected graph. Given a nice H-convex digraph G with its bipartition (A, B), one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* $12|E(H)|$*. P_2-convex digraphs have unbounded bi-mim-width.*
(vi) *Tournaments and directed acyclic graphs have unbounded bi-mim-width.*

(i) *Interval digraphs*: Recall that Müller [35] devised a recognition algorithm for interval digraphs, which also outputs a representation. By testing the reflexivity of a digraph, we can recognize reflexive interval digraphs, and output its representation. We convert it into a linear branch decomposition of bi-mim-width at most 2. On the other hand, interval digraphs generally have unbounded bi-mim-width, witnessed by orientations of grids with an interval digraph representation. By Theorem 1, we can solve all DLCV problems on reflexive interval digraphs in polynomial time. This extends the polynomial-time algorithms for Independent Dominating Set and Kernel on interval nest digraphs given by Prisner [39], and includes polynomial-time algorithms for Absorbing Set, Dominating Set, and Kernel by Francis, Hell, and Jacob [22].

(ii) *Permutation digraphs*: A *permutation digraph* is an intersection digraph of pairs of line segments whose endpoints lie on two parallel lines. Müller [35] considered permutation digraphs under the name "matching diagram digraph," and observed that every interval digraph is a permutation digraph. Therefore, permutation digraphs have unbounded bi-mim-width. We say that a permutation digraph is *adjusted* if there exists one of the parallel lines, say Λ, such that for all $v \in V(G)$, S_v and T_v have the same endpoint in Λ. We show that every adjusted permutation digraph has linear bi-mim-width at most 4.

(iii) *Rooted directed path digraphs*: It is known that chordal graphs have unbounded mim-width [30, 34]. As restrictions of chordal graphs, it has been shown that rooted directed path graphs, and more generally, leaf power graphs have mim-width at most 1 [28], while they have unbounded linear mim-width. A *rooted directed path digraph* is an intersection digraph of pairs of directed paths in a rooted directed tree (every node is reachable from the root), and it is *adjusted* if for every vertex v, the head of S_v is equal to the head of T_v. We show that every adjusted rooted directed path digraph has bi-mim-width at most 2. Since this class includes the biorientations of trees, it has unbounded linear bi-mim-width.

(iv) H-*digraphs*: For an undirected graph H, an H-*graph* is an undirected intersection graph of connected subgraphs in an H-subdivision, introduced by Bíró, Hujter, and Tuza [11]. For example, interval graphs and circular-arc graphs are P_2-graphs and C_3-graphs, respectively. Fomin, Golovach, and Raymond [21] showed that H-graphs have linear mim-width at most $2|E(H)| + 1$. Motivated by H-graphs, we introduce an H-*digraph* that is the intersection digraph of pairs of connected subgraphs in an H-subdivision (where H and its subdivision are undirected). We prove that reflexive H-digraphs have linear bi-mim-width at most $12|E(H)|$. This extends the linear bound of Fomin et al. [21] for H-graphs.

(v) H-*convex digraphs*: For an undirected graph H, a bipartite digraph G with bipartition (A, B) is an H-*convex digraph*, if there exists a subdivision F of H with $V(F) = A$ such that for every vertex b of B, each of the set of out-neighbors and the set of in-neighbors of v induces a connected subgraph in F. As an analogous concept to reflexivity in H-digraphs, we say that an H-convex digraph is *nice* if for every vertex b of B, there is a bi-directed edge between b and some vertex of A. Note that H-convex graphs, introduced by Bonomo-Braberman et al. [12], can be seen as nice H-convex digraphs, by replacing every edge with bi-directed edges. We prove that nice H-convex digraphs have linear bi-mim-width at most $12|E(H)|$. This implies that H-convex graphs have linear mim-width at most $6|E(H)|$. For the special case when T is a tree with maximum degree Δ and t branching nodes, Bonomo-Braberman et al. [12] showed an improved bound of $2 + t(\Delta - 2)$ on the mim-width of T-convex graphs.

(vi) *Directed acyclic graphs and tournaments*: We show that if H is the underlying undirected graph of a digraph G, then the bi-mim-width of G is at least the mim-width of H. Using this, we can show that acyclic orientations of grids have unbounded bi-mim-width. We also prove that tournaments have unbounded bi-mim-width. This refines an argument that they have unbounded bi-rank-width [5, Lemma 9.9.11].

We can summarize our algorithmic results as follows.

Corollary 1. *Given a reflexive interval digraph, or a representation of either an adjusted permutation digraph, or an adjusted rooted directed path digraph, or a reflexive H-digraph, or a nice H-convex digraph, we can solve all DLCV problems represented by finite or cofinite sets, and their distance variants, in polynomial time.*

1.4 | Related work

Intersection digraphs have first been considered by Beineke and Zamfirescu in 1982 [9]. Sen et al. [44] introduced the class of interval digraphs and Sen et al. [43] the class of circular-arc digraphs. Permutation digraphs were first studied under the name "matching diagram digraphs" by Müller [35]. Prisner [39] showed that the problems Clique, Chromatic Number, Independent Set, Partition into Cliques, Kernel, and Independent Dominating Set are polynomial-time solvable on interval *nest* digraphs, a subclass of interval digraphs G having a representation $\{(S_v, T_v) : v \in V(G)\}$ where for each vertex $v \in V(G)$, either $S_v \subseteq T_v$ or $T_v \subseteq S_v$. Very recently, and independently of this work, Francis, Hell, and Jacob [22] showed that Absorbing Set, Dominating Set, and Kernel are polynomial-time solvable on *reflexive* interval digraphs, a superclass of interval nest digraphs. They also showed that these problems remain hard on interval digraphs, even when all intervals are single points. Feder et al. [20] considered the List H-Homomorphism problem, but posing a structural restriction on H rather than the input graph. They showed that if H is an *adjusted* interval digraph, that is, an interval digraph with a representation where both intervals associated with each vertex have the same left endpoint, then List H-Homomorphism is polynomial-time solvable.

The algorithmic result for undirected graphs analogous to ours is that all (undirected) locally checkable vertex problems are polynomial-time solvable if the input graph is given together with a decomposition of constant mim-width. This has been shown by Bui-Xuan, Telle, and Vatshelle [14]. In their work, the run time of the algorithms is stated in terms of the number of equivalence classes of the *d-neighborhood equivalence* relation, and the connection between this notion and mim-width was made explicit by Belmonte and Vatshelle [10].

1.5 | Organization of the paper

The paper is organized as follows. In Section 2, we introduce basic notations. In Section 3, we formally introduce bi-mim-width and compare it with other known width parameters. In Section 4, we prove Theorem 3, and in Section 5, we prove Theorems 1 and 2.

2 | PRELIMINARIES

For a positive integer n, we use the shorthand $[n] := \{1, \ldots, n\}$.

2.1 | Undirected graphs

We use standard notions of graph theory and refer to [19] for an overview. All undirected graphs considered in this work are finite and simple. For an undirected graph G, we denote by

$V(G)$ the vertex set of G and $E(G)$ the edge set of G. For an edge $\{u, v\} \in E(G)$, we may use the shorthand "uv."

For an undirected graph G and two disjoint vertex sets $A, B \subseteq V(G)$, we denote by $G[A, B]$ the bipartite graph on bipartition (A, B) such that $E(G[A, B])$ is exactly the set of edges of G incident with both A and B.

Let G be a graph and $e = uv \in E(G)$. The *(edge) subdivision* of e is the operation of removing the edge e and adding a new vertex x and the edges ux and xv to G. A graph H is a *subdivision* of G if H can be obtained from G by a series of edge subdivisions. If H is a subdivision of G, then each vertex in $V(G)$ is called a *branching vertex* in H. A path P in H is called a *branching path* if its endpoints are branching vertices and no other vertices in P are branching vertices.

2.2 | Digraphs

All digraphs considered in this work are finite and have no multiple edges, but may have loops. For a digraph G, we denote by $V(G)$ its vertex set and by $E(G) \subseteq V(G) \times V(G)$ its edge set. We say that an edge $(u, v) \in E(G)$ is directed from u to v. For a vertex $v \in V(G)$, the set of its *out-neighbors* is $N_G^+(v) = \{w | (v, w) \in E(G)\}$, and the set of its *in-neighbors* is $N_G^-(v) = \{u | (u, v) \in E(G)\}$. We may drop G as a subscript if it is clear from the context.

For a digraph G, the undirected graph obtained by replacing every edge with an undirected edge and then removing multiple edges is called its *underlying undirected graph*. For an undirected graph G, a digraph obtained by replacing every edge uv with one of (u, v) and (v, u) is called an *orientation*, and the digraph obtained by replacing every edge uv with two directed edges (u, v) and (v, u) is called its *biorientation*.

For a digraph G and two disjoint vertex sets $A, B \subseteq V(G)$, we denote by $G[A \to B]$ the bipartite digraph on bipartition (A, B) with edge set $E(G[A \to B]) = E(G) \cap (A \times B)$, and denote by $G[A, B]$ the bipartite digraph on bipartition (A, B) with edge set $E(G[A \to B]) \cup E(G[B \to A])$.

A *tournament* is an orientation of a complete graph.

2.3 | Common notation

Let G be an undirected graph or a digraph. A set M of edges in G is a *matching* if no two edges share an endpoint, and it is an *induced matching* if there are no edges in G meeting two distinct edges in M. We denote by $\nu(G)$ the maximum size of an induced matching of G.

For two undirected graphs or two directed graphs G and H, we denote by $G \cap H = (V(G) \cap V(H), E(G) \cap E(H))$ and $G \cup H = (V(G) \cup V(H), E(G) \cup E(H))$.

For a vertex set A of G, we denote by $\overline{A} := V(G) \backslash A$. A vertex bipartition $(A, \overline{A})$ of G for some vertex set A of G will be called a *cut*. A cut $(A, \overline{A})$ of G is *balanced* if $|V(G)|/3 < |A| \leq 2|V(G)|/3$.

For two vertices $u, v \in V(G)$, the *distance* between u and v, denoted by $\text{dist}_G(u, v)$ or simply $\text{dist}(u, v)$, is the length of the shortest path from u to v (if G is a digraph, then we consider directed paths). For $u \in V(G)$ and $A \subseteq V(G)$, we let $\text{dist}_G(u, A) = \min_{v \in A} \text{dist}_G(u, v)$. For a positive integer r, we denote by G^r the graph obtained from G by, for every pair x, y of vertices in G with $\text{dist}_G(x, y) \leq r$, adding an edge from x to y. We call it the *r-th power* of G.

For an equivalence relation ~, we denote by nec(~) the number of equivalence classes of ~.

3 | BI-MIM-WIDTH

In this section, we introduce the bi-mim-width of a digraph. For an undirected graph G and $A \subseteq V(G)$, let $\text{mim}_G(A) := \nu(G[A, \overline{A}])$. For a digraph G and $A \subseteq V(G)$,

- let $\text{mim}_G^+(A) := \nu(G[A \to \overline{A}])$ and $\text{mim}_G^-(A) := \nu(G[\overline{A} \to A])$ and
- let $\text{bimim}_G(A) := \text{mim}_G^+(A) + \text{mim}_G^-(A)$.

A tree is *subcubic* if it has at least two vertices and every internal vertex has degree 3. A tree T is a *caterpillar* if it contains a path P such that every vertex in $V(T)\setminus V(P)$ has a neighbor in P. Let G be an undirected graph or a digraph. A *branch decomposition over the vertex set* of G is a pair (T, λ) of a subcubic tree T and a bijection λ from $V(G)$ to the leaves of T. We would like to point out that branch decompositions were originally defined over edge sets of graphs [41]. Throughout this work, we will only use branch decompositions over vertex sets which from now on we simply refer to as *branch decompositions*. If T is a caterpillar, then (T, λ) is called a *linear branch decomposition* of G. Note that in Figure 3 on page 11, we give an illustration of the following definition.

Definition 1 (Bi-mim-width)**.** Let G be a digraph and let (T, λ) be a branch decomposition of G. For each edge $e \in E(T)$, let T_A and T_B be the connected components of $T - e$. Let (A_e, B_e) be the cut of G where A_e is the set of vertices that λ maps to the leaves in T_A and B_e is the set of vertices that λ maps to the leaves in T_B. The *bi-mim-width* of (T, λ) is $\text{bimimw}(T, \lambda) := \max_{e \in E(T)}(\text{bimim}_G(A_e))$. The *bi-mim-width* of G, denoted by $\text{bimimw}(G)$, is the minimum bi-mim-width of any branch decomposition of G. The *linear bi-mim-width* of G, denoted by $\text{lbimimw}(G)$, is the minimum bi-mim-width of any linear branch decomposition of G.

The definition of bi-mim-width is motivated by the mim-width of an undirected graph introduced by Vatshelle [49].

Definition 2 (Mim-width)**.** Let G be an undirected graph and let (T, λ) be a branch decomposition of G. For each edge $e \in E(T)$, let T_A and T_B be the two connected components of $T - e$. Let (A_e, B_e) be the cut of G where A_e is the set of vertices that λ maps to the leaves in T_A and B_e is the set of vertices that λ maps to the leaves in T_B. The *mim-width* of (T, λ) is $\text{mimw}(T, \lambda) := \max_{e \in E(T)} \text{mim}_G(A_e)$. The *mim-width* of G, denoted by $\text{mimw}(G)$, is the minimum mim-width of any branch decomposition of G. The *linear mim-width* of G, denoted by $\text{lmimw}(G)$, is the minimum mim-width of any linear branch decomposition of G.

Remark 1. Note that there is a natural correspondence between linear branch decompositions of a (di)graph G and linear orders of $V(G)$. Given a linear branch decomposition (T, λ) of G, consider a maximal path P in T such that each internal node of P has exactly one neighbor outside of P. We can associate a linear order of $V(G)$ with

P as follows. We start at one endpoint t of P, which is a leaf of T, and let the vertex mapped to t be the first vertex. Then, we follow the nodes P in the order in which they appear on the path, at each node letting the vertex of G mapped to its single pendant leaf be the next vertex in the order. Consider the set of cuts induced by this order as follows. For each position i, take the vertices on one side to be the one whose index is at most i, and the remaining ones on the other side. All cuts induced by (T, λ) are of this form, apart from the ones that have a single vertex on one side. However, for mim-width and bi-mim-width, such cuts have low complexity, in particular, their mim-value can be at most one and their bi-mim-value can be at most two. (Constructing a linear branch decomposition from a linear order of $V(G)$ can be done analogously.) Therefore, when dealing with linear mim-width of at least one or bi-mim-width of at least two, we can safely switch to the language of linear orders which we do throughout the text.

The following two lemmas are clear by definition.

Lemma 1. *Let G be a digraph and let H be an induced subdigraph of G. Then* $\mathrm{bimimw}(H) \leq \mathrm{bimimw}(G)$ *and* $\mathrm{lbimimw}(H) \leq \mathrm{lbimimw}(G)$.

Lemma 2. *Let G be an undirected graph and let H be the biorientation of G. Then for every vertex partition (A, B) of G, we have* $\nu(G[A, B]) = \frac{\nu(H[A \to B]) + \nu(H[B \to A])}{2}$. *In particular, we have* $\mathrm{mimw}(G) = \frac{\mathrm{bimimw}(H)}{2}$.

We show that if a digraph G has small bi-mim-width, then its underlying undirected graph has small mim-width. But the other direction does not hold; the class of tournaments has unbounded bi-mim-width.

Lemma 3. *Let G be a digraph and let H be the underlying undirected graph of G. Then* $\mathrm{mimw}(H) \leq \mathrm{bimimw}(G)$ *and* $\mathrm{lmimw}(H) \leq \mathrm{lbimimw}(G)$. *On the other hand, the class of tournaments has unbounded bi-mim-width, while their underlying undirected graphs have linear mim-width* 1.

Proof. Let (T, λ) be a branch decomposition of H, and assume that it has mim-width t. Then, T has an edge e inducing a cut (A_e, B_e) such that $\mathrm{mim}_H(A_e) = t$. Let M be a maximum induced matching of $H[A_e, B_e]$, and let (M_1, M_2) be the partition of M such that original edges in M_1 are contained in $G[A_e \to B_e]$ and original edges of M_2 are contained in $G[B_e \to A_e]$. It shows that the sum of the sizes of the maximum induced matchings in $G[A_e \to B_e]$ and in $G[B_e \to A_e]$ is at least t. This implies that (T, λ) has bi-mim-width at least t, as a branch decomposition of G. As we chose (T, λ) arbitrarily, we conclude that G has bi-mim-width at least $\mathrm{mimw}(H)$. The same argument obtained by replacing a branch decomposition with a linear branch decomposition shows that $\mathrm{lmimw}(H) \leq \mathrm{lbimimw}(G)$.

We prove the second statement. For every integer $n \geq 2$, we define G_n as the digraph on vertex set $\{v_{i,j} : i, j \in [n]\}$ satisfying the following:

- for all $i \in [n]$ and $j_1, j_2 \in [n]$ with $j_1 < j_2$, there is an edge from v_{i,j_1} to v_{i,j_2},
- for all $i \in [n-1]$ and $j \in [n]$, there is an edge from $v_{i,j}$ to $v_{i+1,j}$,

- for all $i_1, i_2, j_1, j_2 \in [n]$ with $i_1 < i_2$,
  - if $i_2 - i_1 \equiv 0 \pmod 3$ and $j_1 \geq j_2$, then there is an edge from v_{i_1,j_1} to v_{i_2,j_2},
  - if $i_2 - i_1 \equiv 0 \pmod 3$ and $j_1 < j_2$, then there is an edge from v_{i_2,j_2} to v_{i_1,j_1},
  - if $i_2 - i_1 \equiv 1 \pmod 3$ and $(i_2, j_2) \neq (i_1 + 1, j_1)$, then there is an edge from v_{i_2,j_2} to v_{i_1,j_1},
  - if $i_2 - i_1 \equiv 2 \pmod 3$, then there is an edge from v_{i_1,j_1} to v_{i_2,j_2}.

This construction adds, for each $(i_1, j_1), (i_2, j_2) \in [n] \times [n]$ with $(i_1, j_1) \neq (i_2, j_2)$ precisely one of the edges $(v_{i_1,j_1}, v_{i_2,j_2}), (v_{i_2,j_2}, v_{i_1,j_1})$, therefore G_n is a tournament. For each $i \in [n]$, we let $R_i := \{v_{i,x} : x \in [n]\}$ and $C_j := \{v_{y,j} : y \in [n]\}$.

We claim that for every positive integer k, G_{18k} has bi-mim-width at least k. Suppose for contradiction that there is a branch decomposition of G_{18k} of bi-mim-width at most $k - 1$. Let $n := 18k$ and $G := G_n$.

Observe that there is a cut (A, B) of G where

$$\frac{|V(G)|}{3} \leq |A| < \frac{2|V(G)|}{3} \quad \text{and} \quad \text{bimim}_G(A) \leq k - 1.$$

To obtain such a cut, we consider the branch decomposition of G as rooted, and let t be the node that is farthest from the root such that the number q of vertices that are mapped to the descendants of t is at least $|V(G)|/3$. Because t is chosen to be farthest from the root, q is also less than $2|V(G)|/3$. We can set A to be the set of vertices of G that are mapped to the descendants of t.

We divide into two cases.

Case 1. For every $j \in [n]$, C_j contains a vertex of A and a vertex of B: Then there exists $a_j \in [n - 1]$ for each $j \in [n]$ such that $v_{a_j,j}$ and $v_{a_j+1,j}$ are contained in distinct sets of A and B. Let $J_1 \subseteq [n]$ be a set of size at least $n/2$ such that either

- for all $j \in J_1$, $v_{a_j,j} \in A$, or
- for all $j \in J_1$, $v_{a_j,j} \in B$.

Without loss of generality, we assume that for all $j \in J_1$, $v_{a_j,j} \in A$. The proof is symmetric when $v_{a_j,j} \in B$. Furthermore, we take a subset J_2 of J_1 of size at least $n/6$ such that integers in $\{a_j : j \in J_2\}$ are pairwise congruent modulo 3.

Now, we verify that $\{(v_{a_j,j}, v_{a_j+1,j}) : j \in J_2\}$ is an induced matching in $G[A \to B]$. Let $x, y \in J_2$ be distinct integers and assume that $a_x \leq a_y$. If $a_x = a_y$, then $(a_x + 1) - a_y \equiv 1 \pmod 3$, and thus there is an edge from $v_{a_x+1,x}$ to $v_{a_y,y}$ and similarly, there is an edge from $v_{a_y+1,y}$ to $v_{a_x,x}$. Assume that $a_x < a_y$. Then $a_y - (a_x + 1) \equiv 2 \pmod 3$ and $(a_y + 1) - a_x \equiv 1 \pmod 3$, and thus there is an edge from $v_{a_x+1,x}$ to $v_{a_y,y}$ and there is an edge from $v_{a_y+1,y}$ to $v_{a_x,x}$. This shows that there are no edges between $\{v_{a_x,x}, v_{a_x+1,x}\}$ and $\{v_{a_y,y}, v_{a_y+1,y}\}$ in $G[A \to B]$, and therefore $\{(v_{a_j,j}, v_{a_j+1,j}) : j \in I_2\}$ is an induced matching in $G[A \to B]$ of size at least $n/6 = 3k$. This contradicts the assumption that $\text{bimim}_G(A) \leq k - 1$.

Case 2. For some $j \in [n]$, C_j is fully contained in one of A and B: Without loss of generality, we assume that C_j is contained in A. Since $|B| > n/3$, there is a subset $I_1 \subseteq [n]$

such that $|I_1| \geq n/3$, and for each $i \in I_1$, $a_{i,b_i} \in B$ for some $b_i \in [n]$. We take a subset I_2 of I_1 of size at least $n/9$ where all integers in I_2 are pairwise congruent modulo 3. Lastly, we take a subset I_3 of I_2 of size at least $n/18$ such that either

- for all $i \in I_3$, $b_i > j$, or
- for all $i \in I_3$, $b_i < j$.

First, we assume that $b_i > j$ for all $i \in I_3$. We verify that $\{(v_{i,j}, v_{i,b_i}) : i \in I_3\}$ is an induced matching in $G[A \to B]$. Let $x, y \in I_3$ with $x < y$. As x and y are congruent modulo 3, there is an edge from v_{x,b_x} to $v_{y,j}$ and there is an edge from v_{y,b_y} to $v_{x,j}$. It shows that there are no edges between $\{v_{x,j}, v_{x,b_x}\}$ and $\{v_{y,j}, v_{y,b_y}\}$ in $G[A \to B]$, and therefore $\{(v_{x,j}, v_{x,b_x}) : x \in I_3\}$ is an induced matching in $G[A \to B]$ of size at least $n/18 = k$. It contradicts the assumption that $\text{bimim}(A) \leq k - 1$. The argument when $b_i < j$ for all $i \in I_3$ is similar. □

We argue that directed tree-width [29] and bi-mim-width are incomparable. The (rather technical) definition of directed tree-width is not necessary for the remainder of this paper, so we only give it in Appendix A. The only facts we need are that acyclic digraphs have directed tree-width 1, and that subdivisions of cylindrical grids (see Figure 2) have unbounded directed tree-width [29, 33].

Lemma 4. *Directed tree-width and bi-mim-width are incomparable.*

Proof. The class of all acyclic orientations of undirected grids has directed tree-width 1 but has unbounded bi-mim-width, by Lemma 3.

We construct a class of digraphs of bounded bi-mim-width, but unbounded directed tree-width. For positive integers n and k, let $C_{n,k}$ be the digraph obtained from the disjoint union of n independent sets $A_1, A_2, \ldots, A_n$ of size k with $A_{n+1} := A_1$ such that

- $E(C_{n,k}) = \{(u, v) : u \in A_i, v \in A_{i+1}, i \in [n]\}$.

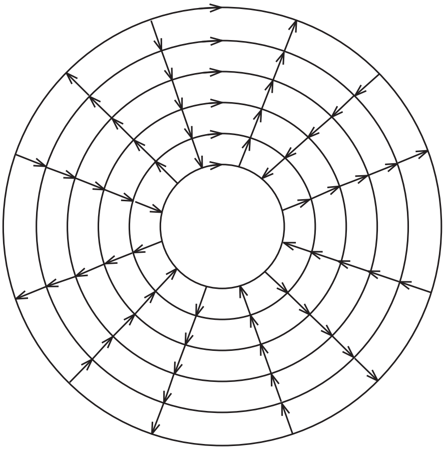

FIGURE 2 Illustration of a cylindrical grid.

We argue that any linear order of $V(C_{n,k})$ such that v appears before w whenever $v \in A_i$ and $w \in A_j$ with $i < j$ has bi-mim-width at most three. Let $(B, \overline{B})$ be any cut induced by such a linear order. There is at most one $i \in [n]$ such that $A_i \cap B \neq \emptyset$ and $A_i \cap \overline{B} \neq \emptyset$. Therefore, the edges in $C_{n,k}[B, \overline{B}]$ are made up of the sets (whenever they exist) $A_{i-1} \times \overline{B} \cap A_i, B \cap A_i \times A_{i+1}$, and $\overline{B} \cap A_n \times B \cap A_1$. Each such set contributes with at most one edge to the bi-mim-value of the cut $(B, \overline{B})$. To see that $C_{n,k}$ with sufficiently large n, k has large directed tree-width, one can create a subdivision of a cylindrical grid as a subgraph which has large directed tree-width [29, 33]. □

We compare the bi-mim-width with the bi-rank-width of a digraph, introduced by Kanté [31]. Kanté and Rao [32] later generalized this notion to edge-colored graphs. For a digraph G, and disjoint $A, B \subseteq V(G)$, we denote by $M_G[A \to B]$ the matrix whose rows are indexed by A and columns are indexed by B such that for $a \in A$ and $b \in B$, $M_G[A \to B](a, b) = 1$ if there is an edge from a to b and 0 otherwise. For a digraph G and $A \subseteq V(G)$,

- let $\text{cutrk}_G^+(A) := \text{rank}(M_G[A \to \overline{A}])$ and $\text{cutrk}_G^-(A) := \text{rank}(M_G[\overline{A} \to A])$ and
- let $\text{bicutrk}_G(A) := \text{cutrk}_G^+(A) + \text{cutrk}_G^-(A)$,

where the rank of a matrix is computed over the binary field.

Definition 3 (Bi-rank-width). Let G be a digraph and let (T, λ) be a branch decomposition of G. For each edge $e \in E(T)$, let T_A and T_B be the two connected components of $T - e$. Let (A_e, B_e) be the cut of G where A_e is the set of vertices that λ maps to the leaves in T_A and B_e is the set of vertices that λ maps to the leaves in T_B. The *bi-rank-width* of (T, λ) is $\text{birw}(T, \lambda) := \max_{e \in E(T)} \text{bicutrk}_G(A_e)$. The *bi-rank-width* of G, denoted by $\text{birw}(G)$, is the minimum bi-rank-width of any branch decomposition of G. The *linear bi-rank-width* of G, denoted by $\text{lbirw}(G)$, is the minimum bi-rank-width of any linear branch decomposition of G.

We can verify that for every digraph G, $\text{bimimw}(G) \leq \text{birw}(G)$. Interestingly, we can further show that for every positive integer r, the bi-mim-width of the r-th power of G is at most the bi-rank-width of G. This does not depend on the value of r.

Lemma 5. *Let r and w be positive integers. If (T, λ) is a branch decomposition of a digraph G of bi-rank-width w, then it is a branch decomposition of G^r of bi-mim-width at most w.*

Proof. It is sufficient to prove that for every ordered vertex partition (A, B) of G, we have $\nu(G^r[A \to B]) \leq \text{rank}(M_G[A \to B])$. Assume $\text{rank}(M_G[A \to B]) = t$ and suppose for contradiction that $\nu(G^r[A \to B]) \geq t + 1$.

Let $\{(a_i, b_i) : i \in [t+1]\}$ be an induced matching of $G^r[A \to B]$ with $\{a_i : i \in [t+1]\} \subseteq A$. It means that for each $i \in [t+1]$, there is a directed path P_i of length at most r from a_i to b_i in G such that the paths in $\{P_i : i \in [t+1]\}$ are pairwise vertex-disjoint. We choose an edge (u_i, v_i) in each P_i where $u_i \in A$ and $v_i \in B$. As $\text{rank}(M_G[\{u_i : i \in [t+1]\} \to \{v_i : i \in [t+1]\}]) \leq \text{rank}(M_G[A \to B]) = t$, the matrix $M_G[\{u_i : i \in [t+1]\} \to \{v_i : i \in [t+1]\}]$ is linearly dependent. In particular, there is a nonempty subset I of $[t+1]$ where the sum of $M_G[\{u_i\} \to \{v_j : j \in I\}]$ over $i \in I$ becomes a zero vector (as we are working on the binary field). We choose such a subset with minimum $|I|$. Without loss of generality, we may assume that $I = [q]$ for some $q \geq 2$.

Now, we claim that for each $x \in [q]$, u_x has an out-neighbor in $\{v_j : j \in [q] \backslash \{x\}\}$. Suppose that this is not true, that is, there is $x \in [q]$ where u_x has no out-neighbor in $\{v_j : j \in [q] \backslash \{x\}\}$. This means that the row of $M_G[\{u_i : i \in [q]\} \rightarrow \{v_j : j \in [q]\}]$ indexed by u_x does not affect the sum of columns indexed by $\{v_j : j \in [q] \backslash \{x\}\}$. Therefore, the sum of $M_G[\{u_i\} \rightarrow \{v_j : j \in [q] \backslash \{x\}\}]$ over $i \in [q] \backslash \{x\}$ must be a zero vector. This contradicts the minimality of I.

We deduce that there exists a sequence $(i_1, i_2, ..., i_y)$ of at least two distinct elements in $[q]$ with $i_{y+1} = i_1$ such that for each $h \in [y]$, there exists an edge from u_{i_h} to $v_{i_{h+1}}$. For each $h \in [q]$, let ℓ_{i_h} be the length of the subpath of P_{i_h} from a_{i_h} to u_{i_h}. Observe that if $\ell_{i_h} \leq \ell_{i_{h+1}}$, then there is a directed path of length at most r from a_{i_h} to $b_{i_{h+1}}$ by using the sub-path a_{i_h} to u_{i_h}, then the edge to $v_{i_{h+1}}$ and then the sub-path to $b_{i_{h+1}}$, which contradicts the assumption that there is no edge from a_{i_h} to $b_{i_{h+1}}$ in G^r. On the other hand, because of the cycle structure it is not possible that for all $h \in [x]$, $\ell_{i_h} > \ell_{i_{h+1}}$. So, we have a contradiction. □

Note that the same argument holds for undirected graphs; if an undirected graph G has rank-width w, then any power of G has mim-width at most w. This extends the two arguments in [28] that any power of an undirected graph of tree-width $w - 1$ has mim-width at most w, and any power of an undirected graph of clique-width w has mim-width at most w, because such graphs have rank-width at most w [37, 38].

Next, we show that the r-th power of a digraph of bi-mim-width w has bi-mim-width at most rw. This will be used to prove Theorem 2.

Lemma 6. *Let r and w be positive integers. If (T, λ) is branch decomposition of a digraph G of bi-mim-width w, then it is a branch decomposition of G^r that has bi-mim-width at most rw.*

Proof. It is sufficient to prove that for every ordered vertex partition (A, B) of G, we have $\nu(G^r[A \rightarrow B]) \leq r\nu(G[A \rightarrow B])$. Assume $\nu(G[A \rightarrow B]) = t$ and suppose for contradiction that $\nu(G^r[A \rightarrow B]) \geq rt + 1$.

Let $\{(a_i, b_i) : i \in [rt + 1]\}$ be an induced matching of $G^r[A \rightarrow B]$ with $\{a_i : i \in [rt + 1]\} \subseteq A$. For each $i \in [rt + 1]$, let P_i be a directed path of length at most r from a_i to b_i in G. We choose an edge (c_i, d_i) in each P_i where $c_i \in A$ and $d_i \in B$. For each $i \in [rt + 1]$, let ℓ_i be the length of the subpath of P_i from a_i to c_i. Observe that $0 \leq \ell_i \leq r - 1$.

By the pigeonhole principle, there exists a subset I of $[rt + 1]$ of size at least $t + 1$ such that for all $i_1, i_2 \in I$, $\ell_{i_1} = \ell_{i_2}$. Since $\nu(G[A \rightarrow B]) = t$, there exist distinct integers $i_1, i_2 \in I$ such that there is an edge from c_{i_1} to d_{i_2}. Then there is a path of length at most d from a_{i_1} to b_{i_2}, contradicting the assumption that there is no edge from a_{i_1} to b_{i_2} in G^r. □

4 | CLASSES OF DIGRAPHS OF BOUNDED BI-MIM-WIDTH

In this section, we present several digraph classes of bounded bi-mim-width, which are reflexive H-digraphs (Proposition 1), adjusted permutation digraphs (Proposition 4), adjusted rooted directed path digraphs (Proposition 5), and nice H-convex digraphs (Proposition 6).

We recall that a digraph G is an *intersection digraph* if there exists a family $\{(S_v, T_v) : v \in V(G)\}$ of ordered pairs of sets, called a *representation*, such that there is an edge from v to w in G if and only if S_v intersects T_w. We say that G is *reflexive* if for each $v \in V(G)$, $S_v \cap T_v \neq \emptyset$.

4.1 | *H*-digraphs and interval digraphs

We define H-digraphs, which generalize interval digraphs. For an illustration of a reflexive interval digraph, see Figure 3.

Definition 4 H(-digraph)**.** Let H be an undirected graph. A digraph G is an *H-digraph* if there is a subdivision F of H and a family $\{(S_v, T_v) : v \in V(G)\}$ of ordered pairs of connected subgraphs of F such that G is the intersection digraph with representation $\{(S_v, T_v) : v \in V(G)\}$.

Definition 5 (Interval digraph)**.** A digraph G is an interval digraph if it is a P_2-digraph.

We show that for fixed H, reflexive H-digraphs have bounded linear bi-mim-width.

Proposition 1. *Let H be an undirected graph. Given a representation of a reflexive H-digraph G, one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* $12|E(H)|$.

Proof. Let $m := |E(H)|$. We may assume that H is connected. If H has no edge, then it is trivial. Thus, we may assume that $m \geq 1$.

Let G be a reflexive H-digraph, let F be a subdivision of H, and let $\mathcal{M} := \{(S_v, T_v) : v \in V(G)\}$ be a given reflexive H-digraph representation of G with

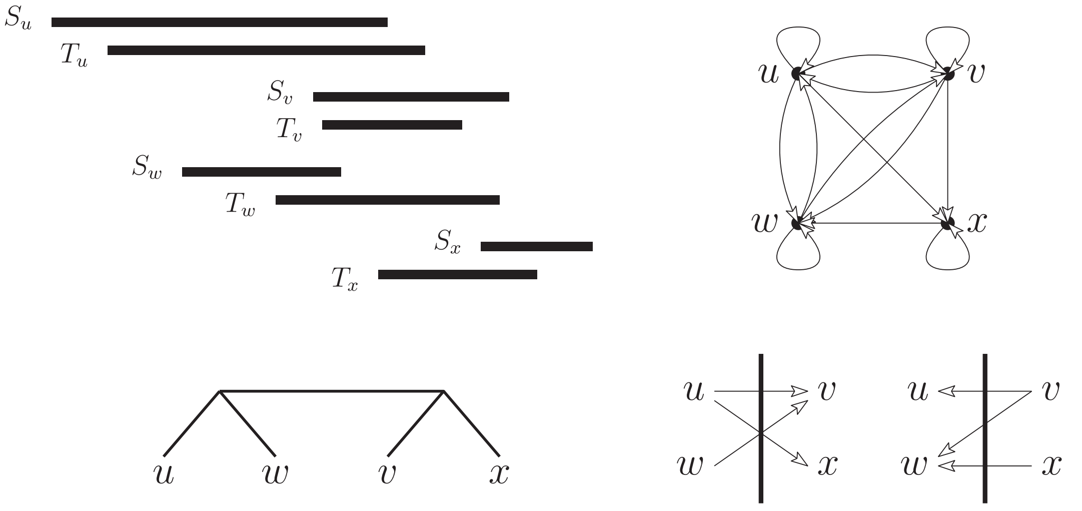

FIGURE 3 An example of a reflexive interval digraph. On the top left is its representation, on the top right one of its drawings, on the bottom left a linear branch decomposition and the bottom right shows that the cut associated with the "middle" edge of the branch decomposition has bi-mim-width value 2.

underlying graph F. For each $v \in V(G)$, choose a vertex α_v in $S_v \cap T_v$. We may assume that vertices in $\{\alpha_v : v \in V(G)\}$ are pairwise distinct and they are not branching vertices, by subdividing F more and changing $\mathcal{M}$ accordingly, if necessary.

Let r be a branching vertex of F; we obtain a BFS ordering $<_B$ of F starting from r. We denote by $v <_B w$ if v appears before w in the BFS ordering. We give a linear ordering L of G such that for all $v, w \in V(G)$, if $\alpha_v <_B \alpha_w$, then v appears before w in L. This can be done in linear time.

We claim that L has bi-mim-width at most $12m$. We choose a vertex v of G arbitrarily, and let A be the set of vertices in G that are v or a vertex appearing before v in L, and let $B := V(G)\backslash A$. It suffices to show $\text{bimim}_G(A) \leq 12m$. Let A^* be the set of vertices of F that are α_v or a vertex appearing before α_v, and let $B^* := V(F)\backslash A^*$. Let $\mathcal{P}$ be the set of paths in F such that

- for every $P \in \mathcal{P}$, P is a subpath of some branching path of F and it is a maximal path contained in one of A^* and B^*,
- $\bigcup_{P\in\mathcal{P}} V(P) = V(F)$.

We claim that each branching path of F is partitioned into at most three vertex-disjoint paths in $\mathcal{P}$. Let Q be a branching path of F. Observe that each connected component of $F[A^* \cap V(Q)]$ contains an endpoint of Q, because the BFS ordering $<_B$ starts from a branching vertex of F. Therefore, $F[A^* \cap V(Q)]$ contains at most two connected components, and it shows that Q is partitioned into at most three vertex-disjoint paths in $\mathcal{P}$. Thus, we have $|\mathcal{P}| \leq 3m$. Note that two paths in $\mathcal{P}$ from two distinct branching paths may share an endpoint.

We first show that $\text{mim}_G^+(A) \leq 6m$. Suppose for contradiction that $G[A \to B]$ contains an induced matching M of size $6m + 1$. By the pigeonhole principle, there is a subset $M_1 = \{(x_i, y_i) : i \in [3]\}$ of M of size three and a path P in $\mathcal{P}$ such that for every $(x, y) \in M_1$, S_x and T_y meet on P. Let p_1, p_2 be the endpoints of P.

Observe that $V(P) \subseteq A^*$ or $V(P) \subseteq B^*$. So, for each $i \in [3]$, it is not possible that α_{x_i} and α_{y_i} are both contained in $V(P)$. It implies that each connected component of $(S_{x_i} \cup T_{y_i}) \cap P$ contains an endpoint of P, as $S_{x_i} \cup T_{y_i}$ is connected. Therefore, there are at least two integers $j_1, j_2 \in [3]$ and a connected component C_1 of $(S_{x_{j_1}} \cup T_{y_{j_1}}) \cap P$ and a connected component C_2 of $\left(S_{x_{j_2}} \cup T_{y_{j_2}}\right) \cap P$ so that

- C_1 and C_2 contain the same endpoint of P, and
- for each $i \in [2]$, C_i contains a vertex of $S_{x_{j_i}}$ and a vertex of $T_{y_{j_i}}$.

However, it implies that (x_{j_1}, y_{j_2}) or (x_{j_2}, y_{j_1}) is an edge, a contradiction.

We deduce that $\text{mim}_G^+(A) \leq 6m$. By a symmetric argument, we get $\text{mim}_G^-(A) \leq 6m$. Therefore, we have $\text{bimim}_G(A) \leq 12m$, as required. □

We obtain a better bound for reflexive interval digraphs.

Proposition 2. *Given a reflexive interval digraph, one can output a linear branch decomposition of bi-mim-width at most* 2 *in polynomial time.*

Proof. Let G be a given reflexive interval digraph. By Müller's recognition algorithm for interval digraphs [35], one can output its representation in polynomial time.

Now, we follow the proof of Proposition 1. In this case, $\mathcal{P}$ consists of exactly two paths, one induced by A^* and the other induced by B^*. Because of it, it is not difficult to observe that $\text{mim}_G^+(A) \leq 1$ and $\text{mim}_G^-(A) \leq 1$ (similar to interval graphs). Thus, it has linear bi-mim-width at most 2. □

Proposition 3. *Interval digraphs have unbounded bi-mim-width.*

Proof. We will construct some orientation of the $(n \times n)$-grid as an interval digraph.

For $i, j \in [n]$, we construct $S_{v_{i,j}}$ and $T_{v_{i,j}}$ as follows. For every odd integer i, we set

- $S_{v_{i,j}} := [2(n+1)j + 2i - 1, 2(n+1)j + 2i + 1]$ and
- $T_{v_{i,j}} := [2(n+1)(j-1) + 2i, 2(n+1)(j-1) + 2i]$,

and for every even integer i, we set

- $S_{v_{i,j}} := [2(n+1)(j-1) + 2i, 2(n+1)(j-1) + 2i]$ and
- $T_{v_{i,j}} := [2(n+1)j + 2i - 1, 2(n+1)(j-1) + 2i + 1]$.

Let G_n be the intersection digraph on the vertex set $\{v_{i,j} : i, j \in [n]\}$ with representation $\{(S_{v_{i,j}}, T_{v_{i,j}}) : i, j \in [n]\}$. Observe that for $i, j \in [n-1]$, if i is odd, then $(v_{i,j}, v_{i+1,j})$ and $(v_{i,j}, v_{i,j+1})$ are edges, and if i is even, then $(v_{i+1,j}, v_{i,j})$ and $(v_{i,j+1}, v_{i,j})$ are edges. See Figure 4 for an illustration.

This is an orientation of the $(n \times n)$-grid. It is known that the $(n \times n)$-grid has mim-width at least $n/3$; see [49, Theorem 4.3.10]. By Lemma 3, it has bi-mim-width at least $n/3$. □

4.2 | Permutation digraphs

Permutation digraphs are directed analogs of permutation graphs. We also introduce its subclass class of *adjusted* permutation digraphs in the following definition.

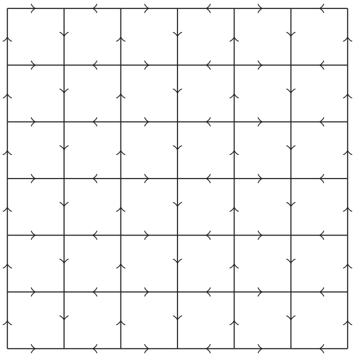

FIGURE 4 Interval digraph G_7 in Proposition 3. The left bottom vertex is $v_{1,1}$ and its right-hand vertex is $v_{2,1}$ and so on.

Definition 6 ([Adjusted] permutation digraph). A digraph G is a *permutation digraph* if there is a family $\{(S_v, T_v) : v \in V(G)\}$ of pairs of line segments whose endpoints lie on two parallel lines Λ_1 and Λ_2 where G is the intersection digraph with representation $\{(S_v, T_v) : v \in V(G)\}$. A permutation digraph is *adjusted* if it has such a representation where for all $v \in V(G)$, S_v and T_v have a common endpoint in Λ_1.

We observe that interval digraphs are permutation digraphs. When we have a representation $\{(S_v, T_v) : v \in V(G)\}$ for an interval digraph on a line $\Lambda = \{x : x \in \mathbb{R}\}$, we consider two copies $\Lambda_1 = \{(x, 0) : x \in \mathbb{R}\}$ and $\Lambda_2 = \{(x, 1) : x \in \mathbb{R}\}$ of Λ, and then for each $S_v = [a_v, b_v]$, we make S_v^* as the line segment linking $(a_v, 0)$ and $(b_v, 1)$, and for each $T_v = [c_v, d_v]$, we make T_v^* as the line segment linking $(c_v, 1)$ and $(d_v, 0)$. One can verify that $\{(S_v^*, T_v^*) : v \in V(G)\}$ is a representation of the same digraph. Because of this and by Proposition 3, permutation digraphs also have unbounded bi-mim-width.

Let us motivate how we arrived at the given definition of adjusted permutation digraphs. As we have just seen, permutation digraphs generalize interval digraphs. In [20], adjusted *interval* digraphs are defined as the interval digraphs with an intersection model where for each vertex, both intervals have the same left endpoint. An analogous idea in the context of permutation digraphs is to require that the two line segments representing a vertex have a common endpoint, and it seems natural to require that the same endpoint is always on the (upper or lower) line of the representation.

We show that adjusted permutation digraphs have linear bi-mim-width at most 4. Note that an adjusted permutation digraph is reflexive, but we were not able to show that all reflexive permutation digraphs have bounded bi-mim-width. It remains open whether there is a constant bound on (linear) bi-mim-width of reflexive permutation digraphs.

Proposition 4. *Given a representation of an adjusted permutation digraph G, one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* 4.

Proof. Let $\Lambda_1 := \{(x, 0) : x \in \mathbb{R}\}$ and $\Lambda_2 := \{(x, 1) : x \in \mathbb{R}\}$ be two lines. Let G be a given adjusted permutation digraph with its representation $\{(S_v, T_v) : v \in V(G)\}$ where S_v and T_v are line segments whose endpoints lie on Λ_1 and Λ_2 and they have a common endpoint in Λ_1, say $(\alpha_v, 0)$. For each $v \in V(G)$, let $(\beta_v, 1)$ be the endpoint of S_v in Λ_2 and $(\gamma_v, 1)$ be the endpoint of T_v in Λ_2.

We give a linear ordering L of G such that for all $v, w \in V(G)$, if $\alpha_v < \alpha_w$, then v appears before w in L. This can be done in linear time.

We claim that L has bi-mim-width at most 4. We choose a vertex v of G arbitrarily, and let A be the set of vertices in G that are v or a vertex appearing before v in L, and let $B := V(G) \backslash A$.

We verify that $\text{mim}_G^+(A) \le 2$. Suppose for contradiction that $G[A \to B]$ has an induced matching $\{(v_i, w_i) : i \in [3]\}$ with $v_1, v_2, v_3 \in A$. Without loss of generality, we assume that $\alpha_{v_1} \le \alpha_{v_2} \le \alpha_{v_3}$. Observe that $\alpha_{w_1}, \alpha_{w_2} > \alpha_{v_3}$ and $\alpha_{w_3} \ge \alpha_{v_3}$. Let $w \in \{w_i : i \in [3]\}$ such that $|\alpha_w - \alpha_{v_3}|$ is minimum. We distinguish the case depending on whether $\alpha_w = \alpha_{v_3}$ or not.

Case 1: $\alpha_w \neq \alpha_{v_3}$. First assume that $w = w_3$. As $\{(v_i, w_i) : i \in [3]\}$ is an induced matching, we have $\beta_{v_1} < \gamma_{w_3}$ and $\beta_{v_3} < \gamma_{w_1}$. This implies that S_{v_1} and T_{w_1} do not meet, a contradiction. Now, assume that $w \neq w_3$ and thus, S_{v_3} and T_w do not meet. Let v^* be the vertex where (v^*, w) is in the induced matching. As $\{(v_i, w_i) : i \in [3]\}$ is an induced matching, $\beta_{v^*} < \gamma_{w_3}$ and $\gamma_w > \beta_{v_3}$. Then S_{v^*} cannot meet T_w, a contradiction.

Case 2: $\alpha_w = \alpha_{v_3}$. In this case, w must be w_3. If $\gamma_{w_3} \leq \beta_{v_3}$, then S_{v_1} and T_{w_1} cannot meet. So, $\gamma_{w_3} > \beta_{v_3}$, and it is not difficult to verify that $\beta_{v_3} < \beta_{v_1}, \beta_{v_2}, \gamma_{w_1}, \gamma_{w_2} < \gamma_{w_3}$, otherwise, $\{(v_i, w_i) : i \in [3]\}$ cannot be an induced matching. If $\beta_{v_1} \leq \beta_{v_2}$, then T_{w_1} has to meet S_{v_2}, a contradiction. Similarly, if $\beta_{v_2} < \beta_{v_1}$, then T_{w_2} has to meet S_{v_1}, a contradiction.

It shows that $\text{mim}^+_G(A) \leq 2$, as claimed. By a symmetric argument, we have $\text{mim}^-_G(A) \leq 2$ and $\text{bimim}_G(A) \leq 4$.

We conclude that L has bi-mim-width at most 4. □

4.3 | Rooted directed path digraphs

A *directed tree* is an orientation of a tree, and it is *rooted* if there is a root node r such that every vertex is reachable from r by a directed path. Gavril [24] introduced the class of *rooted directed path graphs*, that are intersection graphs of directed paths in a rooted directed tree. We introduce its directed analog.

For a directed path P, the first and last vertices of P are called the *tail* and *head* of P, respectively.

Definition 7 (Rooted directed path digraph)**.** A digraph G is a *rooted directed path digraph* if there is a rooted directed tree T and a family $\{(S_v, T_v) : v \in V(G)\}$ of pairs of directed paths in T where G is the intersection digraph with representation $\{(S_v, T_v) : v \in V(G)\}$. A directed rooted path digraph is *adjusted* if there is such a representation where for every $v \in V(G)$, the heads of S_v and T_v are the same.

Clearly, interval digraphs are rooted directed path digraphs, and therefore, rooted directed path digraphs have unbounded bi-mim-width. We prove that adjusted rooted directed path digraphs have bounded bi-mim-width and have unbounded linear bi-mim-width. Regarding the notion of *adjusted* rooted directed path digraphs, note once more that fixing one "endpoint" of the objects in each pair of the intersection representation to be the same is in line with the definition of adjusted interval digraphs. Moreover, any intersection representation of an adjusted interval digraph can naturally corresponds to one that shows that it is an adjusted rooted directed path digraph in our definition.

Proposition 5. *Given a representation of an adjusted rooted directed path digraph G, one can construct in polynomial time a branch decomposition of G of bi-mim-width at most 2. Adjusted rooted directed path digraphs have unbounded linear bi-mim-width.*

Proof. Let F be a rooted directed tree with root node r. Let G be an adjusted rooted directed path digraph with representation $\mathcal{M} = \{(S_v, T_v) : v \in V(G)\}$ where S_v and T_v are directed paths in F.

Now, we modify $(F, \mathcal{M})$ into $(F^*, \mathcal{M}^* = \{(S_v^*, T_v^*) : v \in V(G)\})$ so that F^* is a rooted directed tree and $\mathcal{M}^*$ is an adjusted rooted directed path digraph representation of G consisting of directed paths in F^* with additional conditions that

- every internal node of F^* has out-degree at most 2,
- for every vertex $v \in V(G)$, the head α_v of S_v^* has out-degree at most 1 in F^*, and
- nodes in $(\alpha_v : v \in V(G))$ are pairwise distinct.

Let $(F_1, \mathcal{M}_1) := (F, \mathcal{M})$. For every $v \in V(G)$, let α_v^1 be the head of S_v in F. We recursively construct $(F_i, \mathcal{M}_i = \{(S_v^i, T_v^i) : v \in V(G)\}), (\alpha_v^i : v \in V(G)))$ until we get the desired conditions.

- Assume that F_i has a node t of out-degree at least 3. Let $t_1, ..., t_x$ be the out-neighbors of t in F_i. Now, we remove all edges between t and $\{t_1, ..., t_x\}$, and then add a directed path $p_1 p_2 \cdots p_x$ and add an edge (t, p_1) and edges (p_i, t_i) for all $i \in [x]$. Now, for every path P containing tt_j for some j in the representation $\mathcal{M}_i$ we replace tt_j with a path $tp_1 p_2 \cdots p_j t_j$. The other paths do not change. The resulting rooted directed tree and representation are F_{i+1} and $\mathcal{M}_{i+1}$, respectively.
- Assume that $\alpha_v^i = \alpha_w^i = t$ for some distinct vertices v and w in G. Let q be an out-neighbor of t in F_i if one exists, and otherwise we attach a new node q and add an edge (t, q). We replace (t, q) with a directed path $tt'q$ to obtain F_{i+1}. For every path in $\mathcal{M}_i$ containing (t, q) we replace (t, q) with a directed path $tt'q$, and assign $\alpha_v^{i+1} := t'$. For every path in $\{S_v^i, T_v^i\}$, we extend it by adding tt'. The other paths do not change. The resulting rooted directed tree and representation are F_{i+1} and $\mathcal{M}_{i+1}$, respectively.
- Assume that $\alpha_v^i = t$ for some vertex v in G, where t is a node of out-degree at least 2. Let q be an out-neighbor of t in F_i. We replace (t, q) with a directed path $tt'q$ to obtain F_{i+1}. For every path in $\{S_v^i, T_v^i\}$, we extend it by adding tt', and we assign $\alpha_v^{i+1} := t'$. The other paths do not change. The resulting rooted directed tree and representation are F_{i+1} and $\mathcal{M}_{i+1}$, respectively.

In each iteration, either the number of nodes of out-degree at least 3 decreases, or the number of pairs of vertices v and w for which $\alpha_v^i = \alpha_w^i$ decreases, or the number of vertices v for which α_v^i is a node of out-degree at least 3 decreases. Also, the process in each iteration preserves that the representation is an adjusted rooted directed path digraph representation. Thus, at the end, we obtain an adjusted rooted directed path digraph representation $(F^*, \mathcal{M}^*)$ with the set of common endpoints $(\alpha_v^* : v \in V(G))$, as desired. Let U be the underlying undirected tree of F^*.

For each $v \in V(G)$, we add a new node β_v and add an edge $\beta_v \alpha_v^*$ to U, and obtain a tree U^*. Now, let e be an edge of U^*, and let U_X and U_Y be the connected component of $U^* - e$ where U_Y contains the root of F^*. Let (X_e, Y_e) be the cut of G where X_e is the set of vertices v for which α_v^* is in U_X, and $Y_e = V(G) \backslash X_e$. Let Q be the path in U^* from the root to e.

We claim that $\text{mim}_G^+(X_e) \leq 1$. Suppose for contradiction that there is an induced matching $\{(v_i, w_i) : i \in [2]\}$ in $G[X_e \to Y_e]$. Assume that the distance from e to T_{w_1} in U^* is at most the distance from e to T_{w_2}. Since S_{v_1} and S_{v_2} are directed paths, T_{w_1} and T_{w_2} contain a vertex of Q. It means that S_{v_2} has to meet T_{w_1}, which is a contradiction. If the distance from e to T_{w_2} in U^* is at most the distance from e to T_{w_1}, then S_{v_1} has to meet T_{w_2}, a contradiction. Thus, we have $\text{mim}_G^+(X_e) \leq 1$.

Similarly, we show that $\text{mim}_G^-(X_e) \leq 1$. Suppose for contradiction that there is an induced matching $\{(v_i, w_i) : i \in [2]\}$ in $G[Y_e \to X_e]$. Assume that the distance from e to S_{v_1} in U^* is at most the distance from e to S_{v_2}. Since T_{w_1} and T_{w_2} are directed paths, S_{v_1} and S_{v_2} contain a vertex of Q. It means that T_{w_2} has to meet S_{v_1}, which is a contradiction. If the distance from e to S_{v_2} in U^* is at most the distance from e to S_{v_1}, then T_{w_1} has to meet S_{v_2}, a contradiction. Thus, we have $\text{mim}_G^-(X_e) \le 1$.

These imply that $\text{bimim}_G(X_e) \le 2$. By smoothing degree 2 nodes of U^* (i.e., for a degree two node v with neighbors u and w, we remove v and make u adjacent to w), we obtain a branch decomposition of bi-mim-width at most 2.

Now, we argue that adjusted rooted directed path digraphs have unbounded linear bi-mim-width. It is well known that trees are rooted directed path graphs. Also, Høgemo, Telle, and Vågset [27] proved that trees have unbounded linear mim-width. As trees are rooted directed path graphs, we can obtain the biorientations of trees as adjusted rooted directed path digraphs, where $S_v = T_v$ for all $v \in V(G)$. As trees have unbounded linear mim-width, adjusted rooted directed path digraphs have unbounded linear bi-mim-width. □

4.4 | H-convex digraphs

As a generalization of convex graphs, Brettell, Munaro, and Paulusma [12] introduced H-convex graphs. Note that they defined $\mathcal{H}$-convex graphs, rather than H-convex graphs, where $\mathcal{H}$ is a family of graphs. However, they mostly considered $\mathcal{H}$ as the set of all subdivisions of a fixed graph H, so we may simply call them H-convex graphs. We generalize this notion to H-convex digraphs.

Definition 8 H(-convex digraph)**.** Let H be an undirected graph. A bipartite digraph G with bipartition (A, B) is an *H-convex digraph* if there is a subdivision F of H with $V(F) = A$ such that for every vertex b of B, each of $N_G^+(b)$ and $N_G^-(b)$ induces a connected subgraph of F. An H-convex digraph is *nice* if for every vertex b of B, there is a bi-directed edge between b and some vertex of A.

In principle, H-convex digraphs and H-digraphs are closely related, as $\{(N_G^+(v), N_G^-(v)) : v \in B\}$ can be seen as an H-digraph representation on F. Observe that the way in which we restrict convex H-digraphs to nice convex H-digraphs is quite analogous to the way in which H-digraphs are restricted to reflexive H-digraphs. The latter have representations in which for each vertex, the connected subgraph of the subdivision of H representing its out-neighbors, and the one representing its in-neighbors, intersect. Consider the definition of nice H-convex digraphs and adapt its notation from here. For each vertex $b \in B$, its out- and in-neighborhood induce connected subgraphs of F. These connected subgraphs can be seen as the "models" for the out- and in-neighborhoods. Now, to have a bidirected edge incident with b is the same as requiring these two models to intersect, hence the analogy with reflexive H-digraphs.

We prove that nice H-convex digraphs have linear bi-mim-width at most $12|E(H)|$.

Proposition 6. *Let H be an undirected graph. Given a nice H-convex digraph G with its bipartition (A, B), one can construct in polynomial time a linear branch decomposition of G of bi-mim-width at most* $12|E(H)|$.

Proof. Let $m := |E(H)|$. Let G be a reflexive H-convex digraph with bipartition (X, Y). Let F be a subdivision of H with $X = V(F)$ and let $\mathcal{M} = \{(N_G^+(y), N_G^-(y)) : y \in Y\}$. For each $y \in Y$, we choose $\alpha_y \in N_G^+(y) \cap N_G^-(y)$. By adding more vertices in X if necessary (corresponding to a subdivision of F), we may assume that vertices in $(\alpha_y : y \in Y)$ are pairwise distinct and they are not branching vertices of F. This assumption is possible because of Lemma 1.

We fix a vertex r of F and obtain a BFS ordering of F starting from r. We denote by $v <_B w$ if v appears before w in the BFS ordering. This gives an ordering of $V(F) = X$. Now, we extend it to an ordering L of $V(G)$ by, for each $y \in Y$, putting y just after α_y.

We claim that L has bi-mim-width at most $12m$. We choose a vertex v of G arbitrarily, and let A be the set of vertices in G that are v or a vertex appearing before v in L, and let $B := V(G)\backslash A$. It is sufficient to show that $\text{mim}^+_G(A) \leq 6m$. By symmetry, we will get $\text{mim}^-_G(A) \leq 6m$.

Let $A_X := A \cap X, A_Y := A \cap Y, B_X := B \cap X$, and $B_Y := B \cap Y$. We first show that $\nu(G[A_X \to B_Y]) \leq 4m$.

Suppose for contradiction that $G[A_X \to B_Y]$ contains an induced matching M of size $4m + 1$. As defined in Proposition 1, we define $\mathcal{P}$ as the set of paths in F such that

- for every $P \in \mathcal{P}, P$ is a subpath of some branching path of F and it is a maximal path contained in one of A_X and B_X,
- $\bigcup_{P \in \mathcal{P}} V(P) = V(F)$.

Note that each branching path contains at most two paths in $\mathcal{P}$ that are contained in A_X. By the pigeonhole principle, there is a subset $M_1 = \{(x_i, y_i) : i \in [3]\}$ of M of size three and a path $P \in \mathcal{P}$ with $V(P) \subseteq A_X$ such that for every $(x, y) \in M_1, x$ is in P. Observe that α_{y_i} is not in P for every $i \in [3]$.

Let p_1 and p_2 be the endpoints of P. Observe that for each $i \in [3], N^-_G(y_i)$ contains either the subpath of P from p_1 to x_i or the subpath of P from x_i to p_2. Without loss of generality, we may assume that for each $i \in [2], N^-_G(y_i)$ contains the subpath of P from p_1 to x_i. But this implies that (x_1, y_2) or (x_2, y_1) is an edge, contradicting the assumption that M is an induced matching.

We deduce that $\nu(G[A_X \to B_Y]) \leq 4m$. Note that each branching path contains at most one path in $\mathcal{P}$ that is contained in B_x. Thus, by a similar argument, we can show that $\nu(G[A_Y \to B_X]) \leq 2m$. So, we have $\text{mim}^+_G(A) = \nu(G[A \to B]) \leq 6m$. By symmetry, we get $\text{mim}^-_G(A) \leq 6m$ as well, and these imply that $\text{bimim}_G(A) \leq 12m$, as required. □

Proposition 7. *P_2-convex digraphs have unbounded bi-mim-width.*

Proof. We recall the interval digraph representation of an orientation of the $(n \times n)$-grid, given in Proposition 3. For $i, j \in [n]$, we construct $S_{i,j}$ and $T_{i,j}$ as follows. For every odd integer i, we set

- $S_{i,j} := [2(n+1)j + 2i - 1, 2(n+1)j + 2i + 1]$ and
- $T_{i,j} := [2(n+1)(j-1) + 2i, 2(n+1)(j-1) + 2i]$,

and for every even integer i, we set

- $S_{i,j} := [2(n+1)(j-1) + 2i, 2(n+1)(j-1) + 2i]$ and
- $T_{i,j} := [2(n+1)j + 2i - 1, 2(n+1)(j-1) + 2i + 1]$.

Now, we create a bipartite digraph G_n with bipartition $(A, B = \{v_{i,j} : i, j \in [n]\})$ such that for $i, j \in [n]$,

- if i is odd, then $N^+_G(v_{i,j}) = S_{i,j}$ and $N^-_G(v_{i,j}) = T_{i,j}$,
- if i is even, then $N^+_G(v_{i,j}) = T_{i,j}$ and $N^-_G(v_{i,j}) = S_{i,j}$.

It is not difficult to verify that G_n contains an orientation of the 1-subdivision of the $(n \times n)$-grid as an induced subdigraph. By modifying the proof for the fact that the $(n \times n)$-grid has mim-width at least $n/3$, it is straightforward to show that the 1-subdivision of the $(n \times n)$-grid has mim-width at least $n/3$. Therefore, by Lemma 3, G_n has bi-mim-width at least $n/3$. □

5 | ALGORITHMIC APPLICATIONS

In this section we give the algorithmic applications of the width measure bi-mim-width. In particular, we show that all DLCVS and all DLCVP problems can be solved in XP time parameterized by the bi-mim-width of a given branch decomposition of the input digraph. We do so by adapting the framework of the d-neighborhood equivalence relation introduced by Bui-Xuan et al. [14] to digraphs. For an undirected graph G, given a set $A \subseteq V(G)$, two subsets X and Y of A are d-neighborhood equivalent w.r.t. A if the intersection of the neighborhood of each vertex in $\overline{A}$ with X and Y have the same size, when counting up to d.

In the adaptation of this concept to digraphs, we essentially take the Cartesian product of the d-neighborhood equivalences given by the edges going from A to $\overline{A}$ and the edges going from $\overline{A}$ to A. In the resulting d-bi-neighborhood equivalence relation of a set A of vertices in some digraph G, two subsets X and Y of A are d-bi-neighborhood equivalent, if for each vertex of $\overline{A}$, the number of both its in-neighbors in X and the number of its out-neighbors in X are equal to the number of its in- and out-neighbors in Y, respectively, when counting up to d. We show that this notion allows us to lift the frameworks presented in [14] to the realm of digraphs and prove the aforementioned results.

The rest of this section is organized as follows. In Section 5.1, we formally define the d-bi-neighborhood equivalence relation and show how to efficiently compute descriptions of its equivalence classes. In Section 5.2 we give the algorithms for generalized directed domination problems and in Section 5.3 for the directed vertex partitioning problems. We discuss how to use these algorithms to solve distance-r versions of DLCV problems in Section 5.4.

5.1 | d-Bi-neighborhood-equivalence

We now present the central notion that is used in our algorithms, the d-bi-neighborhood equivalence relation which we introduced informally earlier. The reason why it compares the sizes of the intersection of a neighborhood with subsets only up to some integer d is as follows. The subsets of natural numbers that characterize locally checkable vertex subset/partitioning problems can be fully characterized when counting in- and out-neighbors up to some d, depending on the described problem. Therefore, if a vertex v has more than d, say, in-neighbors in two sets X and Y, then these two sets look the same to v in terms of its in-neighborhood. Think, for instance, of the d-Dominating Set problem, where each vertex outside of the solution needs to have d in-neighbors in it.

In the following definition, we present the d-in-neighborhood equivalence relation and the d-out-neighborhood equivalence relation separated before combining them to the d-bi-neighborhood equivalence relation, since in some proofs it is convenient to only consider the edges going in one direction.

Definition 9. Let $d \in \mathbb{N}$. Let G be a digraph and $A \subseteq V(G)$. For two sets $X, Y \subseteq A$, we say that X and Y are *d-out-neighborhood equivalent w.r.t. A*, written $X \equiv^{+}_{d,A} Y$, if

$$\forall u \in V(G)\backslash A : \min\{d, |N^{-}(u) \cap X|\} = \min\{d, |N^{-}(u) \cap Y|\}.$$

(Since the definition is given in terms of vertices from $\overline{A}$, we consider the directions of the edges in reverse, i.e., we consider $N^{-}(v)$ for $v \in \overline{A}$ when defining $\equiv^{+}$.) Similarly, we say that X and Y are *d-in-neighborhood equivalent w.r.t. A*, written $X \equiv^{-}_{d,A} Y$, if

$$\forall u \in V(G)\backslash A : \min\{d, |N^{+}(u) \cap X|\} = \min\{d, |N^{+}(u) \cap Y|\}.$$

If $X \equiv^{+}_{d,A} Y$ and $X \equiv^{-}_{d,A} Y$ then we say that X and Y are d-bi-neighborhood equivalent w.r.t. A and write $X \equiv^{\pm}_{d,A} Y$.

The run time of the algorithms in this section crucially depends on the number of equivalence classes of the d-bi-neighborhood equivalence relations associated with cuts induced by a branch decomposition of the input graph. For d, G, and A as in the previous definition, we denote by $\text{nec}(\equiv^{\pm}_{d,A})$ the number of equivalence classes of $\equiv^{\pm}_{d,A}$. If (T, λ) is a rooted branch decomposition of G, we let

$$\text{nec}_d(T, \lambda) = \max_{t \in V(T)} \max\left\{\text{nec}\left(\equiv^{\pm}_{d,V_t} \right), \text{nec}\left(\equiv^{\pm}_{d,\overline{V_t}}, \right)\right\}.$$

5.1.1 | Descriptions of equivalence classes of $\equiv^{\pm}_{d,A}$

Since $\equiv^{\pm}_{d,A}$ is an equivalence relation over subsets of A, we cannot trivially enumerate all its equivalence classes without risking an exponential running time. We now show that we can enumerate the equivalence classes with a relatively small overhead depending polynomially on n, d, and $\log(\text{nec}(\equiv^{\pm}_{d,A}))$. This enumeration is based on pairs of vectors called *d-bi-neighborhoods* of a subset X of A, one that describes the in-neighborhood of vertices in $\overline{A}$ intersected with X, and one for the out-neighborhood.

Definition 10. Let G be a digraph, $X \subseteq A \subseteq V(G)$, and $d \in \mathbb{N}$. The *d-out-neighborhood* of X, denoted by $U^{+}_{d,A}(X)$ is a vector in $\{0, 1, ..., d\}^{\overline{A}}$, which stores for every vertex $v \in \overline{A}$ the minimum between d and the number of in-neighbors of v in X. Formally,

$$U^{+}_{d,A}(X) = (\min\{d, |N^{-}(v) \cap X|\})_{v\in\overline{A}}.$$

Similarly, the *d-in-neighborhood*, denoted by $U^{-}_{d,A}(X)$, is the vector

$$U^{-}_{d,A}(X) = (\min\{d, |N^{+}(v) \cap X|\})_{v\in\overline{A}}.$$

We refer to the pair $(U^{+}_{d,A}(X), U^{-}_{d,A}(X))$ as the *d-bi-neighborhood* $U^{\pm}_{d,A}(X)$; and we denote the set of all d-bi-neighborhoods as $\mathcal{U}^{\pm}_{d,A}$.

Observation 1. Let G be a digraph and $X, Y \subseteq A \subseteq V(G)$. Then, $X \equiv^{\pm}_{d,A} Y$ if and only if $U^{\pm}_{d,A}(X) = U^{\pm}_{d,A}(Y)$.

By the previous observation, there is a natural bijection between the d-bi-neighborhoods and the equivalence classes of $\equiv^{\pm}_{d,A}$. In our algorithm we will therefore use the d-bi-neighborhoods as descriptions for the equivalence classes of $\equiv^{\pm}_{d,A}$. We now show that we can efficiently enumerate them. While the ideas are parallel to the algorithm presented in [14], we work directly with the d-bi-neighborhoods rather than with representatives to streamline the presentation.

Lemma 7. *Let G be a digraph on n vertices, $A \subseteq V(G)$, and $d \in \mathbb{N}$. There is an algorithm that enumerates all members of $\mathcal{U}^{\pm}_{d,A}$ in time $\mathcal{O}(\text{nec}(\equiv^{\pm}_{d,A})\log \text{nec}(\equiv^{\pm}_{d,A}) \cdot n^2 \log d)$. Furthermore, for each $Y \in \mathcal{U}^{\pm}_{d,A}$, the algorithm can provide some $X \subseteq A$ with $U^{\pm}_{d,A}(X) = Y$.*

Proof. We describe the procedure to enumerate $\mathcal{U}^{\pm}_{d,A}$ in Algorithm 1. Let us argue that this algorithm is correct. Observe that $\mathtt{U}$ only contains pairwise distinct d-bi-neighborhoods. Suppose for a contradiction that there is a set $X \subseteq A$ such that $U^{\pm}_{d,A}(X) \notin \mathtt{U}$, and assume wlog. that X is a minimal subset of vertices whose d-bi-neighborhood is not contained in $\mathtt{U}$. Let $u \in X$. We know that for all $Y \in \mathtt{U}$, $U^{\pm}_{d,A}(X\backslash\{u\}) \neq Y$, for otherwise, $U^{\pm}_{d,A}(X)$ would have been added to $\mathtt{U}$. But this contradicts the minimality of X.

Algorithm 1 can easily be modified to satisfy the second claim of the lemma: In line 5.1.1, instead of adding only $U^{\pm}_{d,A}(R')$ to $\mathtt{U}$, we may add the pair $(R', U^{\pm}_{d,A}(R'))$.

We analyze the run time as follows. For each d-bi-neighborhood that is added to $\mathtt{U}$, we test for at most n additional sets whether their d-bi-neighborhoods need to be added to $\mathtt{U}$ or not. Together with Observation 1, this implies that we test for at most $\mathcal{O}(\text{nec}(\equiv^{\pm}_{d,A}) \cdot n)$ sets whether they should be added to $\mathtt{U}$ or not. Computing a d-bi-neighborhood happens incrementally which comes at a cost of at most $\mathcal{O}(n)$ time. Since $\mathtt{U}$ is a balanced binary tree, we can check for containment in time $\mathcal{O}(\log(\text{nec}(\equiv^{\pm}_{d,A})) \cdot n \cdot \log d)$. The latter is due to the fact that $\mathtt{U}$ is a tree of height at most $\mathcal{O}(\log(\text{nec}(\equiv^{\pm}_{d,A})))$ where at each node we need to make a comparison between at most $\mathcal{O}(n)$ d-digit numbers. The total run time of the algorithm is at most $\mathcal{O}(\text{nec}(\equiv^{\pm}_{d,A})\log \text{nec}(\equiv^{\pm}_{d,A}) \cdot n^2 \log d)$. □

Algorithm 1. Computing all d-bi-neighborhoods.

```
Input: Digraph $G, A \subseteq V(G)$, and $d \in \mathbb{N}$.
Output: $\mathcal{U}^{\pm}_{d,A}$.
Let U be an empty balanced binary search tree.    ▷ To enable binary search in U we assume a fixed ordering on $\overline{A}$; the vectors in $\mathbb{N}^{\overline{A}}$ can then be compared lexicographically. For convenience, we sometimes address U as a set whose elements consist of the $d$-bi-neighborhoods stored in its leaves.
Let $\mathcal{B} = \{\emptyset\}$.
while $\mathcal{B} \neq \emptyset$ do
  Let $\mathcal{B}' = \emptyset$.
  for all $R \in \mathcal{B}$ do
    for all $v \in A$ do
      Let $R' = R \cup \{v\}$.
      If $U^{\pm}_{d,A}(R') \notin$ U, then insert $U^{\pm}_{d,A}(R')$ into U and add $R'$ to $\mathcal{B}'$.
    end for
  end for
  Let $\mathcal{B} = \mathcal{B}'$.
end while
```

5.2 | Generalized directed domination problems

In this section we use the d-bi-neighborhood equivalence relation to give algorithms for problems that ask for a maximum- or minimum-size set that can be expressed as a $(\sigma^+, \sigma^-, \rho^+, \rho^-)$-set. The algorithm is bottom-up dynamic programming along the given branch decomposition (T, λ) of the input digraph G, which we assume to be rooted in an arbitrary degree two node. For a node $t \in V(T)$, we let V_t be the vertices of G that are mapped to a leaf in the subtree of T rooted at t. Before we proceed with its description, we recall the formal definition here of $(\sigma^+, \sigma^-, \rho^+, \rho^-)$-sets.

Definition 11. Let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$, and let $\Sigma = (\sigma^+, \sigma^-)$ and $R = (\rho^+, \rho^-)$. Let G be a digraph and $S \subseteq V(G)$. We say that $S(\sigma^+, \sigma^-, \rho^+, \rho^-)$-dominates G, or simply that $S(\Sigma, R)$-dominates G, if

$$\forall v \in V(G) : |N^+(v) \cap S| \in \begin{cases} \sigma^+ & \text{if } v \in S, \\ \rho^+ & \text{if } v \notin S \end{cases} \quad \text{and} \quad |N^-(v) \cap S| \in \begin{cases} \sigma^- & \text{if } v \in S, \\ \rho^- & \text{if } v \notin S. \end{cases}$$

For better readability, it is often convenient to gather the sets σ^+ and σ^- as one and the sets ρ^+ and ρ^- as one. We will mostly use the resulting (Σ, R)-notation. We now recall the definition of the d-value of a finite or cofinite set. Informally speaking, this value tells us how far we have to count to completely describe a finite or cofinite set.

Definition 12. Let $d(\mathbb{N}) = 0$. For a finite or cofinite set $\mu \subseteq \mathbb{N}$, let

$$d(\mu) = 1 + \min\{\max \mu, \max \mathbb{N} \backslash \mu\}.$$

For finite or cofinite $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$, $\Sigma = (\sigma^+, \sigma^-)$ and $R = (\rho^+, \rho^-)$

$$d(\sigma^+, \sigma^-, \rho^+, \rho^-) = d(\Sigma, R) = \max\{d(\sigma^+), d(\sigma^-), d(\rho^+), d(\rho^-)\}.$$

As our algorithm progresses, it keeps track of partial solutions that may become a (Σ, R)-set once the computation has finished. This does not necessarily mean that at each node $t \in V(T)$, such a partial solution $X \subseteq V_t$ has to be a (Σ, R)-dominating set of $G[V_t]$. Instead, we additionally consider what is usually referred to as the "expectation from the outside" [14] in the form of a subset Y of $\overline{V_t}$ such that $X \cup Y$ is a (Σ, R)-dominating set of $G[V_t]$. This is captured in the following definition.

Definition 13. Let $\mu^+, \mu^- \subseteq \mathbb{N}$ and let $M = (\mu^+, \mu^-)$. Let G be a digraph, and let $A \subseteq V(G)$ and $X \subseteq V(G)$. We say that X M-dominates A if for all $v \in A$, we have that $|N^+(v) \cap X| \in \mu^+$ and $|N^-(v) \cap X| \in \mu^-$. Let Σ and R be as above. For $X \subseteq A$ and $Y \subseteq \overline{A}$, we say that (X, Y) (Σ, R)-dominates A, if $X \cup Y$ Σ-dominates X and $X \cup Y$ R-dominates $A \backslash X$.

The next lemma shows that the previous definition behaves well with respect to $\equiv^{\pm}_{d,A}$.

Lemma 8. *Let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$ be finite or cofinite, let $\Sigma = (\sigma^+, \sigma^-)$ and $R = (\rho^+, \rho^-)$, and let $d = d(\Sigma, R)$. Let G be a digraph and let $A \subseteq V(G)$. Let $X \subseteq A$ and $Y, Y' \subseteq V(G) \backslash A$ such that $Y \equiv^{\pm}_{d,\overline{A}} Y'$. Then, (X, Y) (Σ, R)-dominates A if and only if (X, Y') (Σ, R)-dominates A.*

Proof. Suppose (X, Y) (Σ, R)-dominates A. Let $v \in A \backslash X$. Since $Y \equiv^{-}_{d,\overline{A}} Y'$, we have that

$$\min\{d, |N^+(v) \cap Y|\} = \min\{d, |N^+(v) \cap Y'|\}. \tag{1}$$

If $|N^+(v) \cap Y| \leq d$, then immediately by (1) we have that $|N^+(v) \cap Y| = |N^+(v) \cap Y'|$. Therefore, $|N^+(v) \cap (X \cup Y)| = |N^+(v) \cap (X \cup Y')| \in \rho^+$. If $|N^+(v) \cap Y| > d$, then by the definition of the d-value we have that for all $n > d, n \in \rho^+$. By (1), this implies that $|N^+(v) \cap Y'| \in \rho^+$, and in particular that $|N^+(v) \cap (X \cup Y')| \in \rho^+$. Similarly we can show that $|N^-(v) \cap (X \cup Y')| \in \rho^-$, and so $X \cup Y'$ R-dominates $A \backslash X$. Since $Y \equiv^{+}_{d,\overline{A}} Y'$, we can use the same arguments to show that for all $v \in X, |N^+(v) \cap (X \cup Y')| \in \sigma^+$ and $|N^-(v) \cap (X \cup Y')| \in \sigma^-$, and we conclude that (X, Y') (Σ, R)-dominates A.

A symmetric argument yields the other direction, that is, if (X, Y') (Σ, R)-dominates A then (X, Y) (Σ, R)-dominates A. □

We now turn to the definition of the table entries. To describe an equivalence class $\mathcal{Q}$ of $\equiv^{\pm}_{d,A}$ we use the d-bi-neighborhoods of its members. Note that by Observation 1, the following notion of a description of an equivalence class is well-defined.

Definition 14. Let G be a digraph, $A \subseteq V(G)$, and $d \in \mathbb{N}$. For an equivalence class $\mathcal{Q}$ of $\equiv^{\pm}_{d,A}$, its *description*, denoted by $\mathsf{desc}(\mathcal{Q})$, is the d-bi-neighborhood of all members of $\mathcal{Q}$.

As the table entries are indexed by equivalence classes of $\equiv^{\pm}_{d,A}$, we use their descriptions as compact representations.

Definition 15. Let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$ be finite or cofinite, let $\Sigma = (\sigma^+, \sigma^-)$, $R = (\rho^+, \rho^-)$, and $d = d(\Sigma, R)$. Let opt stand for min if we consider a minimization problem and for max if we consider a maximization problem. Let G be a digraph with rooted branch decomposition (T, λ) and let $t \in V(T)$. For an equivalence class $\mathcal{Q}_t$ of $\equiv^{\pm}_{d,V_t}$, and an equivalence class $\mathcal{Q}_{\bar{t}}$ of $\equiv^{\pm}_{d,\overline{V_t}}$, we let

$$Tab_t[\mathsf{desc}(\mathcal{Q}_t), \mathsf{desc}(\mathcal{Q}_{\bar{t}})] = \begin{cases} \mathsf{opt}_{S \subseteq V_t} |S|: & S \in \mathcal{Q}_t \text{ and for any } S_{\bar{t}} \in \mathcal{Q}_{\bar{t}} : (S, S_{\bar{t}})\ (\Sigma, R)\text{-dominates} \\ & \quad V_t, \\ \infty & \text{if } \mathsf{opt} = \min \text{ and no such } S \text{ exists}, \\ -\infty & \text{if } \mathsf{opt} = \max \text{ and no such } S \text{ exists}. \end{cases}$$

We use the shorthand "$Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}]$" for "$Tab_t[\mathsf{desc}(\mathcal{Q}_t), \mathsf{desc}(\mathcal{Q}_{\bar{t}})]$."

We first initialize the table entries for all $t \in V(T)$ as follows. We use the algorithm of Lemma 7 to enumerate all descriptions of equivalence classes $\mathcal{Q}_t$ of $\equiv^{\pm}_{d,V_t}$ and $\mathcal{Q}_{\bar{t}}$ of $\equiv^{\pm}_{d,\overline{V_t}}$ and we let

$$Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] = \begin{cases} -\infty & \text{if } \mathsf{opt} = \max \quad \text{and} \\ \infty & \text{if } \mathsf{opt} = \min. \end{cases} \tag{2}$$

5.2.1 | Leaves of T

For a leaf $\ell \in V(T)$, let $v \in V(G)$ be such that $\lambda(v) = \ell$. Clearly, $\equiv^{\pm}_{d,\{v\}}$ has at most two equivalence classes, namely, the one containing $\varnothing$ and the one containing $\{v\}$. For each of the at most $\mathcal{O}(d^2)$ equivalence classes $\mathcal{Q}$ of $\equiv^{\pm}_{d,V(G)\setminus\{v\}}$, let $R \in \mathcal{Q}$ which we can assume is given to us by Lemma 7.

- If $|N^+(v) \cap R| \in \sigma^+$ and $|N^-(v) \cap R| \in \sigma^-$, then $Tab_\ell[\{\{v\}\}, \mathcal{Q}] = 1$.
- If $|N^+(v) \cap R| \in \rho^+$ and $|N^-(v) \cap R| \in \rho^-$, then $Tab_\ell[\{\varnothing\}, \mathcal{Q}] = 0$.

Before we proceed with the description of the algorithm updating the table entries at internal nodes, we give one more auxiliary observation.

Observation 2. Let $d \in \mathbb{N}$, and let G be a digraph with a 3-partition (A, B, W) of $V(G)$. For each equivalence class $\mathcal{Q}_a$ of $\equiv^{\pm}_{d,A}$ and each equivalence class $\mathcal{Q}_b$ of $\equiv^{\pm}_{d,B}$, the following holds. There is an equivalence class $\mathcal{Q}$ of $\equiv^{\pm}_{d,A\cup B}$ such that for all $R_a \in \mathcal{Q}_a$ and $R_b \in \mathcal{Q}_b$, $R_a \cup R_b \in \mathcal{Q}$. Moreover, given a description of $\mathcal{Q}_a$ and a description of $\mathcal{Q}_b$, we can compute a description of $\mathcal{Q}$ in time $\mathcal{O}(|W| \cdot \log d)$.

Proof. The first statement follows immediately from the definitions. For the second statement, let $D = \mathsf{desc}(\mathcal{Q})$, $D_a = \mathsf{desc}(\mathcal{Q}_a)$, and $D_b = \mathsf{desc}(\mathcal{Q}_b)$. Then, each entry of D can be computed by adding the corresponding entry from D_a to the corresponding entry from D_b, capping the result at d. Each such operation can be done in $\mathcal{O}(\log d)$ time, and there are $\mathcal{O}(|W|)$ entries to compute. □

5.2.2 | Internal nodes of T

Let $t \in V(T)$ be an internal node with children a and b.

1. Consider each triple $\mathcal{Q}_a$, $\mathcal{Q}_b$, $\mathcal{Q}_{\bar{t}}$ of equivalence classes of $\equiv^{\pm}_{d,V_a}$, $\equiv^{\pm}_{d,V_b}$, and $\equiv^{\pm}_{d,\overline{V_t}}$, respectively.
2. Let $R_a \in \mathcal{Q}_a$, $R_b \in \mathcal{Q}_b$, and $R_{\bar{t}} \in \mathcal{Q}_{\bar{t}}$. Determine:
   - $\mathcal{Q}_{\bar{a}}$, the equivalence class of $\equiv^{\pm}_{d,\overline{V_a}}$ containing $R_b \cup R_{\bar{t}}$.
   - $\mathcal{Q}_{\bar{b}}$, the equivalence class of $\equiv^{\pm}_{d,\overline{V_b}}$ containing $R_a \cup R_{\bar{t}}$.
   - $\mathcal{Q}_t$, the equivalence class of $\equiv^{\pm}_{d,V_t}$ containing $R_a \cup R_b$.
3. Update $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] = \mathsf{opt}\{Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}], Tab_a[\mathcal{Q}_a, \mathcal{Q}_{\bar{a}}] + Tab_b[\mathcal{Q}_b, \mathcal{Q}_{\bar{b}}]\}$.

The next two lemmas establish the correctness of the above algorithm.

Lemma 9. *Let Σ, R be as above. Let G be a digraph and let (A, B, W) be a 3-partition of $V(G)$. Let $S_a \subseteq A$, $S_b \subseteq B$, and $S_w \subseteq W$. Then $(S_a, S_b \cup S_w)$ (Σ, R)-dominates A and $(S_b, S_a \cup S_w)$ (Σ, R)-dominates B if and only if $(S_a \cup S_b, S_w)$ (Σ, R)-dominates $A \cup B$.*

Proof. Suppose $(S_a, S_b \cup S_w)$ (Σ, R)-dominates A and $(S_b, S_a \cup S_w)$ (Σ, R)-dominates B; let $S = S_a \cup S_b \cup S_w$. Then, S Σ-dominates S_a and S_b, and therefore S Σ-dominates $S_a \cup S_b$. Moreover, S R-dominates $A \backslash S_a$ and $B \backslash S_b$, so S R-dominates $(A \cup B) \backslash (S_a \cup S_b)$ which yields that $(S_a \cup S_b, S_w)$ (Σ, R)-dominates $A \cup B$. The other direction follows similarly. □

Lemma 10. *For each node $t \in V(T)$, the table entries in Tab_t are computed correctly.*

Proof. We prove the lemma by induction on the height of t. In the base case, t is a leaf. Correctness in this case is immediate. Suppose t is an internal node with children a and b.

For the first direction, assume that $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] = k$ for an equivalence class $\mathcal{Q}_t$ of $\equiv^{\pm}_{d,V_t}$ and an equivalence class $\mathcal{Q}_{\bar{t}}$ of $\equiv^{\pm}_{d,\overline{V_t}}$. We show that in this case, there is a set $S \in \mathcal{Q}_t$ of size k such that for all $S_{\bar{t}} \in \mathcal{Q}_t$, $(S, S_{\bar{t}})$ (Σ, R)-dominates V_t. By the update of the internal nodes and by the induction hypothesis, there are equivalence classes $\mathcal{Q}_a$ of $\equiv^{\pm}_{d,V_a}$, $\mathcal{Q}_{\bar{a}}$ of $\equiv^{\pm}_{d,\overline{V_a}}$, $\mathcal{Q}_b$ of $\equiv^{\pm}_{d,V_b}$, and $\mathcal{Q}_{\bar{b}}$ of $\equiv^{\pm}_{d,\overline{V_b}}$ such that: There exist $S_a \in \mathcal{Q}_a$, $S_b \in \mathcal{Q}_b$ with $|S_a| + |S_b| = k$ such that for all $S_{\bar{a}} \in \mathcal{Q}_{\bar{a}}$ and $S_{\bar{b}} \in \mathcal{Q}_{\bar{b}}$, $(S_a, S_{\bar{a}})$ (Σ, R)-dominates V_a, and $(S_b, S_{\bar{b}})$ (Σ, R)-dominates V_b. Additionally, $S_a \cup R_{\bar{t}} \in \mathcal{Q}_{\bar{b}}$ and $S_b \cup R_{\bar{t}} \in \mathcal{Q}_{\bar{a}}$, where $R_{\bar{t}} \in \mathcal{Q}_{\bar{t}}$. Using Lemma 8, we conclude that $(S_a, S_b \cup R_{\bar{t}})$ (Σ, R)-dominates V_a and that $(S_b, S_a \cup R_{\bar{t}})$ (Σ, R)-dominates V_b. Lemma 9 yields that $(S_a \cup S_b, R_{\bar{t}})$ (Σ, R)-dominates $V_a \cup V_b = V_t$.

For the other direction, suppose that $\mathsf{opt} = \min$ and note that the case of $\mathsf{opt} = \max$ is analogous. We have to show for every pair of an equivalence class $\mathcal{Q}_t$ of $\equiv^{\pm}_{d,V_t}$ and an equivalence class $\mathcal{Q}_{\bar{t}}$ of $\equiv^{\pm}_{d,\overline{V_t}}$, and $R_{\bar{t}} \in \mathcal{Q}_{\bar{t}}$ that if there exists some $S_t \in \mathcal{Q}_t$ of size at most k such that $(S_t, R_{\bar{t}})$ (Σ, R)-dominates V_t, then $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] \le |S_t|$. Let $S_a := S_t \cap V_a$ and $S_b := S_t \cap V_b$. At some point, the algorithm considered the equivalence classes $\mathcal{Q}_a$ of $\equiv^{\pm}_{d,V_a}$ and $\mathcal{Q}_b$ of $\equiv^{\pm}_{d,V_b}$ such that $S_a \in \mathcal{Q}_a$ and $S_b \in \mathcal{Q}_b$. Since $(S_a \cup S_b, R_{\bar{t}})$ (Σ, R)-dominates $V_a \cup V_b$, it follows from Lemma 9 that $(S_a, S_b \cup R_{\bar{t}})$ (Σ, R)-dominates V_a. Note that by the above algorithm, $S_b \cup R_{\bar{t}} \in \mathcal{Q}_{\bar{a}}$, so by the induction hypothesis we have that

$Tab_a[\mathcal{Q}_a, \mathcal{Q}_{\bar{a}}] \le |S_a|$. Similarly we can deduce that $Tab_b[\mathcal{Q}_b, \mathcal{Q}_{\bar{b}}] \le |S_b|$. Clearly, $S_a \cup S_b = S_t \in \mathcal{Q}_t$, so the algorithm above guarantees that $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] \le |S_t|$. □

The algorithm described above results in the following theorem which is the first main result of this section.

Theorem 4. *Let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$ be finite or cofinite, $\Sigma = (\sigma^+, \sigma^-)$, $R = (\rho^+, \rho^-)$, and $d = d(\Sigma, R)$. There is an algorithm that given a digraph G on n vertices together with one of its branch decompositions (T, λ), computes and optimum-size (Σ, R)-dominating set in time $\mathcal{O}(\mathrm{nec}_d(T,\lambda)^3 \log(\mathrm{nec}_d(T,\lambda)) \cdot n^3 \log d)$. For $n \le \mathrm{nec}_d(T,\lambda)$, the algorithm runs in time $\mathcal{O}(\mathrm{nec}_d(T,\lambda)^3 \cdot \log(\mathrm{nec}_d(T,\lambda)) \cdot n^2 \log d)$.*

Proof. First, we subdivide an arbitrary edge of T, and make the node created in the subdivision, say r, the root of T. For simplicity, we keep referring to the tree created this way as T. We do bottom-up dynamic programming along T. We first initialize the table entries at all nodes as described in Equation (2). At each node $t \in V(T)$, we perform the update of all table entries as described above in the corresponding paragraph, depending on whether t is a leaf or an internal node. We find the solution to the instance at hand at the table entry $Tab_r[2^{V(G)}, \{\emptyset\}]$.

Correctness of the algorithm follows from Lemma 10; we now analyze its run time. We may assume that $|V(T)| = \mathcal{O}(n)$. By Lemma 7, we can compute all descriptions of the equivalence classes of all equivalence relations associated with the nodes of T, and therefore initialize the table entries, in time at most $\mathcal{O}(\mathrm{nec}_d(T,\lambda) \log \mathrm{nec}_d(T,\lambda) \cdot n^3 \log d)$.

Updating the entries at leaf nodes takes time $\mathcal{O}(d^2 \cdot \log d)$ per node, as there are $\mathcal{O}(d^2)$ pairs of equivalence classes to consider. This is upper bounded by $\mathcal{O}(\mathrm{nec}_d(T,\lambda) \cdot \log d)$ by definition, so we spend at most $\mathcal{O}(\mathrm{nec}_d(T,\lambda) \cdot n \cdot \log d)$ time in total for leaf nodes. At each internal node, we consider triples of equivalence classes, of which there are at most $\mathrm{nec}_d(T,\lambda)^3$. Once such a triple is fixed, we can compute descriptions of the remaining relevant equivalence classes in time $\mathcal{O}(n \log d)$ by Observation 2. Lastly, we need to access a constant number of table entries. We can assume that the tables are stored in balanced binary search trees, where the descriptions of the equivalence classes are used as the keys. The overhead of querying one table entry is therefore $\mathcal{O}(\log(\mathrm{nec}_d(T,\lambda)) \cdot n \cdot \log d)$, with $\mathcal{O}(\log(\mathrm{nec}_d(T,\lambda)))$ being the height of the tree, and at each node, we need to make an entry-wise comparison of two sequences of $\mathcal{O}(n)$ d-digit numbers, which is doable in $\mathcal{O}(n \cdot \log d)$ time. This implies that updating the table entries at all internal nodes takes time at most $\mathcal{O}(\mathrm{nec}_d(T,\lambda)^3 \log(\mathrm{nec}_d(T,\lambda)) \cdot n^2 \cdot \log d)$. We can bound the total run time of the algorithm by $\max\{\mathcal{O}(\mathrm{nec}_d(T,\lambda) \log \mathrm{nec}_d(T,\lambda) \cdot n^3 \log d), \mathcal{O}(\mathrm{nec}_d(T,\lambda)^3 \cdot \log(\mathrm{nec}_d(T,\lambda)) \cdot n^2 \log d)\}$. □

5.2.3 | Run time in terms of bi-mim-width

We now show how to express the run time of the algorithm from Theorem 4 as an XP-run time parameterized by the bi-mim-width of (T, λ), in analogy with the case of undirected graphs [10]. The crucial observation is the following. (Recall that for an equivalence relation ~, we denote by nec(~) the number of equivalence classes of ~.)

Observation 3. For $d \in \mathbb{N}$, a digraph G, and $A \subseteq V(G)$: $\text{nec}(\equiv^{\pm}_{d,A}) \leq n^{d \cdot \text{bimim}_G(A)}$.

Proof. By the same arguments given in [10, Lemma 2] for the undirected case, we can show that $\text{nec}(\equiv^{+}_{d,A}) \leq n^{d \cdot \text{mim}^{+}_{G}(A)}$ and that $\text{nec}(\equiv^{-}_{d,A}) \leq n^{d \cdot \text{mim}^{-}_{G}(A)}$. Therefore, $\text{nec}(\equiv^{\pm}_{d,A}) = \text{nec}(\equiv^{+}_{d,A}) \cdot \text{nec}(\equiv^{-}_{d,A}) \leq n^{d \cdot \text{mim}^{+}_{G}(A)} \cdot n^{d \cdot \text{mim}^{-}_{G}(A)} = n^{d \cdot \text{bimim}_G(A)}$. □

Corollary 2. *Let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$ be finite or cofinite, $\Sigma = (\sigma^+, \sigma^-), R = (\rho^+, \rho^-)$, and $d = d(\Sigma, R)$. Let G be a digraph on n vertices with branch decomposition (T, λ) of bi-mim-width $w \geq 1$. There is an algorithm that given any such G and (T, λ) computes an optimum-size (Σ, R)-dominating set in time $\mathcal{O}(n^{3wd+2}w(\log n)(\log d))$ which is in $n^{3wd+2+o(1)}w$.*

In the simplification of the run time bound of the previous corollary we used that we can assume that in each instance, $d \leq n$.

5.3 | Directed vertex partitioning problems

We now show that the locally checkable vertex partitioning problems can be solved in XP time parameterized by the bi-mim-width of a given branch decomposition. In analogy with [14], we lift the d-bi-neighborhood equivalence to q-tuples over vertex sets, which allows for devising the desired dynamic programming algorithm. We omit several technical details as they are very similar to the ones in the previous section. We begin by recalling the definition of a bi-neighborhood constraint matrix.

Definition 16. A *bi-neighborhood-constraint* matrix is a $(q \times q)$-matrix D_q over pairs of finite or cofinite sets of natural numbers. Let G be a digraph, and $\mathcal{X} = (X_1, \ldots, X_q)$ be a q-partition of $V(G)$. We say that $\mathcal{X}$ is a *D_q-partition* if for all $i, j \in \{1, \ldots, q\}$ with $D_q[i, j] = (\mu^+_{i,j}, \mu^-_{i,j})$, we have that for all $v \in X_i$, $|N^+(v) \cap X_j| \in \mu^+_{i,j}$ and $|N^-(v) \cap X_j| \in \mu^-_{i,j}$. The *$d$-value* of D_q is $d(D_q) = \max_{i,j}\{d(\mu^+_{i,j}), d(\mu^-_{i,j})\}$.

Definition 17. Let G be a digraph and $A \subseteq V(G)$. Two q-tuples of subsets of A, $\mathcal{X} = (X_1, \ldots, X_q)$ and $\mathcal{Y} = (Y_1, \ldots, Y_q)$, are *$d$-bi-neighborhood equivalent* w.r.t. A, if

$$\begin{aligned} \forall i \in [q]\ \forall v \in \overline{A}: \quad & \min\{d, |N^+(v) \cap X_i|\} = \min\{d, |N^+(v) \cap Y_i|\} \quad \text{and} \\ & \min\{d, |N^-(v) \cap X_i|\} = \min\{d, |N^-(v) \cap Y_i|\}. \end{aligned}$$

In this case we write $\mathcal{X} \equiv^{\pm}_{q,d,A} \mathcal{Y}$.

Observation 4. Let G be a digraph, $A \subseteq V(G)$, and let $\mathcal{X} = (X_1, \ldots, X_q) \in (2^A)^q$ and $\mathcal{Y} = (Y_1, \ldots, Y_q) \in (2^A)^q$. Then, $\mathcal{X} \equiv^{\pm}_{q,d,A} \mathcal{Y}$ if and only if for all $i \in \{1, \ldots, q\}, X_i \equiv^{\pm}_{d,A} Y_i$. Therefore, $\text{nec}(\equiv^{\pm}_{q,d,A}) \leq \text{nec}(\equiv^{\pm}_{d,A})^q$.

For a q-tuple $\mathcal{X} = (X_1, \ldots, X_q)$ of subsets of A, the (q, d)-*bi-neighborhood* of $\mathcal{X}$ w.r.t. A is $U^{\pm}_{q,d,A}(\mathcal{X}) = (U^{\pm}_{d,A}(X_1), \ldots, U^{\pm}_{d,A}(X_q))$, and we denote the set of all (q, d)-bi-neighborhoods w.r.t.

A by $\mathcal{U}^{\pm}_{q,d,A}$. By the previous observation, for any $A \subseteq V(G)$, there is a natural bijection between the (q, d)-bi-neighborhoods w.r.t. A and the equivalence classes of $\equiv^{\pm}_{q,d,A}$. Furthermore, we can enumerate $\mathcal{U}^{\pm}_{q,d,A}$ by invoking the algorithm of Lemma 7 q times and then generating all q-tuples of $\mathcal{U}^{\pm}_{d,A}$.

Corollary 3. *Let G be a digraph on n vertices, $A \subseteq V(G)$, and $q, d \in \mathbb{N}$ with $q \geq 2$. There is an algorithm that enumerates all members of $\mathcal{U}^{\pm}_{q,d,A}$ in time $\mathcal{O}(\text{nec}(\equiv^{\pm}_{d,A})^{q} \cdot n^{2} q \log d)$. Furthermore, for each $\mathcal{Y} \in \mathcal{U}^{\pm}_{q,d,A}$, the algorithm provides some $\mathcal{X} \in (2^{A})^{q}$ with $U^{\pm}_{q,d,A}(\mathcal{X}) = \mathcal{Y}$.*

Definition 18. Let D_q be a bi-neighborhood constraint matrix. Let G be a digraph and $A \subseteq V(G)$. Let $\mathcal{X} = (X_1, \ldots, X_q) \in (2^{A})^{q}$ and $\mathcal{Y} = (Y_1, \ldots, Y_q) \in (2^{\overline{A}})^{q}$. We say that $(\mathcal{X}, \mathcal{Y})$ D_q-dominates A if for all $i, j \in \{1, \ldots, q\}$, $X_i \cup Y_j$ $D_q[i, j]$-dominates A.

For an equivalence class $\mathcal{Q}$ of $\equiv^{\pm}_{q,d,A}$, its *description*, $\mathsf{desc}(\mathcal{Q})$, is the (q, d)-bi-neighborhood of all its members. Again we index the table entries with descriptions of equivalence classes. For a clearer presentation we will also here skip explicit mentions of the desc-operator.

Definition 19. Let D_q be a bi-neighborhood constraint matrix with $d(D_q) = d$, and let G be a digraph with rooted branch decomposition (T, λ), and $t \in V(T)$. Let $\mathcal{Q}_t$ be an equivalence class of $\equiv^{\pm}_{q,d,V_t}$ and $\mathcal{Q}_{\bar{t}}$ be an equivalence class of $\equiv^{\pm}_{q,d,\overline{V_t}}$. Then,

$$Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] \coloneqq \begin{cases} \texttt{True} & \text{if } \exists \ q\text{-partition } \mathcal{S} \text{ of } V_t \text{ such that:} \\ & \mathcal{S} \in \mathcal{Q}_t \text{ and for all } \mathcal{R}_{\bar{t}} \in \mathcal{Q}_{\bar{t}} : (\mathcal{S}, \mathcal{R}_{\bar{t}})\, D_q\text{-dominates } V_t; \\ \texttt{False} & \text{otherwise.} \end{cases}$$

By the definition of the table entries we have that G has a D_q-partition if and only if some entry in Tab_r is true, where r is the root of the given branch decomposition of G. We now describe the algorithm. Initially, we set all table entries at all nodes to `False`.

5.3.1 | Leaves of T

If $\ell \in V(T)$ is a leaf of T, then let $v \in V(G)$ be such that $\lambda(v) = \ell$. We have to consider the following q q-partitions of $\{v\}$ (recall that parts of a partition may be empty): For $i \in \{1, \ldots, q\}$, we have to consider the partition $\mathcal{X}_i = (X_1, \ldots, X_q)$ where $X_i = \{v\}$ and for $j \neq i$, $X_j = \emptyset$. While these partitions are equal up to renaming, they might differ with respect to D_q. We have that $U^{\pm}_{q,d,\{v\}}(\mathcal{X}_i)$ is the all-zeroes vector in coordinates $j \neq i$, and $U^{\pm}_{d,A}(\{v\})$ in coordinate i. We denote the corresponding equivalence class of $\equiv^{\pm}_{q,d,\{v\}}$ by $\mathcal{Q}_i$. For each equivalence class $\mathcal{Q}_{\overline{\ell}}$ of $\equiv^{\pm}_{q,d,V(G)\setminus\{v\}}$, let $\mathcal{Y} = (Y_1, \ldots, Y_{\ell})$ be one of its elements. We then perform the following updates: For all j, let $(\mu^{+}_{i,j}, \mu^{-}_{i,j}) = D_q[i, j]$; then

$$Tab_{\ell}[\mathcal{Q}_i, \mathcal{Q}_{\bar{\ell}}] = \texttt{True} \text{ if } \forall j : |N^+(v) \cap Y_j| \in \mu^+_{i,j} \text{ and } |N^-(v) \cap Y_j| \in \mu^-_{i,j}.$$

In the following, for two q-tuples $(X_1, ..., X_q)$ and $(Y_1, ..., Y_q)$, we denote their coordinate-wise union as $(X_1, ..., X_q) \cup_q (Y_1, ..., Y_q) = (X_1 \cup Y_1, ..., X_q \cup Y_q)$.

5.3.2 | Internal nodes of T

Let $t \in V(T)$ be an internal node with children a and b.

1. Consider each triple $\mathcal{Q}_a$, $\mathcal{Q}_b$, $\mathcal{Q}_{\bar{t}}$ of equivalence classes of $\equiv^{\pm}_{q,d,V_a}$, $\equiv^{\pm}_{q,d,V_b}$, and $\equiv^{\pm}_{q,d,\overline{V_t}}$, respectively.
2. Let $R_a \in \mathcal{Q}_a$, $R_b \in \mathcal{Q}_b$, and $R_{\bar{t}} \in \mathcal{Q}_{\bar{t}}$. Determine:
   - $\mathcal{Q}_{\bar{a}}$, the equivalence class of $\equiv^{\pm}_{q,d,\overline{V_a}}$ containing $R_b \bigcup_q R_{\bar{t}}$.
   - $\mathcal{Q}_{\bar{b}}$, the equivalence class of $\equiv^{\pm}_{q,d,\overline{V_b}}$ containing $R_a \bigcup_q R_{\bar{t}}$.
   - $\mathcal{Q}_t$, the equivalence class of $\equiv^{\pm}_{q,d,V_t}$ containing $R_a \bigcup_q R_b$.
3. If $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] = \texttt{False}$, then update $Tab_t[\mathcal{Q}_t, \mathcal{Q}_{\bar{t}}] = Tab_a[\mathcal{Q}_a, \mathcal{Q}_{\bar{a}}] \wedge Tab_b[\mathcal{Q}_b, \mathcal{Q}_{\bar{b}}]$.

Applying the arguments given in the proof of Lemma 10 to each part of the corresponding partitions yields the correctness of the resulting algorithm. Also the running time can be analyzed in a similar way, using Corollary 3 and Observation 4 to bound the complexity of enumerating all equivalence classes of the equivalence relations $\equiv^{\pm}_{q,d,\cdot}$. We have the following theorem.

Theorem 5. *Let D_q be a bi-neighborhood constraint matrix with $d = d(D_q)$. There is an algorithm that given a digraph G on n vertices together with one of its branch decompositions (T, λ), determines whether G has a D_q-partition in time $\mathcal{O}(\text{nec}_d(T, \lambda)^{3q} \log(\text{nec}_d(T, \lambda)) \cdot n^3 q \log d)$. For $n \leq \text{nec}_d(T, \lambda)$, the algorithm runs in time $\mathcal{O}(\text{nec}_d(T, \lambda)^{3q} \log(\text{nec}_d(T, \lambda)) \cdot n^2 q \log d)$.*

Combining the previous theorem with Observation 3 gives the following algorithms parameterized by the bi-mim-width of a given branch decomposition.

Corollary 4. *Let D_q be a bi-neighborhood constraint matrix with $d = d(D_q)$. Let G be a digraph on n vertices with branch decomposition (T, λ) of bi-mim-width w. There is an algorithm that given any such G and (T, λ) decides whether G has a D_q-partition in time $\mathcal{O}(n^{3wqd+2} wq (\log n)(\log d))$ which is in $n^{3wqd+2+o(1)} wq$.*

5.4 | Distance-r variants

We now turn to distance variants of all problems considered in this section so far. For instance, for $r \in \mathbb{N}$, the Distance-r Dominating Set problem asks for a minimum size set S of vertices of a digraph G, such that each vertex in $V(G) \setminus S$ is at distance at most r from a vertex in S. Note that for $r = 1$, we recover the Dominating Set problem. We can generalize all DLCV problems to their distance-versions.

Definition 20. Let G be a digraph. For $r \in \mathbb{N}$, the *r-out-ball* of a vertex $v \in V(G)$ is the set of vertices $B_r^+ = \{w \in V(G) | \mathrm{dist}_G(v, w) \leq r\}$, and the *$r$-in-ball* of a vertex v is the set of vertices $B_r^-(v) = \{u \in V(G) | \mathrm{dist}_G(u, v) \leq r\}$.

In distance-r versions of DLCV problems, restrictions are posed on $B_r^+(v)$ instead of $N^+(v)$ and on $B_r^-(v)$ instead of $N^-(v)$.

Definition 21. Let $r \in \mathbb{N}$; let $\sigma^+, \sigma^-, \rho^+, \rho^-$ be finite or cofinite subsets of $\mathbb{N}$, let $\Sigma = (\sigma^+, \sigma^-)$ and $R = (\rho^+, \rho^-)$. Let G be a digraph and $S \subseteq V(G)$. We say that S *distance-r (Σ, R)-dominates* G, if

$$\forall v \in V(G) : \left|B_r^+(v) \cap S\right| \in \begin{cases} \sigma^+ & \text{if } v \in S, \\ \rho^+ & \text{if } v \notin S \end{cases} \quad \text{and} \quad \left|B_r^-(v) \cap S\right| \in \begin{cases} \sigma^- & \text{if } v \in S, \\ \rho^- & \text{if } v \notin S. \end{cases}$$

It is not difficult to see that a set $S \subseteq V(G)$ is a distance-r (Σ, R)-dominating set in G if and only if S is a (Σ, R)-dominating set in G^r, the r-th power of G. Therefore, to solve DISTANCE-r (Σ, R)-SET on G, we can simply compute G^r and solve (Σ, R)-SET on G^r. By Lemma 6 and Corollary 2, we have the following consequence.

Corollary 5. *Let $r \in \mathbb{N}$; let $\sigma^+, \sigma^-, \rho^+, \rho^- \subseteq \mathbb{N}$ be finite or cofinite, $\Sigma = (\sigma^+, \sigma^-), R = (\rho^+, \rho^-)$, and $d = d(\Sigma, R)$. Let G be a digraph on n vertices with branch decomposition (T, λ) of bi-mim-width $w \geq 1$. There is an algorithm that given any such G and (T, λ) computes an optimum-size distance-r (Σ, R)-dominating set in time $\mathcal{O}(n^{3wrd+2} wr (\log n)(\log d))$ which is in $n^{3wrd+2+o(1)} wr$.*

Definition 22. Let D be a $(q \times q)$ bi-neighborhood-constraint matrix. Let G be a digraph. A q-partition $\mathcal{X} = (X_1, \ldots, X_q)$ of $V(G)$ is a *distance-r D-partition* of G, if for all i, j, where $D[i, j] = (\mu_{i,j}^+, \mu_{i,j}^-)$, we have that for all $v \in X_i$, $|B_r^+(v) \cap X_j| \in \mu_{i,j}^+$ and $|B_r^-(v) \cap X_j| \in \mu_{i,j}^-$.

By similar reasoning as above and Lemma 6 and Corollary 4, we have the following.

Corollary 6. *Let $r \in \mathbb{N}$; let D_q be a bi-neighborhood constraint matrix with $d = d(D_q)$. Let G be a digraph on n vertices with branch decomposition (T, λ) of bi-mim-width $w \geq 1$. There is an algorithm that given any such G and (T, λ) decides whether G has a distance-r D_q-partition in time $\mathcal{O}(n^{3wqrd+2} wqr (\log n)(\log d))$ which is in $n^{3wqrd+2+o(1)} wqr$.*

6 | CONCLUSION

We introduced the digraph width measure bi-mim-width, and showed that DLCV problems represented by finite or cofinite sets and their distance-r versions can be solved in polynomial time if the input digraph is given together with a branch decomposition of constant bi-mim-width. A natural next step in the understanding of this new parameter would be to determine the complexity of the DIRECTED FEEDBACK VERTEX SET problem on digraphs of bounded bi-mim-width. We showed that several classes of intersection digraphs have constant bi-mim-width which adds a large number of polynomial-time algorithms for locally checkable problems related to domination and independence (given a representation) to the relatively sparse literature on the subject.

Intersection digraph classes such as interval digraphs seem too complex to give polynomial-time algorithms for optimization problems. Our work points to reflexivity as a reasonable additional restriction to give successful algorithmic applications of intersection digraphs, while maintaining a high degree of generality. This was observed independently for interval digraphs by Francis, Hell, and Jacob [22] who studied the Kernel, Absorbing Set, and Dominating Set problems. Apart from giving polynomial-time algorithms for these problems on reflexive interval digraphs, they showed that even for the severely restricted case when the intervals associated with the vertices are single points, the aforementioned problems remain hard.

Reflexivity presents a natural tractability barrier in the case of interval digraphs, or, more generally, H-digraphs for fixed H. The situation is not as clear yet when considering permutation digraphs or rooted directed path digraphs. Both digraph classes contain interval digraphs, therefore the hardness results from [22] apply as well. However, there are no matching polynomial-time algorithms for directed locally checkable vertex problems on reflexive permutation digraphs or reflexive rooted directed path digraphs; in particular, it is not known whether their bi-mim-width is bounded or not. We did show bounds on the bi-mim-width of their *adjusted* subclasses where we additionally require that every pair of objects representing a vertex share a common "endpoint" (where the concrete notion of an endpoint depends on the considered type of representation). Arguably, reflexivity is the more natural restriction and one would hope that also in the case of these two digraph classes, it is the right barrier separating the tractable cases from the intractable ones. However, this question remains open for the time being.

Let us point out two more directions in this line of work. First, as our algorithms rely on a representation of the input digraphs being provided at the input, we are naturally interested in computing representations of intersection digraph classes of bounded bi-mim-width in polynomial time. So far, this is only known for (reflexive) interval digraphs. Second, in Lemma 6, we proved that the r-th power of a digraph of bi-mim-width w has bi-mim-width at most rw. For undirected graphs, there is a bound that does not depend on r [28]. We leave an open question whether a bound that does not depend on r exists.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

ORCID

Lars Jaffke http://orcid.org/0000-0003-4856-5863
O-joung Kwon http://orcid.org/0000-0003-1820-1962

REFERENCES

1. N. Alon, J. Bang-Jensen, and S. Bessy, *Out-colourings of digraphs*, J. Graph Theory. **93** (2020), no. 1, 88–112. https://doi.org/10.1002/jgt.22476
2. S. Arumugam, K. Jacob, and L. Volkmann, *Total and connected domination in digraphs*, Australas. J. Combin. **39** (2007), 283–292. http://ajc.maths.uq.edu.au/pdf/39/ajc_v39_p283.pdf
3. J. Bang-Jensen and T. M. Christiansen, *Degree constrained 2-partitions of semicomplete digraphs*, Theor. Comput. Sci. **746** (2018), 112–123. https://doi.org/10.1016/j.tcs.2018.06.028
4. J. Bang-Jensen, N. Cohen, and F. Havet, *Finding good 2-partitions of digraphs II. Enumerable properties*, Theor. Comput. Sci. **640** (2016), 1–19. https://doi.org/10.1016/j.tcs.2016.05.034

5. J. Bang-Jensen and G. Gutin (eds.), *Classes of directed graphs*, Springer Monographs in Mathematics, Springer, Cham, 2018, pp. xxii+636. https://doi.org/10.1007/978-3-319-71840-8
6. J. Bang-Jensen, S. Bessy, F. Havet, and A. Yeo, *Out-degree reducing partitions of digraphs*, Theor. Comput. Sci. **719** (2018), 64–72. https://doi.org/10.1016/j.tcs.2017.11.007
7. J. Bang-Jensen, S. Bessy, F. Havet, and A. Yeo, *Bipartite spanning sub(di)graphs induced by 2-partitions*, J. Graph Theory. **92** (2019), no. 2, 130–151. https://doi.org/10.1002/jgt.22444
8. D. W. Bange, A. E. Barkauskas, L. H. Host, and L. H. Clark, *Efficient domination of the orientations of a graph*. Discret Math. **178** (1998), no. 1–3, 1–14. https://doi.org/10.1016/S0012-365X(97)81813-4
9. L. W. Beineke and C. M. Zamfirescu, *Connection digraphs and second-order line digraphs*, Discrete Math. **39** (1982), no. 3, 237–254. https://doi.org/10.1016/0012-365X(82)90147-9
10. R. Belmonte and M. Vatshelle, *Graph classes with structured neighborhoods and algorithmic applications*, Theor. Comput. Sci. **511** (2013), 54–65. https://doi.org/10.1016/j.tcs.2013.01.011
11. M. Biró, M. Hujter, and Z. Tuza, *Precoloring extension. I. Interval graphs*, Discret. Math. **100** (1992), no. 1–3, 267–279. https://doi.org/10.1016/0012-365X(92)90646-W
12. F. Bonomo-Braberman, N. Brettell, A. Munaro, and D. Paulusma, *Solving problems on generalized convex graphs via mim-width*, Proceedings of the 17th International Symposium on Algorithms and Data Structures (WADS 2021), Lecture Notes in Computer Science (A. Lubiw and M. R. Salavatipour, eds.), vol. **12808**, Springer, Cham, 2021, pp. 200–214. https://doi.org/10.1007/978-3-030-83508-8_15
13. A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: A survey*, SIAM Monographs on Discrete Mathematics and Applications (P. L. Hammer, ed.), Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999, pp. xii+304. https://doi.org/10.1137/1.9780898719796
14. B.-M. Bui-Xuan, J. A. Telle, and M. Vatshelle, *Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems*, Theor. Comput. Sci. **511** (2013), 66–76.
15. D. M. Cardoso, M. Kaminski, and V. V. Lozin, *Maximum k-regular induced subgraphs*, J. Comb. Optim. **14** (2007), no. 4, 455–463. https://doi.org/10.1007/s10878-007-9045-9
16. M. Cary, J. Cary, and S. Prabhu, *Independent domination in directed graphs*, Commun. Comb. Optim. **6** (2021), no. 1, 67–80. http://comb-opt.azaruniv.ac.ir/article_14099.html
17. G. Chartrand, P. Dankelmann, M. Schultz, and H. C. Swart, *Twin domination in digraphs*, Ars Combin. **67** (2003), 105–114.
18. B. Courcelle, *The monadic second order logic of graphs VI: On several representations of graphs by relational structures*, Discret. Appl. Math. **54** (1994), no. 2–3, 117–149. https://doi.org/10.1016/0166-218X(94)90019-1
19. R. Diestel, *Graph theory*, Graduate Texts in Mathematics (S. Axler, and K. Ribet, eds.), 4th edn., vol. **173**, Springer, Heidelberg, 2010, pp. xviii+437. https://doi.org/10.1007/978-3-642-14279-6
20. T. Feder, P. Hell, J. Huang, and A. Rafiey, *Interval graphs, adjusted interval digraphs, and reflexive list homomorphisms*, Discrete Appl. Math. **160** (2012), no. 6, 697–707. https://doi.org/10.1016/j.dam.2011.04.016
21. F. V. Fomin, P. A. Golovach, and J.-F. Raymond, *On the tractability of optimization problems on H-graphs*, Algorithmica. **82** (2020), no. 9, 2432–2473. https://doi.org/10.1007/s00453-020-00692-9
22. M. C. Francis, P. Hell, and D. Jacob, *On the kernel and related problems in interval digraphs*, Algorithmica. **85** (2023), no. 6, 1522–1559. https://doi.org/10.1007/s00453-022-01010-1
23. Y. Fu, *Dominating set and converse dominating set of a directed graph*, Amer. Math. Monthly. **75** (1968), no. 8, 861–863. http://www.jstor.org/stable/2314337
24. F. Gavril, *A recognition algorithm for the intersection graphs of directed paths in directed trees*, Discrete Math. **13** (1975), no. 3, 237–249. https://doi.org/10.1016/0012-365X(75)90021-7
25. J. Ghoshal, R. Laskar, and D. Pillone, *Topics on domination in directed graphs*, Domination in Graphs: Advanced Topics (T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, eds.), Taylor & Francis, New York, 2017.
26. P. Hell and J. Nešetřil, *Graphs and homomorphisms*, Oxford Lecture Series in Mathematics and its Applications (J. Ball, and D. Welsh, eds.), vol. **28**, Oxford University Press, Oxford, 2004.
27. S. Høgemo, J. A. Telle, and E. R. Vågset, *Linear mim-width of trees*, Graph-Theoretic Concepts in Computer Science (I. Sau and D. M. Thilikos, eds.), Springer International Publishing, Cham, 2019, pp. 218–231.
28. L. Jaffke, O. Kwon, T. J. F. Strømme, and J. A. Telle, *Mim-width III. Graph powers and generalized distance domination problems*, Theoret. Comput. Sci. **796** (2019), 216–236. https://doi.org/10.1016/j.tcs.2019.09.012
29. T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas, *Directed tree-width*, J. Combin. Theory Ser. B. **82** (2001), no. 1, 138–154. https://doi.org/10.1006/jctb.2000.2031

30. D. Y. Kang, O.-j. Kwon, T. J. F. Strømme, and J. A. Telle, *A width parameter useful for chordal and co-comparability graphs*, Theoret. Comput. Sci. **704** (2017), 1–17. https://doi.org/10.1016/j.tcs.2017.09.006
31. M. M. Kanté, *The rank-width of directed graphs*, preprint arxiv.org/abs/0709.1433 (2007).
32. M. M. Kanté and M. Rao, *The rank-width of edge-coloured graphs*, Theory Comput. Syst. **52** (2013), no. 4, 599–644. https://doi.org/10.1007/s00224-012-9399-y
33. K. Kawarabayashi and S. Kreutzer, *The directed grid theorem*, Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC 2015) (R. A. Servedio and R. Rubinfeld, eds.), ACM, New York, 2015, pp. 655–664. https://doi.org/10.1145/2746539.2746586
34. S. Mengel, *Lower bounds on the mim-width of some graph classes*, Discrete Appl. Math. **248** (2018), 28–32. https://doi.org/10.1016/j.dam.2017.04.043
35. H. Müller, *Recognizing interval digraphs and interval bigraphs in polynomial time*, Discrete Appl. Math. **78** (1997), no. 1–3, 189–205. https://doi.org/10.1016/S0166-218X(97)00027-9
36. L. Ouldrabah, M. Blidia, and A. Bouchou, *On the k-domination number of digraphs*, J. Comb. Optim. **38** (2019), no. 3, 680–688. https://doi.org/10.1007/s10878-019-00405-1
37. S. Oum, *Rank-width and vertex-minors*, J. Combin. Theory Ser. B. **95** (2005), no. 1, 79–100. https://doi.org/10.1016/j.jctb.2005.03.003
38. S.-i. Oum, *Rank-width is less than or equal to branch-width*, J. Graph Theory. **57** (2008), no. 3, 239–244. https://doi.org/10.1002/jgt.20280
39. E. Prisner, *Algorithms for interval catch digraphs*, Discret. Appl. Math. **51** (1994), no. 1–2, 147–157. https://doi.org/10.1016/0166-218X(94)90104-X
40. A. Ramoul and M. Blidia, *A new generalization of kernels in digraphs*, Discret. Appl. Math. **217** (2017), 673–684. https://doi.org/10.1016/j.dam.2016.09.048
41. N. Robertson and P. D. Seymour, *Graph minors. x. Obstructions to tree-decomposition*, J. Combin. Theory Ser. B. **52** (1991), no. 2, 153–190. https://doi.org/10.1016/0095-8956(91)90061-N
42. O. Schaudt, *Efficient total domination in digraphs*, J. Discrete Algorithms. **15** (2012), 32–42. https://doi.org/10.1016/j.jda.2012.02.003
43. M. Sen, S. Das, and D. B. West, *Circular-arc digraphs: A characterization*, J. Graph Theory. **13** (1989), no. 5, 581–592. https://doi.org/10.1002/jgt.3190130508
44. M. Sen, S. Das, A. B. Roy, and D. B. West, *Interval digraphs: An analogue of interval graphs*, J. Graph Theory. **13** (1989), no. 2, 189–202. https://doi.org/10.1002/jgt.3190130206
45. P. Smolíková, *The simple chromatic number of oriented graphs*, Electron. Notes Discrete Math. **5** (2000), 281–283. https://doi.org/10.1016/S1571-0653(05)80186-6
46. É. Sopena, *The chromatic number of oriented graphs*, J. Graph Theory. **25** (1997), no. 3, 191–205. https://doi.org/10.1002/(SICI)1097-0118(199707)25:3<191::AID-JGT3>3.0.CO;2-G
47. É. Sopena, *Homomorphisms and colourings of oriented graphs: An updated survey*. Discret Math. **339** (2016), no. 7, 1993–2005. https://doi.org/10.1016/j.disc.2015.03.018
48. J. A. Telle and A. Proskurowski, *Algorithms for vertex partitioning problems on partial k-trees*, SIAM J. Discrete Math. **10** (1997), no. 4, 529–550. https://doi.org/10.1137/S0895480194275825
49. M. Vatshelle, *New width parameters of graphs*, Ph.D. thesis, University of Bergen, 2012.
50. J. von Neumann, and O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, Princeton, NJ, 1944, pp. xviii+625.

How to cite this article: L. Jaffke, O. Kwon, and J. A. Telle, *Classes of intersection digraphs with good algorithmic properties*, J. Graph Theory. (2023), 1–39. https://doi.org/10.1002/jgt.23065

APPENDIX A: OMITTED DEFINITIONS

Let T be a rooted directed tree. For a vertex $t \in V(T)$, we denote by T_t the subtree of T containing all vertices v such that there is a directed path from t to v in T.

Definition 23 (Strong guard)**.** Let G be a digraph and $X, Y \subseteq V(G)$. We say that Y *is a strong guard for* X if every walk starting and ending in X, and containing a vertex from $V(G)\backslash X$, contains a vertex from Y.

Definition 24 (Directed tree-width)**.** Let G be a digraph. A *directed tree decomposition* is a triple (T, β, γ) of a rooted directed tree T and two maps $\beta : V(T) \to 2^{V(G)}$ and $\gamma : E(T) \to 2^{V(G)}$,

1. The set $\{\beta(t) : t \in V(T)\}$ is a partition of $V(G)$.
2. For each $e = (u, v) \in E(T), \gamma(e)$ is a strong guard for $\bigcup_{t \in V(T_v)} \beta(t)$.

For each $t \in V(T)$, we let $\Gamma(t) := \beta(t) \cup \bigcup_{e \sim t} \gamma(e)$, where $e \sim t$ means that e is incident with t. The *width* of (T, β, γ) is $\max_{t \in V(T)} |\Gamma(t)| - 1$, and the *directed tree-width* of a digraph G is the minimum width over all its directed tree decompositions.