9. Digraphs of Bounded Width

Stephan Kreutzer and O-joung Kwon

9.1 Introduction

Structural parameters for undirected graphs such as the **path-width** or **tree-width** of graphs have played a crucial role in developing a structure theory for graphs based on the minor relation and they have also found many algorithmic applications. Starting in the late 1990s, several ideas for generalizing this theory to digraphs have appeared. Broadly, for the purpose of this chapter, we distinguish these approaches into three categories: *tree-width inspired*, *rank-width inspired* and *density based*. The tree-width inspired approaches are based on the idea of generalizing the concept of undirected tree-width (or path-width) to digraphs. The various attempts, which we will discuss below, all have the goal of generalizing some natural property or some natural characterization of tree-width of undirected graphs to digraphs. In the same way as tree-width can be seen as a global connectivity measure for undirected graphs, the various versions of a directed analogue of tree-width measure global connectivity in digraphs. However, on digraphs, connectivity can be measured in many different natural ways. It turns out that equivalent characterizations of tree-width on undirected graphs yield different concepts on digraphs, with different properties, advantages and disadvantages. We will outline the most prominent of these concepts in Section 9.2 below.

The "tree-width inspired" approaches have in common that they define new classes of digraphs using structural parameters for digraphs which can not also be explained by structural parameters of the underlying undirected graphs. In particular, classes $\mathcal{C}$ of digraphs of, e.g., bounded DAG-width, do not automatically have bounded undirected tree-width (in the sense that the class of undirected graphs obtained from $\mathcal{C}$ by ignoring the direction of arc has bounded tree-width).

Another feature that almost all of these approaches have in common is that the class of DAGs has low width in all these definitions. This is a consequence of the fact that these approaches measure strong connectivity in

S. Kreutzer (✉) and O. Kwon
Chair for Logic and Semantics, Technical University Berlin,
Sekr TEL 7-3 Ernst-Reuter Platz 7, 10587 Berlin, Germany
e-mail: stephan.kreutzer@tu-berlin.de

J. Bang-Jensen and G. Gutin (eds.), *Classes of Directed Graphs*, Springer
Monographs in Mathematics, https://doi.org/10.1007/978-3-319-71840-8_9

various forms. Unfortunately, this does have problematic algorithmic consequences, as many $\mathcal{NP}$-hard computational problems remain hard on acyclic digraphs, and hence remain hard on classes of bounded width in these measures. Therefore, research in algorithmic applications of digraph width measures has tried to develop width measures for separating the class of DAGs into easy and hard instances. The next two types of digraph width measures achieve this goal.

A different approach to digraph width measures is taken in the definition of directed versions of **rank-width** [82] (a graph measure broadly equivalent to **clique-width** [25]).

Clique-width can naturally be defined on digraphs and it was indeed defined this way right from the beginning. However, algorithms for computing clique-width are not based on clique-width but on rank-width of graphs. Rank decompositions can be computed efficiently [82] and from a rank decomposition a clique-width decomposition can be computed.

In order to translate concepts from undirected rank-width, such as **vertex-minors**, to the directed setting, Kanté developed concepts of rank width for digraphs such as **bi-rank-width** and $\mathbb{F}_4$-**rank-width** [56]. This approach has led to a theory of directed rank-width with connections to other types of digraphs. A feature that distinguishes this approach from the tree-width inspired approaches above is that if a class of digraphs has bounded directed clique or rank-width then the class of underlying undirected graphs also has bounded undirected clique width. As a consequence, any graph property definable in **monadic second order logic** can be decided in linear time on any class of digraphs of bounded bi-rank-width [26]. Another consequence of this fact is that the class of DAGs no longer has bounded width. Those DAGs have low width in the tree-width inspired approaches has led to problems for algorithmic applications of tree width based directed width measures as several interesting computational problems remain hard on DAGs. This problem therefore does not appear in classes of bounded bi-rank-width etc.

Whereas on undirected graphs, classes of graphs of bounded tree-width also have bounded clique-width, in the directed setting these concepts are incomparable. We will present the concepts of directed rank-widths in Section 9.9.

A third, and final, approach to digraph width measures covered in this chapter are concepts based on density arguments. In their quest for a solid mathematical definition of "sparse" classes of graphs, Nešetřil and Ossona de Mendez defined classes of graphs of **bounded expansion** and classes which are **nowhere dense** [74, 75]. These concepts can be generalized to digraphs as well and lead to a surprisingly elegant theory. We will cover the resulting theory in Section 9.6.

Overview. The remaining chapter is organized as follows. In Section 9.2 we cover the *tree-width inspired* width measures. In particular, we will briefly introduce *graph searching games* (Section 9.2.1), which provide an intuitive way of defining graph and digraph decompositions, introduce some of the more prominent digraph decompositions (Section 9.2.4 and Theorem 9.2.13) and compare them with respect to generality (Section 9.2.5).

In Section 9.3, we provide a brief overview of the existing structure theory for digraphs based on directed tree-width. In particular we review known obstructions to directed tree width. This also leads to a fixed-parameter algorithm for computing directed tree-decompositions which, together with some algorithmic applications, we present in Section 9.4 and 9.5.

In Sections 9.6 to 9.8 we cover the relatively recent theory of density based width measures: classes of digraphs of bounded expansion (Section 9.7) and nowhere dense classes of digraphs (Section 9.8).

Finally, in the last part of the chapter, Section 9.9, we present the concepts of digraph width measures based on rank-width.

9.2 Tree-Width Inspired Width Measures

In this section we will present some of the best known tree-width inspired width measures for digraphs. Many of them can be explained in terms of **graph searching games** and these games provide an intuitive way to understand these measures. We will therefore first give a brief overview of graph searching games, also known as **Cops and Robber games**.

9.2.1 Graph Searching Games

Graph searching games have been studied intensively in graph theory and they have found a wide range of applications. See [2, 38, 68] for surveys on the subject. Here we will only review the absolute basics needed for our exposition of digraph width measures.

A graph searching game is played on a graph by two players, often called the **cops** and the **robber** or the **searchers** and the **fugitive**. The general goal for the cops is to catch the robber, whereas the robber tries to evade. The cop player controls a number of cops each of which occupies a single vertex of the graph. The robber also occupies a single vertex. In every round of the game, the cop player can move some of the cops from their current position to new positions on the graph or he can place new cops on the graph. However, he first has to announce his move and lift up all cops he wants to move, releasing their current position. Then the robber can react to this by changing his own position. The rules for the robber movement differ between the various types of graph searching games. Finally, the cops are placed on their new positions. If any cop is placed on the vertex occupied by the robber, then the cops win. Otherwise, if the robber can escape forever, he wins.

More formally, given a graph $G = (V(G), E(G))$, a current position in the game can be described by a pair (X, v), where $X \subseteq V(G)$ are the vertices occupied by the cops and $v \in V(G)$ is the vertex occupied by the robber. A single round of the play can therefore be described as a move from a position (X, v) to a new position (X', v'). The game always starts at a position $(\emptyset, v)$, for some vertex $v \in V(G)$.

In most games of interest to us, the cops can move freely, i.e. from the current position (X, v) they can move to any new position X'. The robber is more restricted and the various restrictions on the movement of the robber define different variations of the game. To give an example, the game corresponding exactly to tree-width is played on an undirected graph. From a current position (X, v), once the cops announce their move to X', the robber can choose any vertex v' reachable from v in the graph $G - (X \cap X')$, i.e. any position reachable from v by a path not occupied by a cop that remains on the board.

Given a play $(X_i, v_i)_{0 \leq i < l}$, for some $l \in \mathbb{N} \cup \{\omega\}$, we can define the **width** of the play as $\max\{|X_i| : 0 \leq i < l\}$. In this way, any graph searching game defines a graph parameter assigning to every graph or digraph G the minimal number k such that the cops have a winning strategy against the robber on G of width at most k. A trivial strategy for the cop player to win on any given graph is to put a cop on every single vertex of the graph. Hence, the width is always well defined and it is the minimal number of cops required for a winning strategy that yields an interesting graph parameter.

Graph searching games can be classified in many different ways. An important distinction is whether the cops can always see the robber, called **visible graph searching**, or whether they need to search the graph without knowing where the robber is. This is referred to as **invisible graph searching**. It is known that in the game variant above where the robber can move along any cop free path, the graph parameter defined by the visible variant is exactly tree-width whereas the invisible variant defines path-width [15, 92].

An important concept in graph searching is **monotonicity**. Monotonicity restricts the winning strategies for the cops. We distinguish two forms of monotonicity: **cop monotonicity** and **robber monotonicity**. In a **cop-monotone** strategy, the cop player is not allowed to place a cop on a vertex that had already been occupied by a cop in the past. That is, once a cop is lifted from a vertex $v \in V(G)$, no cop can later on be placed on v. In a **robber-monotone** strategy, the cops have to play in a way such that once, at any particular point in the play, a vertex v is not reachable for the robber, it has to remain unreachable for the rest of the play. More precisely, let $(X_i, v_i)_{0 \leq i < n}$ be a play, for some $n \in \mathbb{N} \cup \{\omega\}$. For all i let R_i be the set of vertices available to the robber starting from v_i in $G - X_i$. The play is robber-monotone, if $R_j \subseteq R_i$ for all $0 \leq i < j$. It is known, that on undirected graphs, in the visible and the invisible graph searching games, the cops have a winning strategy of width k if, and only if, they have a cop- and robber-monotone winning strategy of width k. For digraphs, this will often not be the case and monotone and non-monotone versions will define different parameters, see e.g. [1, 69].

There is a natural correspondence between winning strategies of the cops in a graph searching game and graph decompositions. For instance, in the visible graph searching game on undirected graphs described above, a winning strategy for the cop player can be seen as a tree with the initial position in the game as the root and a child for every possible move of the robber and the corresponding move of the cops. This monotone winning strategy tree immediately defines a tree decomposition of the graph of width one less than the width of the winning strategy. Conversely, a tree decomposition of width k of a graph immediately defines a winning strategy for the cop player of width $k+1$. It is this natural correspondence between winning strategies and graph decompositions that makes graph searching games an elegant characterization of width measures for graphs and digraphs.

9.2.2 Decompositions of Directed Graphs

In the following sections we will define several width measures of directed graphs. All of them are defined in terms of a **decomposition** of digraphs. The type of decompositions will vary but in general they will all have a common structure. A decomposition of a digraph D consists of a digraph T, usually a tree or a DAG, and a labeling function β assigning to every vertex of T a subset of vertices of D. Furthermore, there is a **guarding function** γ that assigns to every arc or to every vertex (or both) a **guard**. Usually, a guard is also a set of vertices. The role of the guard of an arc $e \in A(T)$ is that if $e = (u, v) \in A(T)$ and $X := \bigcup\{\beta(t) : t \text{ is reachable from } v \text{ in } T - e\}$, then $\gamma(e)$ controls connectivity between X and the rest of D. Control can mean that there is no path from X to any vertex outside of X in $D - \gamma(e)$, or that there is no strong component in $D - \gamma(e)$ containing a vertex of X and a vertex not in X. The various types of decompositions defined in the sequel are obtained by varying the type of guards and the type of the decomposition structure T.

Definition 9.2.1 (Strong and Weak Guarding) *Let D be a digraph and let $X, Y \subseteq V(D)$ be sets.*

1. *We say that Y **strongly guards** X, or is a **strong guard** of X, if every directed walk starting and ending in X which contains a vertex of $V(D) \setminus X$ also contains a vertex of Y. In other words, $X \setminus Y$ is the union of the vertex sets of some set of strong components of $D - Y$.*
2. *We say that Y **weakly guards** X, or is a **weak guard** of X, if every arc $e = (u, v) \in A(D)$ with $u \in X \setminus Y$ has $v \in X \cup Y$.*

As an example for the two notions of guarding in the previous definition, consider the set $X := \{3, 4, 5\}$ of vertices in the digraph shown in Figure 9.1 a): The set $\{6, 9\}$ is a weak guard for X. The set $\{6\}$ containing only the vertex 6 is already a strong guard, as every path from X to itself that does contain any vertex not in X must go through 6. But $\{6\}$ is not a weak guard for X.

The names strong and weak guards come from the intuition that strong guards control strong components, i.e. strong connectivity, whereas weak guards control directed paths and therefore weak reachability. Of course, every weak guard is also strong and therefore weak guarding is the more restrictive concept of guards.

Note that for every set X of vertices in a digraph G there is a uniquely defined minimal weak guard, which consists of every vertex in $G \setminus X$ which is an out-neighbour of a vertex in X. But there can be many distinct and even disjoint minimal strong guards.

Definition 9.2.2 (Abstract Digraph Decomposition) *Let D be a digraph. An **abstract digraph decomposition** of D is a triple (T, β, γ), where T is a digraph, $\beta : V(T) \to 2^{V(D)}$ and $\gamma : A(T) \to 2^{V(D)}$ such that $\bigcup\{\beta(t) : t \in V(T)\} = V(D)$.*

For every $t \in V(T)$ we define

$$T_t := T\big[\{s \in V(T) : s \text{ is reachable from } t \text{ by a directed path in } T\}\big]$$

as the subgraph of T induced by the vertices reachable from t. Furthermore, if $S \subseteq T$ then we define $\beta(S) := \bigcup\{\beta(s) : s \in V(S)\}$.

*With every decomposition we will define a **width** $w(t)$ for every $t \in V(T)$. The **width** $w(T, \beta, \gamma)$ is then defined as* $\max\{w(t) : t \in V(T)\}$.

Finally, for every $v \in V(D)$, we define $\beta^{-1}(v) := \{t \in V(T) : v \in \beta(t)\}$.

*Sometimes, guards are more naturally associated with vertices of T instead of its arcs and hence γ is a function from $V(T)$ into $2^{V(D)}$. We call such abstract decompositions **node guarded**.*

Several decompositions below use rooted directed trees as underlying digraph.

Definition 9.2.3 *A **rooted directed tree**[1] is a digraph obtained from an undirected tree by selecting a vertex r as a root and orienting every arc away from the root vertex r.*

9.2.3 Tree-Width Based Digraph Width Measures

In this section we describe some of the most prominent tree-width inspired digraph decompositions proposed in the literature. Throughout the section we will illustrate the different decompositions by the following example digraph shown in Figure 9.1 a).

The first generalization of tree-width to digraphs proposed in the literature was directed tree-width [54, 84].

[1] This is also called an out-tree.

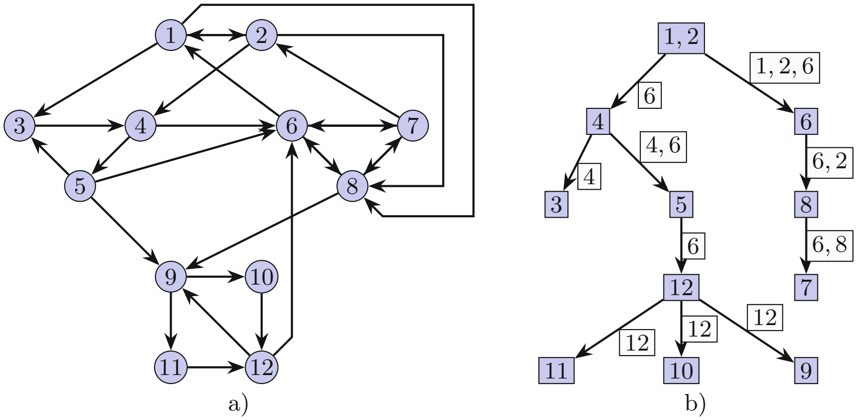

Figure 9.1 a) An example digraph D and b) a directed tree decomposition of D of width 2.

Definition 9.2.4 (Directed Tree Decompositions) *A* ***directed tree decomposition*** *of a digraph D is an abstract digraph decomposition (T, β, γ) such that T is a rooted directed tree, $\{\beta(t) : t \in V(T)\}$ is a partition of $V(D)$ into non-empty sets and for every $e = (s, t) \in A(T)$, $\gamma(e)$ is a strong guard of $\beta(T_t)$.*

For every $t \in V(T)$ we define $\Gamma(t) := \beta(t) \cup \bigcup_{e \sim t} \gamma(e)$ and we define the ***width*** *$w(t)$ as $w(t) := |\Gamma(t)| - 1$, where $e \sim t$ means that the arc e is incident to t.*

See Figure 9.1 b) for an illustration of a directed tree decomposition of the digraph in Figure 9.1. The figure also demonstrates some of the (algorithmically) problematic aspects of directed tree decompositions: The guard 6 on the branch from the root to node labelled by 12 is actually a vertex that is being decomposed in an entirely different subtree of the root. Hence, directed tree decompositions can use vertices in a guard that are contained in strong components which are part of different subtrees. This can cause problems in algorithmic applications. Furthermore, arcs of the digraph D can cross between subtrees in the directed tree decomposition, something that cannot happen in the undirected case. This happens for instance with the arc $(8, 9) \in A(D)$. Finally, on a branch of a directed tree decomposition from its root to a leaf it could happen that a vertex w is contained in a guard of an arc $e = (u, v)$, it then disappears from the next arc of the branch and then reappears later on as a guard on the branch. This phenomenon does not appear in the decomposition on Figure 9.1 but can happen in general.

The second problem, that arcs can cross subtrees – but only in one direction – is an intrinsic feature of directed decompositions. If this were disallowed then we would essentially speak about undirected tree decompositions. The first and the third problem, however, are unavoidable. We will see next the concept of D-decompositions, which are similar to directed tree decompositions but avoid these problems. However, it was shown in [3] that there are classes of digraphs of bounded directed tree-width but unbounded D-width (see Section 9.2.5). The examples separating the two concepts precisely use these properties of guards containing vertices from different strong components as well as guards reappearing along branches, showing that these properties of directed tree decompositions are unavoidable.

In [90], Safari suggests D-width as another structural complexity measure. The definition of D-decompositions is perhaps the most natural translation of undirected tree decompositions to the directed settings in terms of strong connectivity. However, as we will see below, in terms of structural properties, it is directed tree-width that shares most structural properties of undirected tree-width. In the following definition we give a slightly different version of D-decompositions. But the width defined by this concept differs from the original definition at most by a factor of 2. See Figure 9.2 for an illustration.

Definition 9.2.5 (D-Decompositions) *A* ***D-decomposition*** *of a digraph* D *is an abstract digraph decomposition* (T, β, γ) *such that* T *is a rooted directed tree,* $\gamma(e) := \beta(u) \cap \beta(v)$ *for every* $e = (u, v) \in A(T)$ *and* $\gamma(e)$ *is a strong guard of* $\beta(T_v)$ *and* $\beta^{-1}(v)$ *induces a non-empty subtree of* T *for every* $v \in V(D)$.

For every $t \in V(T)$ *we define the* ***width*** $w(t)$ *as* $w(t) := \left|\beta(t)\right|$.

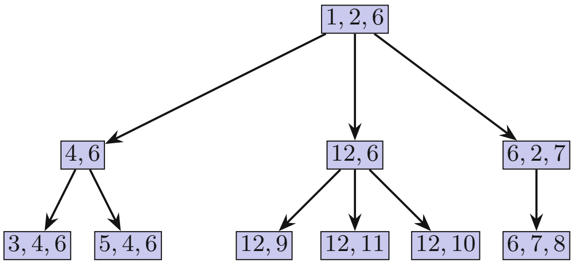

Figure 9.2 A D-decomposition of width 3 of the digraph in Figure 9.1.

Directed tree-width and D-width are related to each other because they both correspond to the same graph searching game, the game where the robber can only stay within a strong component, but they are related to different type of strategies for the cops. The next following three decompositions are based on a different form of game, where the robber can follow any directed path. DAG-width was defined in [11] and independently in [77], cf. [12]. See Figure 9.3 a) for an illustration.

Definition 9.2.6 (DAG Decompositions [12]**)** *Let D be a digraph. A* ***DAG-decomposition*** *of D is an abstract digraph decomposition (T, β, γ) such that:*

1. *T is a DAG.*
2. *$\gamma(e) = \beta(u) \cap \beta(v)$, for every arc $e = (u, v) \in A(T)$, and $\gamma(e)$ is a weak guard of $\beta(T_v)$.*
3. *$\beta(a) \cap \beta(c) \subseteq \beta(b)$ for every triple $a, b, c \in V(T)$ such that a, b, c appear in this order on some directed path in T.*
4. *For every root $t \in V(T)$, $\beta(T_t) = N^+[\beta(T_t)]$.*

For every $t \in V(T)$ we define the width $w(t) := |\beta(t)|$.

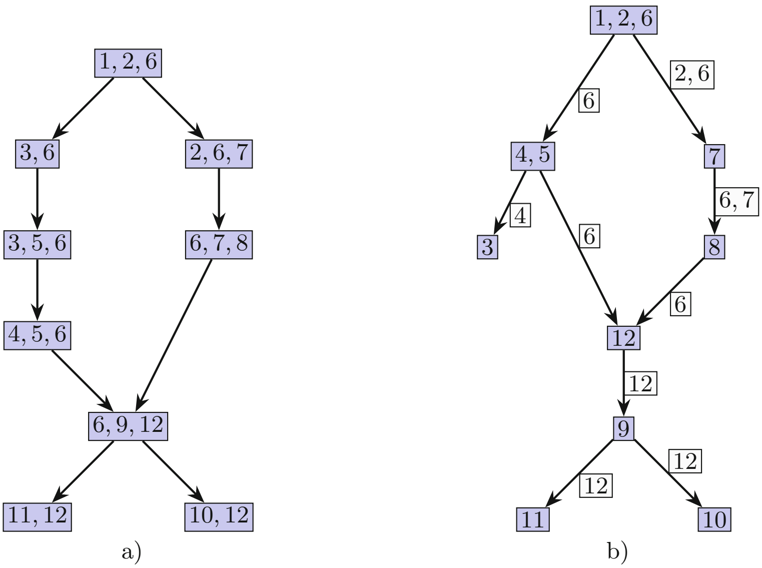

Figure 9.3 a) A DAG-decomposition and b) a Kelly-decomposition of the digraph in Figure 9.1, both of width 3.

A related width measure is Kelly-width which is based on so-called Kelly-decompositions. It was introduced in [51] to overcome some problems of DAG-decompositions. See Figure 9.3 b) for an illustration.

Definition 9.2.7 (Kelly Decompositions [51]**)** *A* ***Kelly-decomposition*** *of a digraph D is a node-guarded abstract decomposition (T, β, γ) so that*

1. *T is a DAG.*
2. *$\{\beta(t) : t \in V(T)\}$ is a partition of $V(D)$ into non-empty subsets.*
3. *$\gamma(t)$ is a weak guard of $\beta(T_t)$ for every $t \in V(T)$.*

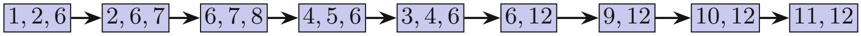

Figure 9.4 A directed path decomposition of the digraph in Figure 9.1 of width 3.

4. For all $s \in V(T)$ *there is a linear order* $<_s$ *on its children* $t_1, \dots, t_p$ *so that for all* $1 \leq i \leq p$, $\gamma(t_i) \subseteq \beta(s) \cup \gamma(s) \cup \bigcup_{j <_s i} \beta(V(T_{t_j}))$.

5. Similarly, there is a linear order $<_r$ *on the roots such that* $\gamma(r_i) \subseteq \bigcup_{j <_r i} \beta(V(T_{r_j}))$.

The width $w(t)$ *of a vertex* $t \in V(T)$ *is defined as* $\beta(t) \cup \gamma(t)$.

Note that the number of nodes in a Kelly-decomposition is at most the number of vertices of the decomposed digraphs, as the bags form a partition. This is not the case for DAG-decompositions and we will see below that DAG-decompositions of optimal width k can become super-polynomially large, i.e. have number of bags proportional to n^{k+1} (see [3]). See Section 9.4.1 for details.

Finally, we introduce the concept of directed path decompositions, introduced by Robin Thomas in the mid-90s but unpublished. See [8, 9] for published references. See Figure 9.4 for an illustration.

Definition 9.2.8 (Directed Path Decompositions) *A* ***directed path decomposition*** *of a digraph* D *is a DAG-decomposition* (T, β, γ) *of* D *such that* T *is a directed path.*

Every type of decomposition introduced above naturally defines a digraph width measure, summarized in the following definition.

Definition 9.2.9 (Directed Width Measures) *Let* D *be a digraph. The* ***directed tree-width*** $\mathrm{dtw}(D)$ *of* D *is defined as the minimum width of any directed tree decomposition of* D*. Analogously, the* ***D-width*** D-width(D), ***DAG-width*** dag-width(D), ***Kelly-width*** Kelly-width(D) *and the* ***directed path-width*** $\mathrm{dpw}(D)$ *are defined as the minimum width of the corresponding decomposition of* D.

A class $\mathcal{C}$ *of digraphs has bounded directed tree-width if there is a constant* $c \geq 0$ *such that* $\mathrm{dtw}(D) \leq c$ *for every* $D \in \mathcal{C}$*. Classes of bounded width for other width measures are defined analogously.*

Digraphs with no directed cycles longer than a fixed constant form an example of a class of digraphs with bounded DAG-width, Kelly-width and directed tree-width. This follows from the following results by Bang-Jensen and Christiansen, respectively, Kintali.

Theorem 9.2.10 *[7] For every natural number* p*, every* D *digraph having no directed cycle of length more than* p *has DAG-width at most* p *and this is best possible.*

Theorem 9.2.11 *[63] For every natural number p, every D digraph having no directed cycle of length more than p has directed tree-width and Kelly-width at most $p + 1$.*

We close this section by mentioning two other digraph width measures which do not fall naturally within the framework of abstract decompositions. The first is the **DAG-depth**, defined in [41]. To define it, we need the concept of reachability component. Let D be a digraph. For $v \in V(D)$ we define $\text{Reach}_D(v) := \{u \in V(D) : u \text{ is reachable from } v \text{ by a directed path in } D\}$. A **reachability component** is a subgraph of D induced by an inclusion-wise maximal non-empty set in $\{\text{Reach}_D(v) : v \in V(D)\}$, i.e. an inclusion-wise maximal induced subgraph with only one initial strong component (see Section 1.5 for the definition of an initial component).

Definition 9.2.12 (DAG-depth) *Let D be a digraph. The **DAG-depth** $\text{dag-depth}(D)$ of D is inductively defined as follows: if $|V(D)| = 0$, then $\text{dag-depth}(D) = 0$. If D has a single reachability component, then we let $\text{dag-depth}(D) = 1 + \min\{\text{dag-depth}(D-v) : v \in V(D)\}$. Otherwise, if $D_1, \dots, D_c$ are the reachability components of D for some $C > 1$, then $\text{dag-depth}(D) := \max\{\text{dag-depth}(D_i) : 1 \leq i \leq c\}$.*

There are various other width measures for digraphs that have been defined in the literature, for instance **oriented tree-width**, **Kenny-width**, **entanglement**, **cycle rank** and others, see e.g. [3, 13, 14, 41, 55].

9.2.4 Alternative Characterizations of Digraph Width Measures

In the previous section we have defined several width measures for directed graphs based on variations of digraph decompositions. Many of these measures can also be defined equivalently and the equivalent definitions yield additional insights and intuition about the corresponding width measures.

All width measures defined above can be characterized by graph searching games. We have already covered the basics of graph searching games in Section 9.2.1. For digraphs, two main variants of games have emerged, depending on the ability of the robber to move. Let (X, v) be the current position in a graph searching game on a digraph D. Suppose the cops announce to move from X to X'. In the **strong reachability game**, the robber can choose any new position v' within the strongly connected component of $D - (X \cap X')$ that contains v. In the **weak reachability game**, the robber can choose any position v' that is reachable from v in $D - (X \cap X')$. Combining this distinction with the distinction between a visible and an invisible robber yields a range of graph searching games on directed graphs that can be used to give game based characterizations of the width measures introduced above.

Theorem 9.2.13 *Let D be a digraph and $k \in \mathbb{N}$.*

1. *If* $\mathrm{dtw}(D) \leq k$, *then k cops have a robber monotone winning strategy in the visible strong cops and robber game on D. Conversely, if k cops have a winning strategy in this game on D, robber-monotone or not, then* $\mathrm{dtw}(D) \leq 3k+2$. *If k cops have a winning strategy in the visible strong cops and robber game on D, then $3k+2$ cops have a robber-monotone winning strategy on D [1, 54].*
2. *D has DAG-width $\leq k$ if, and only if, k cops have a cop-monotone winning strategy on D if, and only if, k cops have a robber-monotone winning strategy on D in the visible weak reachability game [12].*
3. *D has Kelly-width $\leq k$ if, and only if, k cops have robber-monotone winning strategy on D in the invisible inert weak reachability game. Here, in the inert game variant the robber can only move when the cop player announces to place a cop on the current robber position [51].*
4. *D has directed path-width k if, and only if, k cops have a cop-monotone winning strategy on D if, and only if, k cops have robber-monotone winning strategy on D in the invisible weak reachability game [8].*
5. *D has DAG-depth $\leq k$ if, and only if, the cop player has a winning strategy with at most k cops in the visible weak reachability game in which he never moves any cop, i.e. in every round the cop player has to use new cops [41].*

Part (1) of the previous theorem follows from the observation that any directed tree decomposition of a digraph of width k yields a winning strategy for $k+1$ cops. Part (2) – (3), on the other hand, follow from Theorem 9.3.8 below, as a haven of order k yields a winning strategy for the robber against fewer than k cops. See below for details.

We close this section by giving an alternative characterization of Kelly-width in terms of elimination ordering and partial k-DAGs.

Definition 9.2.14 (Directed elimination ordering [51]**)** *An **elimination order** $\sqsubseteq$ for a digraph D is a linear order on $V(D)$. For a vertex v define $V_{v\sqsubseteq} := \{u \in V : v \sqsubseteq u\}$. The **support** of a vertex v with respect to $\sqsubseteq$ is*

$$\mathrm{supp}_{\sqsubseteq}(v) := \{u \in V_{v\sqsubseteq} : \text{there is } v' \in \mathrm{Reach}_{G-V_{v\sqsubseteq}}(v) \text{ with } (v',u) \in E\}\,.$$

*The **width** of an elimination order $\sqsubseteq$ is* $\max_{v\in V} |\mathrm{supp}_{\sqsubseteq}(v)|$.

The name elimination ordering originates in the following equivalent way of defining the width of an elimination ordering based on an explicit elimination process. Let D be a digraph and let $\sqsubseteq$ be a linear order on $V(D)$. Let $(v_0, v_1, \ldots, v_{n-1})$ be the enumeration of $V(D)$ with respect to $\sqsubseteq$. We define $G_0^{\sqsubseteq} := G$ and $G_{i+1}^{\sqsubseteq}$ as the graph obtained from $G_i^{\sqsubseteq}$ by deleting v_i and adding (if necessary) new arcs (u,v) if $(u,v_i),(v_i,v) \in E(G_i^{\sqsubseteq})$ and $u \neq v$. $G_i^{\sqsubseteq}$ is the **directed elimination graph at step i with respect to $\sqsubseteq$**.

Now it is readily verified that the width of the elimination order $\sqsubseteq$ is the maximum over all i of the out-degree of v_i in $G_i^{\sqsubseteq}$.

Definition 9.2.15 ((Partial) k-DAG [51]**)** *The class of k-DAGs is defined recursively as follows:*

- *A complete digraph with k vertices is a k-DAG.*
- *A k-DAG with $n+1$ vertices can be constructed from a k-DAG H with n vertices by adding a vertex v and arcs satisfying the following:*
  - *there are at most k arcs from v to H and*
  - *if X is the set of endpoints of the arcs added in the previous subcondition, then there is an arc from $u \in V(H)$ to v if $(u, w) \in E(H)$ for all $w \in X \setminus \{u\}$. Note that if $X = \emptyset$, this condition is true for all $u \in V(H)$.*

*A **partial** k-**DAG** is a subgraph of a k-DAG.*

Theorem 9.2.16 ([51]) *Let G be a digraph. The following are equivalent:*

1. *G has Kelly-width at most $k+1$.*
2. *G has a directed elimination ordering of width $\leq k$.*
3. *$k+1$ cops have a robber-monotone winning strategy to capture an inert invisible robber.*
4. *G is a partial k-DAG.*

Further characterizations of classes of digraphs of bounded width have been given in terms of forbidden subgraphs and forbidden minors, e.g. in [65], where Kintali and Zhang characterized partial 1-DAGs in terms of forbidden directed minors. See also [33, 73].

9.2.5 Comparing Directed Width Measures

In this section we compare the width measures introduced in the previous section with respect to generality. In particular, we are interested in the question whether classes of digraphs of bounded width with respect to one measure automatically have bounded width in another measure. As we will see, the width measures introduced above form the partial order shown in Figure 9.5.

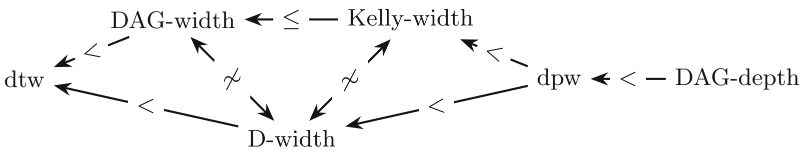

Figure 9.5 The relation between different measures. An arrow labelled by "<" means bounded only in one direction, an arrow labelled by "≤" means bounded at least in one direction. A bidirected arrow labelled "≮" means not bounded in any direction.

Lemma 9.2.17 ([12])

1. *Every class of digraphs of bounded DAG-width has bounded directed tree-width.*
2. *Conversely, there are classes of digraphs of bounded directed tree-width and unbounded DAG-width.*

Part 1 can easily be seen by considering the game characterization of DAG width and directed tree-width (see Theorem 9.2.13): the set of positions the robber can choose at any particular time in the directed tree-width game is a subset (proper or not) of the set of positions he can choose in the DAG width game. Hence, if k cops can catch the robber in the latter, they can also do so in the former.

Towards Part 2, let T_t be a complete directed binary tree of height t, i.e. a tree with all arcs oriented away from the root towards the leaves and every vertex has two or zero successors. Furthermore, every path from the root to a leaf has length t. Now add to T_t an arc from every vertex $v \in V(T_t)$ to every ancestor $u \in V(T_t)$ of v, i.e. to every $u \neq v \in V(T_t)$ on the unique path from the root r of T_t to v. We call this a **tree with back arcs**.

It is not hard to see that two cops can catch the robber on this tree for any value of t in the directed tree-width game: they just start with one cop at the root r. Then the robber has to decide into which of the two subtrees he wants to move. The cops can then put the second cop on the root of this subtree, i.e. on the successor v of r which is the root of the subtree containing the robber. If the robber is on this vertex v, he can only move further down into the subtree, i.e. into a subtree rooted at a successor v' of v. Once the cop on v is in place, the first cop on the root can be lifted and moved to v'. The cops continue in this way chasing the robber down. This is possible because once a cop is on v, every path that starts at the subtree of v containing the robber and which ends in this subtree but has an inner vertex outside of this tree has to go through v. Hence, even with only one cop on v the robber can no longer leave the subtree rooted at v.

In the DAG-width game, however, the robber can simply follow a directed path. In this game, to chase the robber down the tree the cops need to occupy the entire path from the root of T_t to the root of the current subtree containing the robber. This results in a strategy using t cops. With a little extra work one can show that there is no other, substantially better strategy. Hence, the DAG width of T_t is proportional to t. See [12] for details.

The next result we state is that the DAG-width of a digraph is bounded by a function of its Kelly-width. The question whether DAG-width and Kelly-width of a class of digraphs are mutually bounded is equivalent to the question whether the monotone cop numbers of the DAG-width and Kelly-width game on digraphs are bounded by each other. This is a long open problem in the theory of graph searching games. A partial answer was finally given by Rabinovich [3, 83] who introduced the concept of weak monotonicity in the DAG-width game and proved that every strategy for k cops in the Kelly-width game can be translated into a weakly monotone strategy for k cops in the DAG-width game. Furthermore, any winning strategy for k cops in the weakly monotone game can be translated into a monotone strategy for k^2 cops in the DAG width game. This implies the following lemma.

Lemma 9.2.18 *Every class $\mathcal{C}$ of digraphs with bounded Kelly-width has bounded DAG-width.*

The converse of the lemma is still open and it is related to one of the biggest open problems in graph searching, namely whether the monotonicity costs for Kelly- and DAG-width games are bounded, i.e. if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every digraph D, if k cops have a winning strategy on D in the Kelly-game then they also have a robber-monotone winning strategy on D using at most $f(k)$ cops (likewise for DAG-width games).

The next result we mention relates directed path-width to Kelly-width. Again it follows immediately from the game characterizations of the width measures that Kelly-width is more general than directed path width. Towards the converse, it can again be shown through the game connection that if $\mathcal{C}$ is a class of bidirected digraphs, i.e. digraphs where for every arc (u, v) also the reverse arc (v, u) is present, the DAG-width, Kelly-width, directed tree-width and D width all coincide with the undirected tree-width of the class $\mathcal{C}'$ of graphs obtained from $\mathcal{C}$ by replacing every directed arc by an undirected arc (removing duplicates). Furthermore, the directed path width of $\mathcal{C}$ equals the path-width of $\mathcal{C}'$ and the DAG-depth equals the tree-depth. As, for instance, the class of trees has unbounded path width but bounded tree-width, the next lemma follows.

Lemma 9.2.19 *Every class of digraphs of bounded directed path-width has bounded Kelly-width. Conversely, there are classes of digraphs of bounded Kelly-width but unbounded directed path-width.*

Finally, we compare D-width to the other classes. When D-width was introduced, it was conjectured to be equivalent to directed tree-width in the sense that classes of digraphs have bounded D-width if, and only if, they have bounded directed tree-width [90, Page 756]. The observation that bounded

D-width implies bounded directed tree-width is easily seen. One of the main differences between directed tree decompositions and D-decompositions is the concept of guarding. As the example in Figure 9.1 b) and the discussion in the paragraphs following the example show, the guard of an arc e can contain vertices which are contained in bags of an entirely different part of the tree decompositions. Also, along a branch of the directed tree decomposition, vertices can appear in a guard, then disappear from the guards and then reappear later. This leads to strategies for the cop-player which are not cop-monotone. This "external" guarding as well as the non-monotonicity is not possible in D-decompositions. Amiri et al. [3] manage to exploit these differences to exhibit classes of digraphs of bounded directed tree-width but where an unbounded number of cops is needed for the cop-monotone visible strong cops and robber game. This already implies that the D-width is also unbounded. They also exhibited classes of digraphs where a bounded number of cops have cop-monotone winning strategies but where the D-width is still unbounded.

Lemma 9.2.20 *([3][90])*

1. *Every class of digraphs of bounded D-width has bounded directed tree-width.*
2. *Conversely, there are classes of digraphs of bounded directed tree-width with unbounded D-width.*

In [3], D-width is shown to be incomparable to Kelly and DAG-width.

Lemma 9.2.21

1. *There are classes of digraphs of bounded D-width and unbounded Kelly- and DAG-width.*
2. *There are classes of digraphs of bounded DAG-width unbounded D-width.*
3. *Every class of digraphs of bounded DAG-depth has bounded directed path-width but the converse is false.*

Finally, it can again be shown using the game characterization that classes of digraphs of bounded directed path-width have bounded D-width and also bounded Kelly-width. The converse fails in both cases as explained above: the class of trees has bounded tree-width but unbounded path-width in the undirected case and replacing in trees arcs by two directed arcs in opposite directions separates directed path-width from D- and Kelly-width.

To separate directed path-width from DAG-depth note that the class of directed paths has directed path-width 2 but unbounded DAG-depth. On the other hand, one can show that if a digraph D has no path longer than t, then this implies that $t + 1$ cops can win the invisible cops and robber game on D and hence, by the game characterization of directed path-width in [8] and [50], the directed path-width is also at most $t + 1$.

9.3 Structure Theory for Directed Graphs Based on Directed Minors

Originally, Robertson and Seymour introduced the tree-width of undirected graphs as part of their monumental graph minor project culminating in the proof of Wagner's conjecture. At the heart of this project is a very powerful structure theorem explaining what can be said about a graph G knowing that it does not contain a fixed graph H as a minor. One simple reason for this could be that the tree width of G is too small. But G may fail to contain H as a minor even if the tree-width is very high. Therefore the major part of the graph minors project deals with graphs of very high tree-width that do not contain a fixed H as a minor. For this, one needs to understand what information can be gained about a graph knowing that its tree width is very high. The most fundamental result in this context is the **excluded grid theorem** in [89] stating that any graph of sufficiently high tree-width contains a large grid as a minor. Once this grid is found one can then analyze how the rest of the graph attaches to this grid which eventually leads to the local structure theorem and furthermore to the full structure theorem mentioned before.

With the introduction of directed tree-width, Reed, Robertson, Seymour and Thomas initiated the programme of generalizing this structure theory from undirected graphs to digraphs. Again, a major challenge is to understand what information can be obtained about a digraph knowing that its directed tree-width is very high, i.e. what can we say about **obstructions** to small directed tree-width. Consequently, the main open conjecture in the initial papers is the directed analogue of the excluded grid theorem, which, however, was only proved more than a decade after directed tree-width was introduced. In this section we present several powerful duality results between directed tree-width and various forms of obstructions. These results are not only interesting from a structural perspective but have found important algorithmic applications. We will comment on these applications in Section 9.4 below.

We begin by establishing a few fundamental properties of directed tree decompositions. Let (T, β, γ) be a directed tree decomposition of a digraph D.

The next lemma follows easily from the definition of directed tree decompositions and establishes a connection between decompositions and strong separators, i.e. sets of vertices separating strongly connected components into smaller components.

Lemma 9.3.1 *Let $\mathcal{T} := (T, \beta, \gamma)$ be a directed tree decomposition of a digraph D.*

1. *For every $e \in E(T)$, $\gamma(e)$ is a strong separator in D, i.e. if S_1, S_2 are the two components of $T - e$, then every strong component of $D - \gamma(e)$ is either contained in $\beta(S_1)$ or $\beta(S_2)$.*

2. *If $t \in V(T)$ and $T_1, \ldots, T_s$ are the components of $T - t$, then every strong component of $D - \Gamma(t)$ is contained in exactly one $\beta(T_i)$ for some i.*

Definition 9.3.2 *Let D be a digraph and $W \subseteq V(D)$.*

1. *A **balanced** W-**separator** is a set $S \subseteq V(D)$ such that every strong component of $D - S$ contains at most $\frac{|W|}{2}$ vertices of W. The **order** of the separator is $|S|$.*
2. *The set $W \subseteq V(D)$ is k-**linked** if D does not contain a balanced W-separator of order k.*

We show first that in a digraph of directed tree-width at most $k-1$ every set has a balanced separator of order k, i.e. D does not contain a k-linked set.

Lemma 9.3.3 *Let D be a digraph of directed tree-width at most $k - 1$. Then every set $W \subseteq V(D)$ has a balanced W-separator of order at most k.*

We sketch the proof of the lemma. See [70, 84] for details. Let (T, β, γ) be a directed tree decomposition of D of order k. For every arc $e = (u, v) \in A(T)$ let $C_1, \ldots, C_l$ be the strong components of $D - \gamma(e)$ containing an element of W. If none of the C_i contains more than $\frac{1}{2}|W|$ elements of W, then $\gamma(e)$ is a balanced W-separator and we are done. Otherwise, by Lemma 9.3.1, one of the two components T_u, T_v of $T - e$ contains the (unique) component C_i containing more than half of the elements of W. We orient e towards u if C_i is contained in $\beta(T_u)$ and towards v otherwise. This defines an orientation of T and as T is a tree there must be a vertex $t \in V(T)$ such that all incident arcs point towards t. It is easily seen that $\Gamma(t)$ is a balanced W-separator.

The next theorem establishes an even more precise relation between k-linked sets and the directed tree-width.

Theorem 9.3.4 ([54]) *Every digraph D either has directed tree-width at most $3k + 2$ or contains a set W which is k-linked and is a witness that D has directed tree-width at least k.*

We give the proof of this theorem as it will be the basis of an FPT algorithm[2] for computing, for a given digraph D a directed tree decomposition whose width is an approximation of the directed tree-width of D. in Section 9.4.

[2] By an **FPT algorithm** we mean an algorithm with running time $f(k) \cdot n^c$, for some function f and constant c, where n is the input size and k is a parameter defined in the definition of the problem the algorithm solves. See Section 1.11 for details.

Proof. To prove the theorem we inductively construct a directed tree decomposition (T, β, γ) of D. We maintain the property that for every inner vertex $t \in V(T)$, $|\Gamma(t)| \leq 3k + 2$ and for every arc $e \in E(T)$, $|\gamma(e)| \leq 2k + 1$.

Either this process will succeed and therefore produce a directed tree decomposition of the required width or it will fail at some point at which we obtain a k-linked set.

We initialize the construction by the trivial directed tree decomposition $\mathcal{T} := \big(T, \beta, \gamma\big)$, where T is the tree with one node r and $\beta(r) := V(D)$. Clearly this satisfies the invariant above.

Now suppose $\mathcal{T} = (T, \beta, \gamma)$ has already been constructed. If $\mathcal{T}$ does not contain a leaf $t \in V(T)$ with $|\Gamma(t)| > 3k+2$, then we are done. So let $t \in V(T)$ be such a leaf.

Let e be the arc incident with t. By construction, $|\gamma(e)| \leq 2k + 1$. If $\gamma(e)$ is k-linked, we are done. Otherwise, let S be a balanced $\gamma(e)$-separator of order at most k. Let $v \in \beta(t)$ be an arbitrary vertex and let $X := S \cup \{v\}$. By construction, $|X| \leq k + 1$, $X \cap \beta(t) \neq \emptyset$ and every strong component C of $D - X$ contains at most $\frac{1}{2}|\gamma(e)| \leq k$ elements of $\gamma(e)$. Let $C_1, \dots, C_s$ be the strong components of $D - (X \cup \gamma(e))$. By the definition of a directed tree decomposition, either $V(C_i) \subseteq \beta(t)$ or $V(C_i) \cap \beta(t) = \emptyset$, for all $1 \leq i \leq s$. Let $D_1, \dots, D_l$ be the components among $\{C_1, \dots, C_s\}$ with $V(C_i) \subseteq \beta(t)$. For each D_i, let D'_i be the component of $D - X$, such that $V(D_i) \subseteq V(D'_i)$ and let $W_i = \big(\gamma(e) \cap V(D'_i)\big) \cup X$. Then

$$|W_i| \leq |\gamma(e) \cap V(D'_i)| \cup |X| \leq k + k + 1 = 2k + 1$$

and D_i is also a strong component of $D - W_i$.

We extend $\mathcal{T}$ as follows to obtain a new decomposition $\mathcal{T}' := (T', \beta', \gamma')$: add new vertices $t_1, \dots, t_l$ and arcs $e_i := (t, t_i)$ to T, for all $1 \leq i \leq l$, and set $\beta'(t) := X \cap \beta(t)$, $\beta'(t_i) := V(D_i)$ and $\gamma'(e_i) := W_i$. For all other nodes t and arcs e we set $\beta'(t) := \beta(t)$ and $\gamma'(e) := \gamma(e)$. It is easily seen that $\mathcal{T}'$ is a directed tree-decomposition of D maintaining the invariant above. In particular, $|\beta'(t)| \leq |X| \leq k + 1$ and $|\gamma'(e_i)| \leq 2k + 1$. Furthermore, $\gamma'(e_i) \subseteq X \cup \gamma(e)$ and thus $\Gamma'(t) = \beta'(t) \cup \gamma'(e) \cup \bigcup\{\gamma'(e_i) : 1 \leq i \leq s\} \subseteq X \cup \gamma(e)$. It follows that $|\Gamma'(t)| \leq k + 1 + 2k + 1 = 3k + 2$. Furthermore, as $D_1, \dots, D_l$ are strong components of $D - (X \cup \gamma(e))$, the conditions of directed tree decompositions are still satisfied. □

A consequence of the proof of the previous lemma is that if a digraph D has directed tree-width at most k then it also has a directed tree decomposition of width at most $3k + 2$ which has a particularly nice form.

Definition 9.3.5 (Nice Directed Tree Decomposition) *Let D be a digraph. A directed tree decomposition (T, β, γ) of D is* ***nice*** *if*

a) for all $e = (s, t) \in A(T)$ the set $\beta(T_t)$ is a strong component of $G - \gamma(e)$ and

b) if $t \in V(T)$ and $s_1, \dots, s_l$ are the children of t in T, then $\bigcup_{1 \leq i \leq l} \beta(s_i) \cap \bigcup_{e \sim t} \gamma(e) = \emptyset$.

Nice decompositions are easier to work with in algorithmic applications and we will use them in the applications in Section 9.4. One immediate consequence of this definition is the following lemma which is algorithmically useful.

Lemma 9.3.6 *Let (T, β, γ) be a directed tree-decomposition of a digraph D. For every $t \in V(T)$ there is an ordering $<_t$ on the successors $s_1, \ldots, s_k$ of t in T so that if $s_i <_t s_j$, then D does not contain any arc $e = (u, v) \in E(D)$ with $u \in \beta(T_{s_i})$ and $v \in \beta(T_{s_j})$.*

For now we go back to the study of obstructions for directed tree width. We have already seen that a k-linked set is an obstruction to small directed tree-width. The next obstruction we study are known as **havens**. In the sequel, for any set X and $k \geq 0$, we denote the set of all subsets of X of order less than k by $[X]^{<k}$.

Definition 9.3.7 *Let D be a digraph. A* ***haven*** *of D of* ***order*** *k is a function $h : [V(D)]^{<k} \to 2^{V(G)}$ assigning to every set X of fewer than k vertices a strong component of $G - X$ such that if $Y \subseteq X \subseteq V(D)$ with $|X| < k$, then $h(X) \subseteq h(Y)$.*

It is easily seen that any k-linked set W in a digraph D defines a haven of order k: for every set $X \subseteq V(D)$ of order at most k define $h(X)$ as the (unique) strong component of $D - X$ containing more than half of the elements of W. It is straightforward to verify that this satisfies the haven axioms. Hence, we obtain the following theorem.

Theorem 9.3.8 ([54])

1. *If G is a digraph of tree-width at most k, then G does not contain a haven of order k.*
2. *Conversely, if G does not contain a haven of order k, then G has tree-width at most $3k + 2$.*

We now define a sequence of other obstructions for directed tree-width, originally defined in [84].

Definition 9.3.9 *A* ***bramble*** *in a digraph D is a set $\mathcal{B}$ of strongly connected subgraphs of D such that for any pair $B, B' \in \mathcal{B}$, either $V(B) \cap V(B') \neq \emptyset$ or there are arcs e, e' linking B and B' in both directions. A bramble $\mathcal{B}$ is* ***strong*** *if $V(B) \cap V(B') \neq \emptyset$ for all $B, B' \in \mathcal{B}$.*

A ***cover****, or* ***hitting set****, of $\mathcal{B}$ is a set $X \subseteq V(D)$ such that $X \cap V(B) \neq \emptyset$ for all $B \in \mathcal{B}$. The* ***order*** *of $\mathcal{B}$ is the minimum size of a cover of $\mathcal{B}$.*

The last type of obstruction we consider are **well-linked sets**.

Definition 9.3.10 *Let D be a digraph. A set $W \subseteq V(D)$ is **well-linked** if for any $X, Y \subseteq W$ with $|X| = |Y|$ there are $|X| = |Y|$ pairwise vertex disjoint paths from X to Y in $G - (W \setminus (X \cup Y))$.*

The next lemma, proved in [84], connects the various forms of obstructions we have seen so far. See [70] for details.

Lemma 9.3.11 *Let D be a digraph and let $k \geq 0$.*

1. *If D contains a k-linked set, then it contains a strong bramble of order $k+1$.*
2. *If D contains a bramble $\mathcal{B}$ of order k, then D contains a well-linked set of order k.*
3. *If D contains a well-linked set of order $4k+1$, then D contains a k-linked set.*

Proof. To show Part (1), it is not hard to see that a k-linked set W in a digraph D defines a bramble of order $k+1$: for every set $X \subseteq V(D)$ of at most k vertices add to the bramble to the unique strong component of $D - X$ containing more than half of the vertices of W. It is readily verified that this indeed yields a strong bramble.

For (2) one can show that every minimum size cover of a bramble must be well-linked.

Part (3) is slightly more technical and we refer, e.g. to [70] for details. □

As explained at the beginning of this section, one of the most fundamental theorems in Robertson and Seymour's graph minor project is the excluded grid theorem. In the mid-90s, Reed [85] and Johnson et al. [54] conjectured an analogous theorem for directed graphs, i.e. that any digraph of sufficiently high directed tree-width should contain a large cylindrical grid as a **butterfly minor**.

Definition 9.3.12 (Butterfly minor) *Let D be a digraph and let $e = (u, v) \in A(D)$. The digraph D/e obtained from D by **contracting** e is defined as the digraph with vertex set $V(D) \setminus \{u, v\} \cup \{x_{u,v}\}$, where $x_{u,v}$ is a fresh vertex. The edges of D/e are the same as the edges of D except for the edges with u or v as endpoint. Any such edge (w, w') or (w', w), where $w \in \{u, v\}$ and $w' \notin \{u, v\}$ is replaced by an edge $(x_{u,v}, w')$ and $(w', x_{u,v})$ resp.*

*A **butterfly contraction** is the operation of contracting an edge $e = (u, v)$ where u has out-degree 1 or v has in-degree 1. A digraph H is said to be a **butterfly minor** of a digraph D, written $H \preceq^b D$, if it can be obtained from a subgraph of D by a series of butterfly contractions.*

Definition 9.3.13 (cylindrical grid) *A **cylindrical grid** of order k, for some $k \geq 1$, is a digraph G_k consisting of k directed cycles $C_1, \ldots, C_k$, pair-*

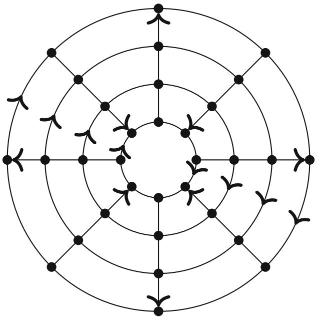

Figure 9.6 Cylindrical grid G_4.

wise vertex disjoint, together with a set of $2k$ *pairwise vertex disjoint paths* $P_1, \ldots, P_{2k}$ *such that*

- *each path* P_i *has exactly one vertex in common with each cycle* C_j*,*
- *the paths* $P_1, \ldots, P_{2k}$ *appear on each* C_i *in this order*
- *for odd* i *the cycles* $C_1, \ldots, C_k$ *occur on all* P_i *in this order and for even* i *they occur in reverse order* $C_k, \ldots, C_1$*.*

See Figure 9.6 for an illustration of G_4. The conjecture by Reed, Johnson, Robertson, Seymour and Thomas was confirmed by Kawarabayashi and Kreutzer in [61].

Theorem 9.3.14 (The directed grid theorem [61]**)** *There is a function* $f : \mathbb{N} \to \mathbb{N}$ *such that every digraph of directed tree-width at least* $f(k)$ *contains a cylindrical grid of order* k *as a butterfly minor.*

9.4 Complexity of Directed Width Measures and Algorithmic Applications

In this section we describe some of the algorithmic applications of directed width measures. In particular, we will see that some $\mathcal{NP}$-complete graph problems can be solved efficiently on classes of digraphs of bounded width. As these applications usually require the computation of the associated decompositions, we first consider the complexity of computing digraph decompositions in the next section. The main algorithmic applications are presented in Section 9.5 below.

9.4.1 Complexity of Directed Width Measures

We first show that for essentially all width measures defined above, the associated decision problem is $\mathcal{NP}$-hard. This follows from the following observation.

Theorem 9.4.1 *Let G be an undirected graph and let D be the digraph obtained from G by replacing each arc $\{u, v\}$ by two arcs (u, v) and (v, u)*[3]*. Then* $\mathrm{tw}(G) + 1 = \mathrm{dtw}(D) + 1 = \text{dag-width}(D) = \text{Kelly-width}(D) = \text{D-width}(D)$, *where* $\mathrm{tw}(G)$ *denotes the tree-width of G.*

Furthermore, the tree-depth of G equals the DAG-depth of D and the path-width of G equals the directed path-width of D minus 1.

The case for directed tree-width was proved in [54, (2.1)]. The equalities for DAG-width and Kelly-width follow immediately from the corresponding game characterizations. For directed path-width and D-width there are direct translations of the corresponding decompositions and for DAG-depth it follows immediately from the definition of DAG depth and tree-depth.

Deciding the tree-width, the tree-depth and the path-width of a graph G is $\mathcal{NP}$-complete (see e.g. [5]) and hence the decision problems for the directed width measures is $\mathcal{NP}$-hard. For all width measures except DAG-width, the decomposition defining the width are of polynomial size in the size of the input graph and hence the problems are even $\mathcal{NP}$-complete. For DAG-width this is not the case, as we shall see below.

Corollary 9.4.2 *Deciding the DAG-depth, the directed tree-width, the directed path-width, the D-width and the Kelly-width of a digraph is $\mathcal{NP}$-complete. Deciding the DAG-width of a digraph is $\mathcal{NP}$-hard.*

Right from the definition, the number of bags in a DAG decomposition of a digraph D is not restricted to be polynomial in the size of the decomposed digraph. And in fact, it was shown in [3], that there are classes of digraphs where DAG decompositions of optimal width require super-polynomially many bags, i.e. there is no fixed degree polynomial bounding the number of bags of a DAG-decomposition in the number of vertices of the digraph. In particular, this rules out that optimal DAG-decompositions can be computed by an FPT algorithm parameterized by the DAG-width. To make matters worse, it was also shown in [3], that there is no polynomial size approximation of an optimal DAG decomposition with an additive constant error in the width. Furthermore, the problem of deciding the DAG-width of a digraph turned out to be much harder than deciding any of the other width measures.

Theorem 9.4.3 ([3]) *The problem, given a digraph G and a number $k \geq 0$, whether the DAG-width of G is at most k, is* PSPACE*-complete.*

9.4.2 Computing Directed Graph Decompositions

We have seen that deciding directed width measures is computationally hard. However, a range of algorithms have appeared for computing decompositions

[3] Thus D is the complete biorientation of G.

which approximate the optimal width. Here we present some of these approximation algorithms.

Directed Tree-Width. The first algorithm we present below is an FPT approximation algorithm[4] for directed tree-width that follows from [54]. See Section 1.11 for details on parameterized algorithms. The proof of Theorem 9.3.4 showing the duality between havens and directed tree-width can easily be made algorithmic using the notion of weakly balanced separations. Recall the definition of a **nice** directed tree decomposition (Definition 9.3.5).

Theorem 9.4.4 *There is an algorithm with running time* $\mathcal{O}(3^{2k+2} \cdot k \cdot |A(D)| \cdot |V(D)|)$ *which, on input* D *and* $k \geq 1$*, either computes a nice directed tree-decomposition of* D *of width at most* $5k + 10$ *or a weakly* k*-linked set* W*.*

Sketch. Essentially, the proof of Theorem 9.3.4 already yields an algorithm for computing directed tree decompositions. The only problem is that balanced separators cannot be computed efficiently. However, in the proof balanced W-separators can be replaced by weakly balanced W-separations. Here, a **weakly balanced** W**-separation** is a triple (X, S, Y) of pairwise disjoint sets $X, Y \subseteq W$ of order $0 < |X|, |Y| \leq \frac{3}{4}|W|$ and $S \subseteq V(D)$ such that $W = X \cup (S \cap W) \cup Y$ and there is no directed path from X to Y in $D - S$. The **order** of the separation is $|S|$.

Adapting the algorithm in [34, Corollary 11.22] to the directed setting one can show that there is an algorithm running in time $\mathcal{O}(3^{2k+2}k|A(D)|)$ which, given as input a digraph D, a number $k \geq 1$ and a set $W \subseteq V(D)$ of size $2k + 2$, computes a weakly balanced W-separation of order at most k if such a separation exists.

Using weakly balanced separations instead of balanced separators in the proof of Theorem 9.3.4 yields an algorithm with the running time as stated in the theorem, at the expense of increasing the width of the constructed directed tree decomposition to $(4k + 1) + (k + 1) = 5k + 2$. ☐

The previous algorithm yields a fixed-parameter approximation algorithm for directed tree-width. Kintali, Kothari and Kumar designed a polynomial time approximation algorithm of directed tree-width up to $\log n$-factors.

Theorem 9.4.5 ([64]) *There exists a polynomial time approximation algorithm that, given a digraph* D*, computes a directed tree decomposition of* D*, whose width is at most* $\mathcal{O}(\log^{\frac{3}{2}} |V(D)| \cdot \mathrm{dtw}(D))$*.*

DAG-Width and Kelly-Width. To date, directed tree-width is the only tree-width inspired width measure which can be computed (approximately)

[4] By an FPT approximation algorithm we mean an algorithm running in time $f(k) \cdot n^c$, for some function f and a constant c, which given a number k and a digraph D computes a directed tree decomposition of D of width $O(k)$ or determines that the directed tree-width of D is $> k$.

by FPT algorithms on general digraphs. For DAG-width an XP-algorithm is known for computing an optimal width decomposition.

Theorem 9.4.6 ([12]) *Given a digraph D of DAG-width at most k, a DAG decomposition of D of width at most k can be computed in time $|D|^{\mathcal{O}(k)}$.*

For Kelly-width it is still open whether optimal decompositions can be computed by an XP-algorithm. The reason is that DAG-width is defined by a cops and robber game with a visible robber, i.e. a game of perfect information. Kelly-width, on the other hand, is defined by an invisible robber game and hence by a game with imperfect information, which are computationally harder. Hence, the game characterization does not immediately yield an XP-algorithm. However, there are explicit algorithms known for computing Kelly decompositions.

Theorem 9.4.7 ([51]) *The Kelly-width of a digraph with n vertices can be determined in time $\mathcal{O}^*(2^n)$ and space $\mathcal{O}^*(2^n)$, or in time $\mathcal{O}^*(4^n)$ and polynomial space, where $\mathcal{O}^*(f(n))$ means that polynomial factors are suppressed.*

Furthermore, the Kelly-width of a digraph can be approximated up to a $\log n$ factor.

Theorem 9.4.8 ([64]) *There exists a polynomial time approximation algorithm that, given a digraph D, computes a Kelly decomposition of D, whose width is $\mathcal{O}(\log^{\frac{3}{2}} n \cdot \text{Kelly-width}(D))$.*

Finally, for small values of k, efficient and explicit algorithms for deciding the Kelly-width and computing corresponding decompositions were given, e.g. in [73].

Directed Path-Width. The situation for directed path-width is similar to the case of Kelly width.

Theorem 9.4.9 ([66, 97])

1. *There is an algorithm which, given a digraph D and $k \in \mathbb{N}$ as input, computes a directed path decomposition of D of width k, if it exists, in time $\mathcal{O}(|D|^{k+1} \cdot |A(D)|)$.*
2. *There is an algorithm computing a directed path-decomposition of a digraph D of optimal width in time $\mathcal{O}^*(1.89^n)$, where $\mathcal{O}^*$ means that polynomial factors are suppressed.*

It is still open whether computing optimal directed path decompositions is fixed-parameter tractable. However, Fomin and Pilipczuk [37] exhibited FPT algorithms for computing optimal width path decompositions on tournaments and semi-complete digraphs.

9.5 Applications of Tree-Width Inspired Directed Width Measures

On classes of undirected graphs of bounded tree width $\mathcal{NP}$-hard problems from a very broad spectrum of areas and types of problems have been shown to become efficiently solvable, often even in linear time for any fixed upper bound on the tree-width. In particular, Courcelle [23] proved that every problem definable in **monadic second-order logic (MSO$_2$)** can be solved in linear time on bounded tree-width classes (of undirected graphs). **Monadic second-order logic(MSO$_2$)** is a logic extending plain first-order logic by quantification over sets of edges and sets of vertices of a graph. It is very powerful logical language in which many graph problems such as 3-Colourability, Hamiltonian paths and -cycles, k-disjoint paths, perfect matchings and many more can be expressed very naturally. See [24] for details on monadic second-order logic and its variants MSO$_2$ and MSO$_1$ used below.

For directed graphs, Ganian et al. [43] showed that no such broad MSO$_2$ based algorithm theory is possible for tree width inspired width measures. Essentially, under some technical conditions, they showed that if one wants tractability of all MSO definable problems on classes of bounded width with respect to some width measure that translates undirected tree-width to digraphs (defined as having a graph searching game characterization similar to tree-width), then the only width achieving this undirected tree width. This establishes a general limit of tractability for digraph width measures based on tree-width but allows for algorithmic applications more specific to directed graphs.

Directed width measures, especially directed tree-width, have found various applications in the design of algorithms: in database theory, Bagan et al. [6] proved that **simple regular path queries** can be evaluated in polynomial time on graph databases of bounded directed tree-width (whereas the problem is intractable in general). In the area of Boolean networks, Tamaki [96] conducted experiments on computing attractors in Boolean networks. It turned out that for networks of small directed path-width he was able to handle networks which were significantly larger than what can be handled by standard tools. Another example motivated by practical applications is given in [94], where Sheppard investigates digraphs obtained from DNA sequencing by hybridization. In this method a digraph is constructed where vertices correspond to so-called **k-mers**. An important algorithmic problem in this context is finding Hamiltonian paths. It was shown in [94] that the digraphs occurring in this context usually have very small DAG-width so that polynomial time algorithms for computing Hamiltonian paths on digraphs of small DAG-width (see below) become applicable. In general, the most intensively studied applications of directed width measures are for routing problems in directed graphs. We present some of these applications in the following sections.

9.5.1 Disjoint Paths and Linkage Problems in Digraphs of Bounded Width

One of the main applications of directed width measures is to routing problems in digraphs. We will see various examples where directed tree-width is used in algorithms for solving various forms of directed disjoint paths problems. In particular, we will show that problems such as the directed Hamiltonian path problem or the k-disjoint paths problem can be solved in polynomial time on classes of digraphs of bounded directed tree-width.

> k-DISJOINT PATHS
> **Input:** A digraph G and terminals $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$
> **Question:** Does D have k pairwise internally vertex disjoint paths $P_1, \ldots, P_k$ such that P_i is from s_i to t_i for $i = 1, \ldots, k$?

The k-disjoint paths problem on directed and undirected graphs is well-known to be $\mathcal{NP}$-complete. But whereas on undirected graphs, the problem is fixed-parameter tractable, it is $\mathcal{NP}$-complete on directed graphs even for $k = 2$, as shown by Fortune, Hopcroft and Wyllie [39]. See Section 1.6.

Theorem 9.5.1 ([39]) *The k-DISJOINT PATHS problem is $\mathcal{NP}$-complete for all $k \geq 2$.*

Furthermore, as shown by Slivkins [95], the k-DISJOINT PATHS problem is already $W[1]$-hard on DAGs. But Johnson, Robertson, Seymour and Thomas [54] proved that it can be solved in polynomial time for every fixed value of k on any fixed class $\mathcal{C}$ of digraphs of bounded directed tree-width.

Definition 9.5.2 *A **linkage** in a digraph D is a set $\mathcal{L}$ of pairwise internally vertex disjoint directed paths. The **order** $|\mathcal{L}|$ is the number of paths in $\mathcal{L}$ and its **size** is $|V(\mathcal{L})|$, where $V(\mathcal{L}) := |\bigcup_{P \in \mathcal{L}} V(P)|$.*

*Let $\sigma := \{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of k pairs of vertices in D. A **σ-linkage** is a linkage $\mathcal{L} := \{P_1, \ldots, P_k\}$ of order k such that P_i links s_i to t_i.*

The first algorithmic result we establish is the following theorem.

Theorem 9.5.3 ([54])*Let D be a digraph and (T, β, γ) be a directed tree decomposition of D of width w. Let $k, l \geq 1$ and let σ be a set of k pairs of vertices in D. It can be decided in time $|V(D)|^{\mathcal{O}(k+w)}$ whether D contains a σ-linkage of size l.*

Problem 9.5.4 *Can the previous theorem be improved to fixed-parameter tractability in the directed tree-width, for any fixed number k? I.e. does there exist for every fixed k an algorithm running in time $f(\mathrm{dtw}(G)) \cdot |V(G)|^c$, for some constant c and function f, both depending on k, that decides whether G has a σ-linkage for any set σ of at most k source/terminal pairs?*

The theorem can also be extended to weighted digraphs, see e.g. [70] for details. Combined with the algorithm for computing directed tree decompositions in Theorem 9.4.4, the theorem immediately implies the following corollary.

Corollary 9.5.5 *The Hamiltonian cycle, the Hamiltonian path and, for all k, the k-*Disjoint paths *problem can be solved in polynomial time on any class $\mathcal{C}$ of digraphs of bounded directed tree-width.*

We sketch the proof of Theorem 9.5.3. Recall that for any digraph D and $S \subseteq V(D)$, $D[S] := (S, A(D) \cap S \times S)$ denotes the subdigraph of D induced by S. Similarly, if $\mathcal{L}$ is a linkage in D and $S' \subseteq V(D)$, we write $\mathcal{L}[S']$ for the **projection** of $\mathcal{L}$ onto $D[S']$, i.e. the linkage $\{P \cap D[S'] : P \in \mathcal{L}\}$. The algorithm is based on the following observation. Let D be a digraph and let $S \subseteq V(D)$ be a set of vertices. For $k \geq 0$ we say that S is **k-protected** if there is a strong guard $Z \subseteq V(D)$ of S of order $|Z| \leq k$. Note that if (T, β, γ) is a directed tree decomposition of a digraph D of width $k-1$ and $t \in V(D)$, then $\beta(T_t) = \bigcup\{\beta(t') :: t' \text{ is reachable from } t \text{ in } T\}$ is k-protected. In particular, if $e = (s,t)$ is an arc in $E(T)$ then we can take $Z := \gamma(e)$ as a witness for $\beta(T_t)$ being k-protected. The main observation is now the following.

Lemma 9.5.6 *Let D be a digraph and let $S \subseteq V(D)$. Let $k, w \geq 0$ and let $\mathcal{L}$ be a linkage of order k in $D[S]$.*

If $S' \subseteq S$ is w-protected, then $\mathcal{L}[S']$ has order at most $k + w$.

Proof. Let $P_1, \cdots, P_k$ be the paths in $\mathcal{L}$ and let $Z \subseteq V(D)$ be such that $|Z| \leq w$ and every directed path in D starting and ending in S' which is not entirely contained in $D[S']$ contains a vertex of Z. It follows that if $P_i[S']$ is the union of j directed paths, then $|V(P_i) \cap Z| \geq j - 1$. Hence, $\mathcal{L}[S']$ has order at most $k + w$. □

The previous lemma is the basis for a dynamic programming algorithm for solving the linkage problem in Theorem 9.5.3. Given a digraph D and a directed tree decomposition (T, β, γ) of D of width $w-1$, the algorithm proceeds as follows. For every $t \in V(T)$ and every tuple $\sigma := \big((u_1, v_1), \ldots, (u_s, v_s)\big)$ of pairs of vertices in $\beta(T_t)$, for some $s \leq k + w$, it computes the set of all $l \leq |V(D)|$ such that $G[\beta(T_t)]$ contains a σ-linkage of size l. As shown in [54], this can be done by dynamic programming. Clearly, once this information is computed for the root of T, the linkage problem for D can be answered for every tuple $\sigma = \big((u_1, v_1), \ldots, (u_k, v_k)\big)$ of order k. This completes the sketch of the proof of Theorem 9.5.3.

In the terminology of parameterized complexity, see Section 1.11, the previous result shows that the k-disjoint paths problem is in XP with parameter $k + w$, where w is the directed tree-width of the input digraph. Unless FPT=$W[1]$, this cannot be improved to fixed-parameter tractability (FPT) in the parameter k for every fixed width w, as Slivkins [95] showed that the disjoint paths problem is $W[1]$-hard already on DAGs, which have directed tree-width 0.

We close this section by mentioning an algorithmic meta theorem by Oliveira Oliveira generalizing the previous linkage algorithm.

Theorem 9.5.7 ([29]) *Let Ω be a finite commutative semigroup. Let φ be an MSO_2 sentence and let $k, w \in \mathbb{N}$. There is a computable function $f : \mathbb{N}^3 \to \mathbb{N}$ such that, given a weighted digraph $D = (V, E, \omega : E(D) \to \Omega)$ of directed tree-width w, a positive integer $l < |V|$ and an element $\alpha \in \Omega$, one can count in time $f(\varphi, w, k) \cdot |D|^{O(k\cdot(w+1))}$ the number of subgraphs H of D simultaneously satisfying the following four properties:*

1. *$H \models \varphi$.*
2. *H is the union of k directed paths.*
3. *H has l vertices.*
4. *H has weight $\omega(H) = \alpha$.*

In fact, one can even choose a semigroup of size polynomial in D.

9.5.2 Linkages in General Digraphs

The results in the previous section exhibit algorithms for linkage type problems on digraphs of small directed tree-width. However, the machinery of directed tree decompositions and obstructions to low directed tree-width can also be used to obtain results for general digraphs.

Given the $\mathcal{NP}$-hardness of the k-Disjoint paths problem already for $k = 2$, it is natural to consider relaxations of the problem in order to obtain polynomial time algorithms. One relaxation that has been studied in the literature is to allow congestions. Let $\sigma := \big((s_1, t_1), \ldots, (s_k, t_k)\big)$ be a k-tuple of pairs of vertices in a digraph D and let $c \geq 1$. A set $P_1, \ldots, P_k$ of directed paths in D is a **σ-linkage with congestion** c if, for all $1 \leq i \leq k$, the path P_i links s_i to t_i and furthermore, every vertex of D is contained in at most c paths. For $c = 2$ we call the linkage **half-integral** and for $c = 4$ it is a **quarter-integral** linkage.

Problem 9.5.8 *Does there exist, for every fixed integer $k \geq 1$, a polynomial algorithm which, given a digraph D and a tuple $\sigma := \big((s_1, t_1), \ldots, (s_k, t_k)\big)$ as input, decides correctly whether D contains a half-integral σ-linkage.*

However, partial results are known. In [60], Kawarabayashi, Kobayashi and Kreutzer show the following result for quarter-integral linkages.

Theorem 9.5.9 ([60]) *For every fixed $k \geq 1$ there is a polynomial time algorithm for deciding the following problem.*

Quarter-integral disjoint paths
Input: *A digraph D and terminals $s_1, t_1, s_2, t_2, \ldots, s_k, t_k \in V(D)$.*
Find: *a quarter-integral linkage of $(s_1, t_1), \ldots, (s_k, t_k)$ or conclude that D does not contain disjoint paths $P_1, \ldots, P_k$ such that P_i is from s_i to t_i, for $i \in [k]$.*

The proof of the previous theorem in [60] precedes the proof of the directed excluded grid theorem (Theorem 9.3.14). Using Theorem 9.3.14, the result can be improved to third-integral linkages. The main idea of the proof is to use the duality between directed tree width and cylindrical grids. Roughly, the algorithm works as follows. If the directed tree-width is small then it uses a simple adaptation of the algorithm in Theorem 9.5.3 to solve the problem optimally. Otherwise, Theorem 9.3.14 implies that D contains a large cylindrical grid C. If there is a linkage L_1 from $s_1, \ldots, s_k$ to C and a linkage L_2 from C to $t_1, \ldots, t_k$ then L_1, C and L_2 can be used to construct a third-integral linkage linking s_i to t_i, for all $1 \leq i \leq k$. Otherwise, by Menger's theorem, there must be a low order separation from, say, $s_1, \ldots, s_k$. The separation does not rule out the existence of a quarter-integral solution but it can sometimes be used to rule out a fully integral solution (which would then be the second outcome of the theorem). If a fully integral solution is not ruled out by this construction, then the problem can be reduced to a smaller instance. In this way, one either gets a third-integral solution or the algorithm certifies that there are no fully disjoint paths linking the sources to the targets.

As mentioned above, it is still an open problem whether the result can be improved to half-integral solutions and, more importantly, whether it can further be improved so that the negative answer also rules out the existence of a half-integral solution.

As a first significant step in this direction, Edwards, Muzi and Wollan proved a polynomial time algorithm for the half-integral linkage problem for highly connected digraphs.

Theorem 9.5.10 ([31]) *For all integers $k \geq 1$, there exists a value $L(k)$ such that every strongly $L(k)$-connected graph is half-integrally k-linked. Moreover, there exists an absolute constant c such that given an instance $(D, (s_1, t_1), \ldots, (s_k, t_k))$ of the half-integral disjoint path problem, where D is $L(k)$-connected, we can find a solution in time $\mathcal{O}(|V(D)|^c)$.*

We close the section by mentioning further applications of these techniques beyond classes of digraphs of small directed tree-width. Fomin and Pilipczuk [37] showed that for tournaments the k-arc disjoint paths problem fixed-parameter tractable. Their algorithm uses directed path-width. They first show that on tournaments directed path-width can be decided by an FPT algorithm. They then use a duality of directed path-width and an obstruction called jungles which was proved in [20, 40].

Finally, the concepts of directed tree-width, or more specifically, its dual notion of well-linked sets, have played a decisive role in the study of approximation algorithms for symmetric routing on planar digraphs. See Section 5.2.

9.5.3 The Erdős-Pósa Property for Directed Graphs

A classical result by Erdős and Pósa states that there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every k, every graph G contains k pairwise vertex disjoint cycles or a set T of at most $f(k)$ vertices such that $G - T$ is acyclic.

There is a natural generalization of this result to arbitrary graphs: a graph H has the Erdős-Pósa property if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that every graph G either has k disjoint copies of H as a minor or contains a set T of at most $f(k)$ vertices such that H is not a minor of $G - T$. As an application of the undirected excluded grid theorem, Robertson and Seymour [89] proved that a graph H has the Erdős-Pósa-property in this sense if, and only if, H is planar.

The Erdős-Pósa property can also be defined for digraphs. Younger [101] conjectured that there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every k every digraph either has k disjoint directed cycles or a set of at most $f(k)$ vertices intersecting every directed cycle. The conjecture was proved by Reed, Robertson, Seymour and Thomas in [86]. In fact, the concept of directed tree-width originated in the work on Younger's conjecture.

Again this can be generalised to arbitrary digraphs, based on directed minors (see Section 9.6.1 for the definition of butterfly and topological minors): a digraph H has the **Erdős-Pósa property** for topological (butterfly) minors if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $k \geq 0$, every digraph D either contains k disjoint subgraphs each containing H as a topological (butterfly) minor or there is a set $S \subseteq V(D)$ of at most $f(k)$ vertices such that $D - S$ does not contain H as a topological (butterfly) minor. In [4], Amiri, Kawayabashi, Kreutzer and Wollan used the directed excluded grid theorem (Theorem 9.3.14) to show the following characterization of strongly connected digraphs with the Erdős-Pósa property.

Theorem 9.5.11 *Let H be a strongly connected digraph.*

1. *H has the Erdős-Pósa property for butterfly minors if, and only if, there is a cylindrical grid G_c, for some constant $c = c(H)$, such that $H \preceq^b G_c$.*
2. *H has the Erdős-Pósa property for topological minors if, and only if, there is a cylindrical wall G_c, for some constant $c = c(H)$, such that $H \preceq^t G_c$.*

Furthermore, for every fixed strongly connected digraph H satisfying these conditions and every k there is a polynomial time algorithm which, given a digraph G as input, either computes k disjoint (butterfly or topological) models of H in G or a set S of $\leq h(k)$ vertices such that $G - S$ does not contain a model of H.

The previous theorem settles the case for strongly connected digraphs. It would be interesting to get a similar characterization also for general digraphs. This may be much harder to get as in this case the techniques based on directed tree-width will no longer be as useful as for strongly connected digraphs. An intermediate case could be **vertex cyclic** digraphs which are digraphs in which every strong component is non-trivial, i.e. contains more than a single vertex. In [4], some special cases of vertex-cyclic digraphs are solved, but the general problem remains open.

Problem 9.5.12

1. *Characterize the class of vertex-cyclic digraphs which have the Erdős-Pósa property.*
2. *Characterize in general the class of digraphs which have the Erdős-Pósa property.*
3. *What is the complexity of deciding, given a digraph H, whether it has the Erdős-Pósa property?*

9.6 Density Based Width Measures

In this section we introduce the second type of directed width measures covered in this chapter: width measures based on directed minors and density arguments. For this, we first need to define the notions of directed minors used in this section.

9.6.1 Directed Minors

On undirected graphs, one usually distinguishes between two types of minors: **topological minors**, obtained by subdividing edges and deleting edges or vertices, and general **minors**, obtained by a sequence of edge and vertex deletion and arc contraction.

Topological minors have a straight forward generalization to directed graphs.

Definition 9.6.1 *A* ***subdivision*** *of a digraph D is obtained by replacing some arcs of D by pairwise internally vertex disjoint directed paths respecting the directions of the replaced arcs. For $r \geq 0$, H is an* r***-subdivision*** *of D if we can replace some arcs of H by paths of length at most $r+1$ to obtain D.*

For digraphs D, H, we say that H is a ***directed topological minor*** *of D, denoted by $H \preceq^t D$, if D contains a subdivision of H as a subgraph. We write $H \preceq^t_r D$ and call H an* r***-shallow topological minor*** *of D if D contains a $2r$-subdivision of H as a subgraph.*

The reason we define r-shallow topological minors as $2r$-subdivisions is that this corresponds more closely to r-shallow directed minors defined below.

For general directed minors, several alternative and not necessarily equivalent definitions have been considered in the literature. The most popular among these are **butterfly minors**, defined in Definition 9.3.12 above.

For undirected graphs, the notion of minors that are obtained by a series of vertex and edge deletions and edge contractions can equivalently be defined in terms of minor models. In the directed setting these two notions are different (every butterfly minor is also a directed minor but not vice versa) [72].

Definition 9.6.2 *A digraph H has a* ***directed model*** *in a digraph D if there is a function δ mapping vertices $v \in V(H)$ of H to sub-graphs $\delta(v) \subseteq D$ and arcs $e \in E(H)$ to arcs $\delta(e) \in E(D)$ such that if $v \neq u$ then $\delta(v) \cap \delta(u) = \emptyset$ and if $e = (u, v)$ and $\delta(e) = (u', v')$ then $u' \in \delta(u)$ and $v' \in \delta(v)$.*

For $v \in V(H)$ let $\text{in}(\delta(v)) := V(\delta(v)) \cap \bigcup_{e=(u,v)\in E(H)} V(\delta(e))$ and $\text{out}(\delta(v)) := V(\delta(v)) \cap \bigcup_{e=(v,w)\in E(H)} V(\delta(e))$.

Furthermore, we require that for every $v \in V(H)$

1. *there is a directed path in $\delta(v)$ from any $u \in \text{in}(\delta(v))$ to every $u' \in \text{out}(\delta(v))$;*
2. *there is at least one source vertex $s_v \in \delta(v)$ that reaches every element of $\text{out}(\delta(v))$;*
3. *there is at least one sink vertex $t_v \in \delta(v)$ that can be reached from every element of $\text{in}(\delta(v))$.*

We write $H \preceq^d D$ if H has a directed model in D and call H a ***directed minor*** *of D. We call the sets $\delta(v)$ for $v \in V(H)$ the* ***branch-sets*** *of the model.*

Note that the conditions (2) and (3) in the previous definition are implies by Condition (1) for vertices of in- and out-degree > 0. They serve the purpose to ensure that sinks and sources in H are represented by a single vertex in D together with paths connecting this vertex to its (in- or out-) neighbours.

Definition 9.6.3 *For $r \geq 0$, a digraph H is a directed* ***depth-r minor*** *of a digraph D, denoted as $H \preceq^d_r D$, if there exists a directed model of H in D in which the length of all the paths in the branch-sets of the model are bounded by r.*

We close the section by relating the different concepts of minors to each other. It is not hard to see that for all digraphs H, D,

$$H \preceq^t D \quad \Rightarrow \quad H \preceq^b D \quad \Rightarrow \quad H \preceq^d D.$$

The same relation extends to shallow minors:

Lemma 9.6.4 *For all digraphs H, D and $r \geq 0$: $H \preceq^t_r D$ implies $H \preceq^d_r D$.*

Bipartite digraphs will play a special role on the rest of this section.

Definition 9.6.5 *A **bipartite digraph** is a digraph $D = (A\dot{\cup}B, E)$ whose vertex set is partitioned into two sets A and B and $E \subseteq A \times B$.*

For bipartite digraphs, the concepts of butterfly minors and directed models coincide. In the following lemma, an **in-branching** is a digraph obtained from an undirected tree by orienting all edges towards a root node r. Analogously, in an **out-branching** all arcs are oriented away from the root, i.e. and out-branching is a rooted directed tree. See Section 1.8.

Lemma 9.6.6 (see [72]) *If H is a bipartite digraph with $H \preceq^d D$, we can choose the branch-sets of the model of H in D to be in- or out-branchings. In this case $H \preceq^d D \Leftrightarrow H \preceq^b D$.*

9.6.2 Width Measures Defined by Shallow Directed Minors and Bounded Edge Densities

Following [71, 72] (see [74, 75] for the undirected case), we define classes of digraphs of bounded expansion, nowhere crownful classes and classes which are nowhere dense. We first need some additional notation.

Definition 9.6.7 *Let G be a digraph and let $r \geq 0$. The **greatest reduced average degree of rank** r (short **grad**) of G, denoted $\nabla_r(G)$ is*

$$\nabla_r(G) := \max\left\{\frac{|E(H)|}{|V(H)|} : H \preceq^d_r G\right\}$$

*and its **topological greatest average degree of rank** r (short **top-grad**) is*

$$\tilde{\nabla}_r(G) := \max\left\{\frac{|E(H)|}{|V(H)|} : H \preceq^t_r G\right\}.$$

A **crown of order** q is a digraph S_q with vertex set $\{v_i : 1 \leq i \leq q\} \cup \{v_{i,j} : 1 \leq i < j \leq q\}$ and arcs $\{(v_{i,j}, v_i), (v_{i,j}, v_j) : 1 \leq i < j \leq q\}$.

Definition 9.6.8 *Let $\mathcal{C}$ be a class of digraphs.*

1. *$\mathcal{C}$ has **bounded expansion** if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\nabla_r(D) \leq f(r)$ for all $r \geq 0$ and $D \in \mathcal{C}$.*
2. *$\mathcal{C}$ is **nowhere crownful** if for every r, there exists a $q = q(r)$ so that $S_q \not\preceq^d_r D$ for all all $D \in \mathcal{C}$.*
3. *$\mathcal{C}$ is **directed nowhere dense** if for every r, there exists an n and an acyclic tournament T_n so that $T_n \not\preceq^d_r D$ for all $D \in \mathcal{C}$.*
4. *$\mathcal{C}$ is **directed somewhere dense** if there is an $r \geq 0$ so that the set of depth r minors of $\mathcal{C}$ contains arbitrarily large tournaments.*

It can be shown that a class $\mathcal{C}$ of digraphs is directed somewhere dense if, and only if, it is not directed nowhere dense. Furthermore, the property of being directed nowhere dense is more general than being nowhere crownful and also more general than bounded expansion.

On the other hand, classes of digraphs of bounded expansion and nowhere crownful classes are incomparable. In particular, as shown in [72], nowhere crownful classes and even crown-minor free classes can be very dense.

Theorem 9.6.9 *For every ϵ, there exists a $q = q(\epsilon)$, such that for every n, there exists an S_q-minor-free digraph on $2n$ vertices that has arc density at least $\Omega(n^{\frac{1}{2}-\epsilon})$.*

It follows that there are classes of digraphs which are S_q-crown-minor free but do not have bounded expansion. Conversely, the class of crowns S_q, $q \geq 0$, has bounded expansion.

On the other hand, for the definition of bounded expansion, the precise notion of directed minor we use is not important, as shown by the following theorem proved in [71].

Theorem 9.6.10 *A class $\mathcal{C}$ of digraphs has bounded expansion if and only if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$ it holds that $\widetilde{\nabla}_r(D) \leq f(r)$ for all $D \in \mathcal{C}$.*

9.7 Classes of Directed Bounded Expansion

Classes of digraphs of bounded expansion can be characterized in many different ways. The various characterizations yield a varied set of algorithmic techniques that can be used in the design of algorithms on bounded expansion classes of digraphs. In the following we will present some of the more promising structural properties of bounded expansion classes.

9.7.1 Generalised Colouring Numbers

The **colouring number** $\mathrm{col}(G)$ of an undirected graph G is the smallest integer k such that there is a linear order $\sqsubset$ on the vertex set of D for which each vertex v has **back-degree** at most $k-1$, i.e. at most $k-1$ neighbours u with $u \sqsubset v$. It is well-known that for any graph G, the chromatic number $\chi(G)$ satisfies $\chi(G) \leq \mathrm{col}(G)$.

Three natural generalization of the colouring number are the series adm_r, col_r and wcol_r of **generalised colouring numbers** defining the **admissibility, colouring number** and **weak colouring numbers** introduced by Kierstead and Yang [62] (see Dvořák [30] for the general definition of adm_r) in the context of colouring games and marking games on graphs. Note that the colouring number is equivalent to the **degeneracy** of a graph. As proved by Zhu [102], these invariants can be used to characterize bounded expansion classes of undirected graphs.

The directed versions of the above invariants have been defined in [71] where it was shown that classes of directed bounded expansion can be characterized by bounds on the generalised colouring numbers.

Let D be a digraph. By $\Pi(D)$ we denote the set of all strict linear orders on $V(D)$. For $\sqsubset \in \Pi(G)$, we write $u \sqsubseteq v$ if $u \sqsubset v$ or $u = v$. Let $u, v \in V(D)$, let $\sqsubset \in \Pi(D)$ and let $r \geq 0$.

The vertex u is **weakly** r**-reachable** from v with respect to $\sqsubset$, if there is a directed path P of length ℓ, $0 \leq \ell \leq r$, connecting u and v (in either direction) such that u is the smallest among the vertices of P (with respect to $\sqsubset$). By $\mathrm{WReach}_r[D, \sqsubset, v]$ we denote the set of vertices that are weakly r-reachable from v w.r.t. $\sqsubset$.

The vertex u is **strongly** r**-reachable** from v with respect to $\sqsubset$, if there is a directed path P of length ℓ, $0 \leq \ell \leq r$, connecting u and v (in either direction) such that $u \sqsubseteq v$ and $v \sqsubset w$ for all internal vertices w of P. Let $\mathrm{SReach}_r[D, \sqsubset, v]$ be the set of vertices that are strongly r-reachable from v w.r.t. $\sqsubset$. Note that we have $v \in \mathrm{SReach}_r[D, \sqsubset, v] \subseteq \mathrm{WReach}_r[D, \sqsubset, v]$.

We also need a third type of colouring number, the admissibility. For a non-negative integer r, the r**-admissibility** $\mathrm{adm}_r[D, \sqsubset, v]$ of v w.r.t. a linear order $\sqsubset \in \Pi(D)$ is the maximum size k of a family $\{P_1, \ldots, P_k\}$ of directed paths of length at most r with one end v and the other end at a vertex w with $w \sqsubset v$, and which satisfies $V(P_i) \cap V(P_j) = \{v\}$ for all $1 \leq i < j \leq k$. As for $r > 0$ we can always let the paths end in the first vertex smaller than v, we can assume that the internal vertices of the paths are larger than v.

Definition 9.7.1 ([71]) *Let D be a digraph. For a non-negative integer r, we define the **weak r-colouring number** $\mathrm{wcol}_r(D)$, the **r-colouring number** $\mathrm{col}_r(D)$ and the **r-admissibility** of D as*

$$\begin{aligned}
\mathrm{wcol}_r(D) &:= \min_{\sqsubset \in \Pi(D)} \max_{v \in V(D)} \big|\mathrm{WReach}_r[D, \sqsubset, v]\big|, \\
\mathrm{col}_r(D) &:= \min_{\sqsubset \in \Pi(D)} \max_{v \in V(D)} \big|\mathrm{SReach}_r[D, \sqsubset, v]\big|. \\
\mathrm{adm}_r(D) &:= \min_{\sqsubset \in \Pi(D)} \max_{v \in V(D)} \mathrm{adm}_r[D, \sqsubset, v].
\end{aligned}$$

The following theorem relates these measures to each other.

Theorem 9.7.2 ([71]) *Let D be a digraph and let $r \geq 1$. Then $\mathrm{adm}_r(D) \leq \mathrm{col}_r(D) \leq \mathrm{wcol}_r(D)$. Furthermore,*

$$\mathrm{col}_r(D) \leq 2 \cdot (\mathrm{adm}_r(D) - 1)^r + 1 \quad \textit{and} \quad \mathrm{wcol}_r(D) \leq 2 \cdot \mathrm{adm}_r(D)^r.$$

The generalised colouring numbers can also be used to characterize bounded expansion classes of digraphs.

Theorem 9.7.3 ([71]) *For every digraph D and every $r \in \mathbb{N}$ it holds that* $\mathrm{adm}_r(D) < 6r^3 \nabla_r(D)^4$. *Conversely, for every digraph D and every $r \in \mathbb{N}$ it holds that* $\widetilde{\nabla}_r(D) \leq 16(\mathrm{adm}_{2r}(D) + 1)$.

Corollary 9.7.4 ([71]) *A class $\mathcal{C}$ of digraphs has bounded expansion if, and only if, there is a function $f : \mathbb{N} \to \mathbb{N}$ such that* $\mathrm{wcol}_r(D) \leq f(r)$ *for all $D \in \mathcal{C}$ and all $r \geq 1$.*

A useful property of admissibility is that for every graph D from a bounded expansion class $\mathcal{C}$ an order $\sqsubset$ on $V(D)$ witnessing that the r-admissibility is small can be computed efficiently.

Theorem 9.7.5 ([71]) *Let $\mathcal{C}$ be a class of digraphs of bounded expansion. There is a function g such that for all $r \geq 0$ and all $D \in \mathcal{C}$ we can compute an optimal order for* $\mathrm{adm}_r(D)$ *in time $g(r) \cdot n^{\mathcal{O}(1)}$, where $n := |V(D)|$.*

9.7.2 Neighbourhood Complexity

We continue the study of structural properties of bounded expansion classes by defining a directed version of neighbourhood complexity, a measure that has very successfully been used in the connection to classes of undirected bounded expansion [87].

Definition 9.7.6 *Let D be a digraph, let $X \subseteq V(D)$ and let $r \geq 1$. The* ***distance-r out-neighbourhood complexity of X in D****, denoted $\nu^+(D, X)$, is defined by*

$$\nu^+(D, X) = \max_{H \subseteq D, X \subseteq V(H)} \left|\{N_r^+(v) \cap X : v \in V(H)\}\right| .$$

Analogously, one can define the **distance-r in-neighbourhood complexity** when using $N_r^-(v)$ and the **distance-r mixed neighbourhood complexity** when using $(N_r^+(v) \cup N_r^-(v))$ in the above definition.

Closure under subgraphs in the above definition is required to characterize sparse graph classes. Classically, this closure is not part of the definition, when it is e.g. used to define classes of bounded VC-dimension [91, 93, 99].

Bounded neighbourhood complexity is not equivalent to directed bounded expansion but at least classes of directed bounded expansion have bounded neighbourhood complexity.

Theorem 9.7.7 ([71]) *Let $\mathcal{C}$ be a class of digraphs of bounded expansion. Then for all $r \geq 1$ there exists $k \geq 1$ such that for all $D \in \mathcal{C}$ and $X \subseteq V(D)$ we have $\nu_r^+(D, X) \leq |X|^k$. The same statement holds for in-neighbourhood complexity and mixed neighbourhood complexity.*

9.7.3 A Splitter Game for Classes of Digraphs of Bounded Expansion

In this section we establish a very useful property of bounded expansion classes of digraphs based on a directed version of a game, known as the **splitter game**, originally introduced as a characterization of nowhere dense classes of undirected graphs in [46].

We first need the following definition. The r**-strong-neighbourhood** of v, denoted by $\tilde{N}_{G,r}(v)$, or just $\tilde{N}_r(v)$ if G is understood, is defined as the set of vertices u in G such that G contains a closed walk of length at most $2r$ containing u and v.

Let G be a digraph and let $\ell, m, r \geq 0$. The (ℓ, m, r)**-strong directed splitter game** on G is played by two players, **Connector** and **Splitter**, as follows. Let $G_0 := G$. In round $i+1$ of the game, Connector picks a vertex $v_{i+1} \in V(G_i)$. Then Splitter chooses a subset $W_{i+1} \subseteq V(G_i)$ with $|W_{i+1}| \leq m$. Define G_{i+1} as the induced subgraph of G_i with $V(G_{i+1}) = \tilde{N}_{G_i,r}(v_{i+1}) \setminus W_{i+1}$. Splitter wins if $V(G_{i+1}) = \emptyset$. Otherwise the game continues to the next round. If Splitter has not won after ℓ rounds, then Connector wins.

A **strategy** for Splitter is a function f associating to every partial play $(v_1, W_1, \ldots, v_s, W_s)$ with associated sequence $G_0, \ldots, G_s$ and every move $v_{s+1} \in V(G_s)$ by Connector a move $W_{s+1} \subseteq V(G_s)$ with $|W_{s+1}| \leq m$ for Splitter. A strategy f is a **winning strategy** for Splitter if she wins every play in which she follows the strategy f. If such a winning strategy exists, we say that Splitter **wins** the (ℓ, m, r)-directed splitter game on G.

The splitter game cannot be used as a characterization of bounded expansion as Splitter wins the $(1, 1, 1)$-strong splitter game on every acyclic digraph, but the class of acyclic digraphs does not have bounded expansion. But on every class of bounded expansion Splitter always has constant length winning strategies. This, together with neighbourhood covers introduced in the following section, can be used to define a bounded depth decomposition of graph from bounded expansion classes.

Theorem 9.7.8 ([71]) *Let D be a graph, let $r \in \mathbb{N}$ and let $\ell = \mathrm{wcol}_{4r}(G)$. Then splitter wins the $(\ell, 1, r)$-strong splitter game.*

9.7.4 Neighbourhood Covers

Neighbourhood covers of small radius and small size play a key role in the design of many data structures for distributed systems. There is also a deep connection between sparse neighbourhood covers of small radius and sparse graph spanners of low stretch. In this section we will see that classes of digraphs of bounded expansion admit sparse strong neighbourhood covers which can be computed by a fixed-parameter algorithm.

Let $r \in \mathbb{N}$. A **strong r-neighbourhood cover** $\mathcal{X}$ of a digraph D is a mapping $\mathcal{X} : V(D) \to 2^{V(D)}$ such that $D[\mathcal{X}(v)]$ is strongly connected and $\tilde{N}_r(v) \subseteq \mathcal{X}(v)$. We call each $D[\mathcal{X}(v)]$ a **cluster** of $\mathcal{X}$.

The **radius** of a cluster $C := D[\mathcal{X}(v)]$ is defined as the minimal $r \in \mathbb{N}$ for which there is a vertex $w \in V(C)$ and for every $w \in V(C)$, the cluster C contains a directed path of length at most r from w to v and a directed path of length at most r from v to w. The **radius** $\mathrm{rad}(\mathcal{X})$ of a cover $\mathcal{X}$ is the maximum radius of any of its clusters.

The **degree** $d^{\mathcal{X}}(v)$ of v in $\mathcal{X}$ is the number of clusters that contain v. The **maximum degree** $\Delta(\mathcal{X})$ of $\mathcal{X}$ is $\Delta(\mathcal{X}) = \max_{v \in V(G)} d^{\mathcal{X}}(v)$.

Theorem 9.7.9 ([71]) *Let $\mathcal{C}$ be a class of digraphs of bounded expansion. There are functions $f, h : \mathbb{N} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$ and all graphs $D \in \mathcal{C}$, there exists a strong r-neighbourhood cover of radius at most $4r$ and maximum degree at most $f(r)$ and this cover can be computed in time $h(r) \cdot n^{\mathcal{O}(1)}$.*

9.7.5 Constant-Factor Approximation Algorithms for Strong Dominating Sets

In this section we give an algorithmic application of the bounded expansion classes in proving that strong dominating sets can be approximated up to a constant factor on any class $\mathcal{C}$ of directed bounded expansion.

Definition 9.7.10 (Strong r-Dominating Sets)

1. *Let $r \geq 1$ and let D be a digraph. A vertex $v \in V(D)$ **strongly-r-dominates** a vertex $u \in V(D)$ if there is a closed walk of length at most $2r$ in D containing u and v.*
2. *A **strong-r-dominating set** is a set $X \subseteq V(D)$ such that every vertex in D is strongly dominated by a vertex in X.*
3. *The **strong r-domination number** of D, denoted $sdom_r(D)$, is the minimum size of a strong r-dominating set of D.*

Note that if D is a digraph obtained from an undirected graph G by replacing every edge e in G by two arcs with the same endpoints but opposite orientation, then any strong-r-dominating set in D is an r-dominating set in G and vice versa. This explains the choice of the length $2r$ in Part (1) of the previous definition. It follows that deciding the strong-r-domination number of a digraph D is $\mathcal{NP}$-complete.

Theorem 9.7.11 *Let $\mathcal{C}$ be a class of digraphs of directed bounded expansion. Let $r \geq 1$. There is a polynomial time constant factor approximation algorithm for strong r-dominating sets. More precisely, for every value of r, there is an algorithm running in time $g(r) \cdot n^{\mathcal{O}(1)}$ for some function g which, on input $D \in \mathcal{C}$ computes a strong-r-dominating set $X \subseteq V(G)$ of order at most* $\mathrm{wcol}_{4r}(D)^2 \cdot \mathrm{sdom}_r(D)$.

The proof is based on computing a linear order witnessing that the $4r$-weak colouring number of the input digraph D is bounded. Following this order, a suitable greedy strategy can be shown to produce a strong r-dominating set of order $\mathrm{wcol}_{4r}(D)^2 \cdot k$ and an **obstruction** witnessing that there is no strong r-dominating set of order k. Hence, the approximation factor is $\mathrm{wcol}_{4r}(D)^2$, which is a constant on bounded expansion classes. Here, an r-**obstruction set** is a set $X \subseteq V(D)$ such that for any distinct $x, y \in X$, there are *no* two closed directed walks $W_1, W_2 \subseteq V(D)$, each of length at most $2r$, such that $W_1 \cap W_2 \neq \emptyset$ and $x \in W_1$ and $y \in W_2$.

As no two distinct vertices of an obstruction set lie on a closed walk of length at most $2r$, no two vertices from the set can be strongly r-dominated by a single vertex. Hence, if D contains an obstruction set of order k then D does not contain a strong r-dominating set of order $< |X|$.

A similar strategy was used by Dvořák in [30] to design a constant factor approximation algorithm for dominating sets on classes of undirected graphs of bounded expansion.

9.8 Nowhere Crownful Classes of Digraphs

We close our exposition of density and minor based width measures by giving another algorithmic application for dominating sets, this time on nowhere crownful classes. Towards this aim, we introduce the notion of directed uniformly quasi-wide classes and show that this concept yields an equivalent characterization of nowhere crownful classes of digraphs.

Definition 9.8.1 *Let D be a digraph and $d \in \mathbb{N} \cup \{0\}$. A set $U \subseteq V(D)$ is d-**scattered** if there is no $v \in V(D)$ and $u_1 \neq u_2 \in U$ such that v has distance at most d to both u and u'.*

Note that any subset of $V(D)$ is 0-scattered since v is the only vertex of distance zero from itself.

Definition 9.8.2 *A class $\mathcal{C}$ of digraphs is **uniformly quasi-wide** if there are functions $s : \mathbb{N} \to \mathbb{N}$ and $N : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for every $D \in \mathcal{C}$ and all $d, m \in \mathbb{N}$ and $W \subseteq V(D)$ with $|W| > N(d, m)$ there is a set $S \subseteq V(D)$ with $|S| \leq s(d)$ and $U \subseteq W$ with $|U| = m$ such that U is d-scattered in $G - S$. s, N are called the margin of $\mathcal{C}$.*

If s and N are computable then we call $\mathcal{C}$ **effectively uniformly quasi-wide**.

The next theorem was shown in [72].

Theorem 9.8.3 *A class $\mathcal{C}$ of digraphs is nowhere crownful if, and only if, it is directed uniformly quasi-wide.*

We demonstrate one algorithmic application of uniformly quasi-wideness by sketching the following theorem. A **directed dominating set** in a digraph D is a set $X \subseteq V(D)$ such that $N_D^+(X) \cup X = V(D)$.

Theorem 9.8.4 ([72]) *Let $\mathcal{C}$ be a class of digraphs which is nowhere crownful. Then the directed dominating set problem is fixed-parameter tractable on $\mathcal{C}$.*

Let $\mathcal{C}$ be nowhere crownful. Given a digraph $D \in \mathcal{C}$ and a number k, we compute a directed dominating set X of order k, if it exists, as follows. We let $W = V(D)$ be the set of vertices still to be dominated. As $\mathcal{C}$ is uniformly quasi-wide, if W is large enough, we can compute a constant-size set S of vertices and a 1-scattered set $A \subseteq W$ of order $k+1$ in $D - S$. As no vertex not in S can dominate two vertices in A, it follows that any set X of vertices dominating every vertex in W needs to contain a vertex in S. As S has constant size we can try each choice of a vertex $v \in S$ for the set X. For any such choice we recurse with the parameter $k-1$ and the set $W' := W \setminus N^+(v)$ of vertices we still need to dominate. This yields a natural recursion where in each recursion step the parameter is decreased. If at some point the set W is too small to contain a large 1-scattered set, then we can use brute force to compute a set of order k dominating W.

Similarly, one can show that on nowhere crownful classes of digraphs, the directed independent dominating set problem, the dominating out-branching problem and the independent set problem as well as their distance-d-versions are fixed-parameter tractable.

9.9 Rank-Width Inspired Width Measures

In this section, we introduce directed versions of clique-width and rank-width. The motivation of **clique-width** comes from the observation that many algorithmic problems are tractable on classes of graphs that can be recursively decomposable along vertex partitions (A, B) where the number of neighbourhood types between A and B is small. Different from tree-width based width measures, acyclic digraphs have arbitrary large directed clique-width, and clique-width separates the class of acyclic digraphs into easy and hard instances for some algorithmic problems.

When clique-width was first introduced, no FPT approximation algorithm for generating a clique-width expression was known. Oum and Seymour [82] first devised an FPT approximation algorithm for undirected clique-width, using an equivalent width parameter called **rank-width**. While clique-width expressions describe how to generate a graph using certain graph operations, rank-width decompositions generalize decomposition scheme called branch-decompositions [88]. Courcelle and Engelfriet [24] argued that directed clique-width can be approximated using undirected rank-width.

Bi-rank-width and **$\mathbb{F}_4$-rank-width** are two natural generalizations of rank-width for directed graphs, introduced by Kanté [56] and Kanté and Rao [59]. They can also be used to approximate directed clique-width. The other motivation of these parameters is on related graph containment relations **vertex-minor** and **pivot-minor**. Because clique-width and rank-width may increase by removing edges or contracting edges, these parameters are not well fit to minor structure theory. Instead, vertex-minor and pivot-minor relations have been studied together with rank-width [80, 81], and provide some structural results, sometimes generalising results on tree-width. Kanté and Rao [59] explained how to generalize these concepts to directed graphs, and generalised known results to directed graphs.

We present FPT approximation algorithms in Subsection 9.9.3. In Subsection 9.9.4, we present algorithmic applications of directed clique-width and bi-rank-width. We discuss structural results on these graph containment relations in Subsection 9.9.5.

9.9.1 Directed Clique-Width

Courcelle, Engelfriet and Rozenberg [25] introduced **clique-width** for both undirected graphs and directed graphs. For a digraph $D = (V, A)$ and a function $\mathrm{lab} : V \to \{1, 2, \ldots, k\}$, the triple (V, A, lab) is called a **k-labeled digraph**. The function lab is called a **labeling** of D, and for each $v \in V$, $\mathrm{lab}(v)$ is called its label.

Definition 9.9.1 (Directed clique-width) *For a positive integer k, the class dcw_k of k-labeled digraphs is recursively defined as follows.*

1. *The digraph on a single vertex v with label i in $\{1, 2, \ldots, k\}$ is in dcw_k. We denote by $\bullet_{i,v}$ the operation creating such a vertex.*
2. *Let $D_1 = (V_1, A_1, \mathrm{lab}_1) \in \mathrm{dcw}_k$ and $D_2 = (V_2, A_2, \mathrm{lab}_2) \in \mathrm{dcw}_k$ be two k-labeled digraphs on disjoint vertex sets. Let $D_1 \oplus D_2 := (V, A, \mathrm{lab})$ where $V := V_1 \cup V_2$, $A := A_1 \cup A_2$ and*
$$\mathrm{lab}(v) := \begin{cases} \mathrm{lab}_1(v) & \text{if } v \in V_1, \\ \mathrm{lab}_2(v) & \text{if } v \in V_2, \end{cases}$$
for every $v \in V$. We have $D_1 \oplus D_2 \in \mathrm{dcw}_k$.
3. *Let $D = (V, A, \mathrm{lab}) \in \mathrm{dcw}_k$ be a k-labeled digraph, and $i, j \in \{1, 2, \ldots, k\}$ be two distinct integers. Let $\rho_{i\to j}(D) := (V, A, \mathrm{lab}')$ where*
$$\mathrm{lab}'(v) := \begin{cases} \mathrm{lab}(v) & \text{if } \mathrm{lab}(v) \neq i, \\ j & \text{if } \mathrm{lab}(v) = i, \end{cases}$$
for every $v \in V$. We have $\rho_{i\to j}(D) \in \mathrm{dcw}_k$.
4. *Let $D = (V, A, \mathrm{lab}) \in \mathrm{dcw}_k$ be a k-labeled digraph, and $i, j \in \{1, 2, \ldots, k\}$ be two distinct integers. Let $\alpha_{i,j}(D)$ be the digraph obtained from D by adding all arcs (a, b) where $\mathrm{lab}(a) = i$ and $\mathrm{lab}(b) = j$. We have $\alpha_{i,j}(D) \in \mathrm{dcw}_k$.*

The ***directed clique-width*** *of a digraph* $D = (V, A)$*, denoted by* $\mathrm{dcw}(D)$*, is the minimum integer* k *such that there is a* k*-labeling* lab *of* D *where* $(V, A, \mathrm{lab}) \in \mathrm{dcw}_k$*.* ***Directed clique-width k-expressions*** *are expressions which recursively construct a graph with the four graph operations in 1-4.*

The difference between directed clique-width and undirected clique-width is on the function $\alpha_{i,j}$; for undirected clique-width, this function adds undirected edges between all pairs (v, w) where $\mathrm{lab}(v) = i$ and $\mathrm{lab}(w) = j$. We can naturally represent a directed clique-width k-expression as a tree-structure; an example is depicted in Figure 9.7. We call this tree a **directed clique-width** k**-expression tree**.

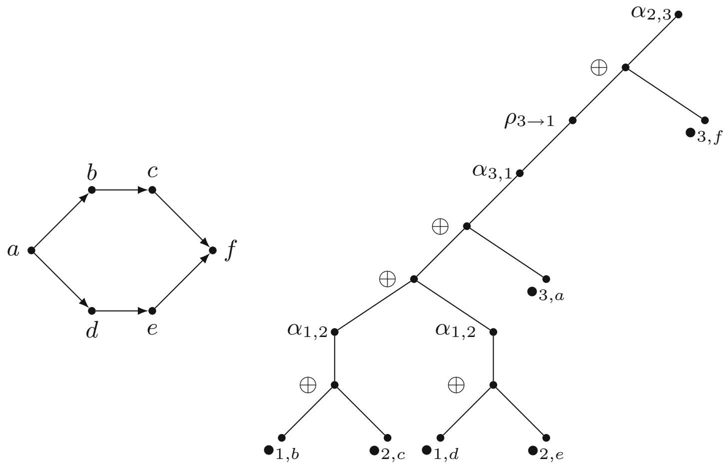

Figure 9.7 An example of a directed clique-width 3-expression tree, which expresses $\alpha_{2,3}((\rho_{3\to1}(\alpha_{3,1}((\alpha_{1,2}(\bullet_{1,b} \oplus \bullet_{2,c}) \oplus \alpha_{1,2}(\bullet_{1,d} \oplus \bullet_{2,e})) \oplus \bullet_{3,a}))) \oplus \bullet_{3,f})$.

Wanke [100] introduced a similar width parameter NLC-width. In NLC-width expressions, we add edges between two labeled graphs at once after taking disjoint union, while we add edges one by one between two vertex subsets with single labels in clique-width expressions. Gurski, Wanke and Yilmaz [47] generalised this parameter to directed graphs.

Definition 9.9.2 (Directed NLC-width) *For a positive integer* k*, the class* dNLC_k *of* k*-labeled digraphs is recursively defined as follows.*

1. *The digraph on a single vertex* v *with label* i *in* $\{1, 2, \dots, k\}$ *is in* dNLC_k*. We denote by* $\bullet_{i,v}$ *the operation creating such a vertex.*
2. *Let* $D_1 = (V_1, A_1, \mathrm{lab}_1) \in \mathrm{dNLC}_k$ *and* $D_2 = (V_2, A_2, \mathrm{lab}_2) \in \mathrm{dNLC}_k$ *be two* k*-labeled digraphs on disjoint vertex sets, and* $\overrightarrow{S}, \overleftarrow{S} \subseteq \{1, 2, \dots, k\} \times$

$\{1, 2, \ldots, k\}$ *be two relations. Let* $D_1 \times_{\overrightarrow{S}, \overleftarrow{S}} D_2 := (V, A, \mathrm{lab})$ *be the labeled graph where* $V := V_1 \cup V_2$, $A := A_1 \cup A_2 \cup \overrightarrow{A} \cup \overleftarrow{A}$ *with*

$$\overrightarrow{A} = \{(v, w) | v \in V_1, w \in V_2, (\mathrm{lab}_1(v), \mathrm{lab}_2(w)) \in \overrightarrow{S}\},$$
$$\overleftarrow{A} = \{(w, v) | v \in V_1, w \in V_2, (\mathrm{lab}_1(v), \mathrm{lab}_2(w)) \in \overleftarrow{S}\},$$

and

$$\mathrm{lab}(v) := \begin{cases} \mathrm{lab}_1(v) & \textit{if } v \in V_1, \\ \mathrm{lab}_2(v) & \textit{if } v \in V_2, \end{cases}$$

for every $v \in V$*. We have* $D_1 \times_{\overrightarrow{S}, \overleftarrow{S}} D_2 \in \mathrm{dNLC}_k$*.*

3. *Let* $D = (V, A, \mathrm{lab}) \in \mathrm{dNLC}_k$ *and* $R : \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$ *be a function. Let* $\circ_R(D) = (V, A, \mathrm{lab}')$ *be the labeled graph where* $\mathrm{lab}'(v) = R(\mathrm{lab}(v))$ *for every* $v \in V$*. We have* $\circ_R(D) \in \mathrm{dNLC}_k$*.*

The ***directed NLC-width*** *of a digraph* $D = (V, A)$*, denoted by* $\mathrm{dnlcw}(D)$*, is the minimum integer* k *such that there is a* k*-labeling* lab *of* D *where* $(V, A, \mathrm{lab}) \in \mathrm{dNLC}_k$*.* ***Directed NLC-width k-expressions*** *are expressions which recursively construct a graph with the three graph operations in 1-3.*

Gurski, Wanke and Yilmaz [47] derived a relationship between directed clique-width and directed NLC-width.

Theorem 9.9.3 ([47]) *For every digraph* D*, the parameters* $\mathrm{dcw}(D)$ *and* $\mathrm{dnlcw}(D)$ *are related as follows:* $\mathrm{dnlcw}(D) \le \mathrm{dcw}(D) \le 2\mathrm{dnlcw}(D)$*.*

One example of digraph classes having bounded directed clique-width is the class of **directed cographs**. This class is a directed variant of the class of undirected cographs. The term cograph stands for complement reducible graph [22], representing the property that the complement of a cograph is again a cograph. Directed cographs are graphs that can be recursively defined as follows:

1. Every single vertex is a directed cograph.
2. If $D_1, \ldots, D_k$ are directed cographs, then the disjoint union of $D_1, \ldots, D_k$ is a directed cograph.
3. If $D_1 = (V_1, A_1), \ldots, D_k = (V_k, A_k)$ are directed cographs, then the digraph obtained from the disjoint union of $D_1, \ldots, D_k$ by adding all arcs (v, w) where $v \in V_i$, $w \in V_j$, and $1 \le i < j \le k$, is a directed cograph.
4. If $D_1, \ldots, D_k$ are directed cographs, then the digraph obtained from the disjoint union of $D_1, \ldots, D_k$ by adding all arcs (v, w) where $v \in V_i$, $w \in V_j$, and $i, j \in \{1, \ldots, k\}$, is a directed cograph.

We observe that the complement of a directed cograph is again a directed cograph.

Theorem 9.9.4 ([47]) *A digraph is a directed cograph if and only if it has directed NLC-width at most* 1*.*

Theorem 9.9.4 implies that every directed cograph has directed clique-width at most 2. However, as far as we know, no complete characterization of digraphs of directed clique-width at most 2 is known. We refer to Section 11.6 for more information about directed cographs.

Directed clique-width is incomparable with directed tree-width. In particular, acyclic digraphs have unbounded directed clique-width. A discussion about it is presented in the next subsection. The complete biorientations of undirected complete graphs are directed cographs, but have unbounded directed tree-width.

Lemma 9.9.5

1. *There are classes of digraphs of bounded directed tree-width and unbounded directed clique-width.*
2. *There are classes of digraphs of bounded directed clique-width and unbounded directed tree-width.*

For fixed $k \geq 2$, it is open whether one can recognize graphs of directed clique-width at most k in polynomial time. This is also an open problem for undirected clique-width with $k \geq 4$, and when $k = 3$, it was solved by Corneil, Habib, Lanlignel, Reed and Rotics [21].

Problem 9.9.6 *For an integer $k \geq 2$, can we recognize digraphs of directed clique-width at most k in polynomial time?*

9.9.2 Bi-Rank-Width and $\mathbb{F}_4$-Rank-Width

Rank-width of undirected graphs is a parameter equivalent to clique-width, in a sense that one is bounded if and only if the other is bounded. The rank of a matrix has a role in counting the number of neighborhood types between two vertex sets. To see this, we consider two disjoint vertex sets A and B in an undirected graph $G = (V, E)$, and an $A \times B$-matrix M where for $a \in A$ and $b \in B$, $M[a, b] = 1$ if a is adjacent to b, and $M[a, b] = 0$ otherwise. If the rank of M over the binary field is k, then there are at most 2^k sets in $\{N_G(v) \cap B | v \in A\}$. Rank-width measures the decomposability along vertex partitions with small rank values of such matrices.

Kanté and Rao [59] introduced two directed versions of rank-width, called **bi-rank-width** and $\mathbb{F}_4$**-rank-width**. Kanté and Rao further generalized these notions to $\mathbb{F}$-edge-colored graphs; that is, graphs whose edges are labeled by elements of a fixed finite field $\mathbb{F}$. Since these generalizations are out of scope of this book, we concentrate on specializations for digraphs. A difference of two notions is that when (A, B) is a vertex partition, bi-rank-width is based on a function summing up ranks of two binary matrices, one for arcs from A to B and the other for arcs from B to A, while $\mathbb{F}_4$-rank-width is based on a function measuring all arcs together, using the field $\mathbb{F}_4$.

For a field $\mathbb{F}$ and a matrix M, $\mathbb{F}$-rank(M) is the rank of the matrix M over the field $\mathbb{F}$. We denote by $\mathbb{F}_4$ the field on 4 elements $\{0, 1, a, a^2\}$ where $a^3 = 1$ and $a^2 + a + 1 = 0$. We denote by $\mathbb{F}_2$ the binary field.

Let $D = (V, A)$ be a digraph. The **out-neighborhood matrix** M_D^+ is the $V \times V$-matrix such that for $v, w \in V$, $M_D^+[v, w] = 1$ if and only if $(v, w) \in A$. The $\mathbb{F}_4$**-adjacency matrix** of D is the $V \times V$-matrix M_D^4 where for $v, w \in V$,

$$M_D^4[v,w] := \begin{cases} a & \text{if } (v,w) \in A \text{ and } (w,v) \notin A, \\ a^2 & \text{if } (v,w) \notin A \text{ and } (w,v) \in A, \\ 1 & \text{if } (v,w) \in A \text{ and } (w,v) \in A, \\ 0 & \text{otherwise.} \end{cases}$$

We define functions $\mathrm{bicutrk}_D$, $\mathrm{cutrk}_D^4 : 2^V \to \mathbb{Z}$ such that for every $S \subseteq V$,

- $\mathrm{bicutrk}_D(S) = \mathbb{F}_2\text{-rank}(M_D^+[S, V \setminus S]) + \mathbb{F}_2\text{-rank}(M_D^+[V \setminus S, S])$,
- $\mathrm{cutrk}_D^4(S) = \mathbb{F}_4\text{-rank}(M_D^4[S, V \setminus S])$.

We define branch-decomposition and f-width for symmetric submodular functions f. A function $f : X \to Y$ is **symmetric** if for $S \subseteq X$, $f(S) = f(X \setminus S)$. A function $f : X \to Y$ is **submodular** if it satisfies that for $A, B \subseteq X$, $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$. A tree is **subcubic** if it has at least two vertices and every internal node has degree 3.

Definition 9.9.7 (Branch-decomposition) *Let V be a finite set and let $f : 2^V \to \mathbb{Z}$ be a symmetric submodular function. A branch-decomposition of V is a pair (T, L), where T is a subcubic tree and L is a bijection from V to the set of leaves of T. For an edge e in T, $T - e$ induces a partition (X_e, Y_e) of the leaves of T. The f**-width** of e is defined as $f(L^{-1}(X_e))$, and the f**-width** of a branch-decomposition (T, L) is the maximum f-width over all edges of T. The f**-width** of V is the minimum f-width over all branch-decompositions of V. If $|V| \leq 1$, then V admits no branch-decomposition and the f-width of V is defined to be 0.*

Definition 9.9.8 (Bi-rank-width and $\mathbb{F}_4$-rank-width) *Let $D = (V, A)$ be a digraph. The **bi-rank-width** of D, denoted by* $\mathrm{birw}(D)$, *is the* $\mathrm{bicutrk}_D$*-width of V, and the $\mathbb{F}_4$**-rank-width** of D, denoted by* $\mathrm{rw}^4(D)$, *is the* cutrk_D^4*-width of V.*

Note that the functions $\mathrm{bicutrk}_D$ and cutrk_D^4 are submodular. This can be shown using a property of the rank function of a matrix in Proposition 9.9.9. There are several proofs of it; for instance see Truemper [98].

Proposition 9.9.9 *Let M be an $X \times Y$-matrix over a field $\mathbb{F}$. Then for all $X_1, X_2 \subseteq X$ and $Y_1, Y_2 \subseteq Y$, we have*

$$\begin{aligned} &\mathbb{F}\text{-}\mathit{rank}(M[X_1 \cup X_2, Y_1 \cap Y_2]) + \mathbb{F}\text{-}\mathit{rank}(M[X_1 \cap X_2, Y_1 \cup Y_2]) \\ \leq\ &\mathbb{F}\text{-}\mathit{rank}(M[X_1, Y_1]) + \mathbb{F}\text{-}\mathit{rank}(M[X_2, Y_2]). \end{aligned}$$

Kanté [56] proved that bi-rank-width and $\mathbb{F}_4$-rank-width are equivalent up to a constant factor.

Lemma 9.9.10 ([56]) *For a digraph* $D = (V, A)$, $\mathrm{rw}^4(D) \leq \mathrm{birw}(D) \leq 4\mathrm{rw}^4(D)$.

Digraphs of bi-rank-width at most 2 are digraphs that are completely decomposable with respect to split decomposition introduced by Cunningham [28]. As a similar concept, Kanté and Rao [58] introduced displit decompositions and showed that digraphs of $\mathbb{F}_4$-rank-width at most 1 are digraphs that are completely decomposable with respect to displit decomposition. Both results provide polynomial-time algorithms for recognizing digraphs of bi-rank-width at most 2 or digraphs of $\mathbb{F}_4$-rank-width at most 1.

Acyclic digraphs and tournaments have unbounded bi-rank-width. The grid-like example in Figure 9.8 is acyclic and its underlying undirected graph has large rank-width; Jelínek [53] proved that the undirected $n \times n$-grid has rank-width exactly $n - 1$. A branch-decomposition of a directed graph with small bicutrk-width is also a branch-decomposition of small undirected rank-width and it means that acyclic digraphs have unbounded bi-rank-width. To see that tournaments have unbounded bi-rank-width, we can modify the example in Figure 9.8 into a tournament, such that

- for every two non-adjacent vertices in a column, we add an arc from the higher one to the lower one,
- for every two non-adjacent vertices contained in distinct columns, we add an arc from the right one to the left one.

One can verify that in every its branch-decomposition, there is a vertex partition with high bicutrk value.

Lemma 9.9.11 *The family of acyclic graphs and the family of tournaments have unbounded bi-rank-width. Thus, these families have unbounded directed clique-width and unbounded* $\mathbb{F}_4$*-rank-width.*

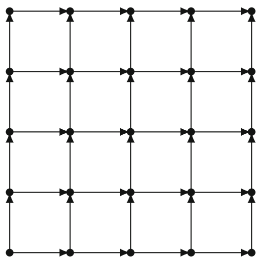

Figure 9.8 Acyclic graphs that have large bi-rank-width.

9.9.3 Computing Rank-Decompositions

We provide FPT approximation algorithms for bi-rank-width and $\mathbb{F}_4$-rank-width. These can be used to obtain an approximated clique-width expression when a graph has small directed clique-width. Oum and Seymour [82] provided a general FPT approximation algorithm on symmetric submodular functions. By adapting the idea of the result of Oum and Seymour, we present FPT approximation algorithms for bi-rank-width and $\mathbb{F}_4$-rank-width.

Let V be a finite set and let $f : 2^V \to \mathbb{Z}$ be a symmetric submodular function. A vertex subset $W \subseteq V$ is called an **f-well-linked set** if for every partition (X, Y) of W and every Z with $X \subseteq Z \subseteq V \setminus Y$, we have $f(Z) \geq \min(|X|, |Y|)$. Oum and Seymour showed that f-well-linked sets are obstructions for graphs of bounded f-width.

Proposition 9.9.12 *1. There exists an algorithm that, given a digraph $D = (V, A)$ and an integer k, runs in time $\mathcal{O}(8^k \cdot poly(|V|))$ either constructs a branch-decomposition of* cutrk^4_D*-width at most* $3k+1$*, or concludes that* $\mathrm{rw}^4(D) > k$.

2. There exists an algorithm that, given a digraph $D = (V, A)$ and an integer k, runs in time $\mathcal{O}(8^k \cdot poly(|V|))$ either constructs a branch-decomposition of $\mathrm{bicutrk}_D$*-width at most* $12k + 4$*, or concludes that* $\mathrm{birw}(D) > k$.

Proof. We claim that if there is a cutrk^4_D-well-linked set of size $3k+1$, then D has $\mathbb{F}_4$-rank-width at least $k+1$. Suppose there is a cutrk^4_D-well-linked set W of size $3k+1$ with respect to cutrk^4_D, and D admits a branch-decomposition (T, L) of cutrk^4_D-width at most k. We proceed to find a vertex partition (A_1, A_2) induced by some edge in T where $\frac{|W|}{3} < |W \cap A_1| \leq \frac{2|W|}{3}$. We subdivide an edge of T, and regard the new vertex as a root node. For each node $t \in V(T)$, let $\mu(t)$ be the number of leaves of T that are descendants of t and mapped to a vertex of W by L. We choose a node t that is farthest from the root node such that $\mu(t) > \frac{|W|}{3}$. By the choice of t, for each child t' of t, $\mu(t') \leq \frac{|W|}{3}$. Therefore, $\frac{|W|}{3} < \mu(t) \leq \frac{2|W|}{3}$. Let e be the edge connecting t and its parent. Clearly, the vertex partition (A_1, A_2) of D induced by e satisfies that for each $i \in \{1, 2\}$, $\frac{|W|}{3} < |A_i \cap W| \leq \frac{2|W|}{3}$. Since W is a cutrk^4_D-well-linked set, we have $\mathrm{cutrk}^4_D(A_1) \geq \max(|W \cap A_1|, |W \cap A_2|) > \frac{|W|}{3} > k$. This contradicts our assumption.

We describe an algorithm that either finds a cutrk^4_D-well-linked set of size $3k+1$ or constructs a branch-decomposition of cutrk^4_D-width at most $3k+1$. In the first case, by the above claim, we conclude that D has $\mathbb{F}_4$-rank-width at least $k+1$.

When we have a mapping g from $V(D)$ to a tree, we say that $g^{-1}(w)$ for $w \in V(D)$ is assigned to the node w. Choose a vertex v of D and start with a tree with two nodes where one contains v and the other contains all vertices of $V \setminus \{v\}$. Recursively choose a node t containing more than one vertex, and let A be the vertex set assigned to t. If $\mathrm{cutrk}^4_D(A) < 3k+1$, then we choose any vertex $a \in A$, and construct a new tree obtained by adding two nodes t_1 and t_2 and edges t_1t, t_2t to T, and assigning a to t_1 and all vertices in $A \setminus \{a\}$ to t_2. Clearly, we have $\text{cutrk}_D^4(A \setminus \{a\}) \leq 3k+1$.

Now we assume $\text{cutrk}_D^4(A) = 3k+1$. In this case, we find a vertex set $B \subseteq V \setminus A$ such that $|B| = 3k+1$ and $\mathbb{F}_4\text{-rank}(M_D^4[A, B]) = 3k+1$. We can find such a set by enumerating a column basis of the matrix $M_D^4[A, V \setminus A]$.

We check whether B is a cutrk_D^4-well-linked set or not. For this, we take all vertex partitions (B_1, B_2) of B, and check for every Z with $B_1 \subseteq Z \subseteq V \setminus B_2$, $\text{cutrk}_D^4(Z) \geq \min(|B_1|, |B_2|)$. We can check this using the submodular function minimization algorithm [52]. If Z is a cutrk_D^4-well-linked set of size $3k+1$, then we output that D has $\mathbb{F}_4$-rank-width at least $k+1$. Otherwise, the procedure outputs a vertex partition (B_1, B_2) of B and a vertex subset Z with $B_1 \subseteq Z \subseteq V \setminus B_2$ where $\text{cutrk}_D^4(Z) < \min(|B_1|, |B_2|)$.

We observe that $A \cap Z$ and $A \setminus Z$ are non-empty. If $A \cap Z = \emptyset$, then $\text{cutrk}_D^4(Z) = \text{cutrk}_D^4(Z \setminus A)$. On the other hand, we have

$$\begin{aligned}
\text{cutrk}_D^4(Z \setminus A) &= \mathbb{F}_4\text{-rank}(M_D^4[Z \setminus A, A \cup (V \setminus Z)]) \\
&\geq \mathbb{F}_4\text{-rank}(M_D^4[B_1, A \setminus Z]) \\
&= |B_1| > \text{cutrk}_D^4(Z),
\end{aligned}$$

which is a contradiction. Therefore, $A \cap Z \neq \emptyset$ and for a similar reason, $A \setminus Z \neq \emptyset$. We construct a tree obtained by adding two nodes t_1 and t_2 and adding edges t_1t, t_2t to T, and assigning $A \cap Z$ to t_1 and $A \setminus Z$ to t_2. We observe

$$\begin{aligned}
\text{cutrk}_D^4(A) + |B_2| &> \text{cutrk}_D^4(A) + \text{cutrk}_D^4(Z) \\
&\geq \text{cutrk}_D^4(A \cap Z) + \text{cutrk}_D^4(A \cup Z) \\
&= \text{cutrk}_D^4(A \cap Z) + \text{cutrk}_D^4(V \setminus (A \cup Z)) \\
&\geq \text{cutrk}_D^4(A \cap Z) + \mathbb{F}_4\text{-rank}(M_D^4[B_2, A]) \\
&= \text{cutrk}_D^4(A \cap Z) + |B_2|.
\end{aligned}$$

This implies that $\text{cutrk}_D^4(A \cap Z) \leq \text{cutrk}_D^4(A) \leq 3k+1$. For a similar reason, we also have $\text{cutrk}_D^4(A \setminus Z) \leq 3k+1$.

Doing this procedure recursively, we obtain either a branch-decomposition of cutrk_D^4-width at most $3k+1$, or conclude that D has $\mathbb{F}_4$-rank-width at least $k+1$. For bi-rank-width, we first run the above algorithm for $\mathbb{F}_4$-rank-width. If it returns that D has $\mathbb{F}_4$-rank-width at least $k+1$, then we can return that it has bi-rank-width at least $k+1$, by Lemma 9.9.10. If the algorithm outputs a branch-decomposition of cutrk_D^4-width at most $3k+1$, then this is also a branch-decomposition of bicutrk_D-width at most $4(3k+1) = 12k+4$, by Lemma 9.9.10. □

Later, Oum [78] investigated an FPT approximation algorithm for undirected rank-width that runs in time $\mathcal{O}(8^k \cdot n^4)$, by replacing the submodular function minimization algorithm with an elementary algorithm that fits to rank-width. Oum [79] raised an open problem whether it can be further reduced to $\mathcal{O}(c^k \cdot n^3)$ for some constant c. We ask the same questions for bi-rank-width and $\mathbb{F}_4$-rank-width on digraphs. Note that when allowing c^{k^2} in the parameter part, one can obtain $\mathcal{O}(n^3)$ running time due to Hliněný [48].

Problem 9.9.13 *Is there a constant-factor FPT approximation algorithm for bi-rank-width or $\mathbb{F}_4$-rank-width that runs in time $\mathcal{O}(c^k n^3)$ for some constant c?*

Kanté and Rao [59] observed that as an application of the result of Hliněný and Oum [49], there are also exact FPT algorithms for both parameters. Briefly, Hliněný and Oum developed an exact FPT algorithm for partitioned matroids with respect to matroid branch-width, and then applied to rank-width. This application is also possible for bi-rank-width or $\mathbb{F}_4$-rank-width. Note that the function $g(k)$ in Theorem 9.9.14 is triple exponential.

Theorem 9.9.14 ([59])

1. *There exists an algorithm that, given a digraph $D = (V, A)$ and an integer k, runs in time $g(k)|V|^3$ for some function g and either constructs a branch-decomposition of cutrk^4_D-width at most k, or concludes that $\mathrm{rw}^4(D) > k$.*
2. *There exists an algorithm that, given a digraph $D = (V, A)$ and an integer k, runs in time $g(k)|V|^3$ for some function g and either constructs a branch-decomposition of $\mathrm{bicutrk}_D$-width at most k, or concludes that $\mathrm{birw}(D) > k$.*

We observe that a branch-decomposition of bounded $\mathrm{bicutrk}_D$-width can be efficiently translated to directed clique-width expression. A similar observation for undirected rank-width and clique-width was discussed by Oum and Seymour [82]. We remark that Courcelle and Engelfriet [24, Proposition 6.8] proved that one can approximate directed clique-width using undirected rank-width.

Lemma 9.9.15 *For a digraph $D = (V, A)$, $\frac{\mathrm{birw}(D)}{2} \leq \mathrm{dcw}(D) \leq 2^{\mathrm{birw}(D)+1} - 1$. Moreover, given a digraph D and its branch-decomposition of $\mathrm{bicutrk}_D$-width k, one can construct a directed clique-width $(2^{k+1} - 1)$-expression in time $\mathcal{O}(4^k|V|^3)$.*

Proof. We prove that $\mathrm{birw}(D) \leq 2\mathrm{dcw}(D)$. If $|V| = 1$, then $\mathrm{birw}(D) = 0$ and $\mathrm{dcw}(D) = 1$, and the statement holds. We may assume $|V| \geq 2$. Let $k = \mathrm{dcw}(D)$ and let T be a directed clique-width k-expression tree of D. Note that this tree is a tree with maximum degree 3, and each leaf node is a node introducing a vertex of D. We choose an edge $e = uv$ of T where u is a child of v. The constructed graph D_u at node u is a k-labeled graph, and each vertex set in D_u having the same label has the same out-neighborhood and in-neighborhood to $V \setminus V(D_u)$. This means that $\text{bicutrk}_D(V(D_u)) \leq 2k$, and it shows that $\text{birw}(D) \leq 2k$.

Now, we prove that given a branch-decomposition of D of bi-rank-width k, one can construct a directed clique-width $(2^{k+1}-1)$-expression in time $\mathcal{O}(4^k|V|^3)$. This also proves the inequality $\text{dcw}(D) \leq 2^{\text{birw}(D)+1}-1$. Let (T, L) be a given branch-decomposition of D of bicutrk_D-width k. We choose an edge of T and subdivide this edge with adding a new node r, and we consider T as a tree with the root node r. For each node t of T, let $D_t = (V_t, A_t)$ be the digraph induced by the set of vertices of D that are mapped to a descendant of t. Let $\sim_t$ be the equivalent class on V_t such that $v \sim_t w$ if and only if $N_D^+(v) \cap (V \setminus V_t) = N_D^+(w) \cap (V \setminus V_t)$ and $N_D^-(v) \cap (V \setminus V_t) = N_D^-(w) \cap (V \setminus V_t)$. We denote by $V_t/\sim_t$ be the set of equivalent classes. We note that since D has bicutrk_D-width k, there are at most 2^k equivalent classes in $V_t/\sim_t$ for each node t.

We prove by induction on the number of descendants of T that $D_t = (V_t, A_t)$ has a labeling lab_t satisfying that

1. (V_t, A_t, lab_t) can be constructed by a directed clique-width $(2^{k+1}-1)$-expression,
2. lab_t is a 2^k-labeling of D_t,
3. for $v, w \in V_t$, if v and w are contained in distinct classes of $V_t/\sim_t$, then $\text{lab}_t(v) \neq \text{lab}_t(w)$, and
4. $\{v \in V_t : \text{lab}_t(v) = 1\}$ is exactly the set of vertices in V_t having no in-neighborhood and no out-neighborhood in $V \setminus V_t$.

If t is a leaf node, then it is clear. Assume that t is not a leaf, and let t_1 and t_2 be the two children of t. By induction hypothesis, for each $i \in \{1, 2\}$, D_{t_i} has a labeling lab_i satisfying the conditions. For $j > 1$, we change each label j of V_{t_1} to $j + (2^k - 1)$, and then take the disjoint union of D_{t_1} and D_{t_2}. Then we add arcs between D_{t_1} and D_{t_2} according to the adjacency relation between D_{t_1} and D_{t_2}. Note that when we add an arc from $v_1 \in V_{t_1}$ to $v_2 \in V_{t_2}$, we add all arcs from the vertices in the class of $V_{t_1}/\sim_{t_1}$ containing v_1 and to the vertices in the class of $V_{t_2}/\sim_{t_2}$ containing v_2.

For each $i \in \{1, 2\}$, we relabel V_{t_i} according to the class $V_t/\sim_t$. This is possible as $V_{t_i}/\sim_{t_i}$ is a refinement of $V_t/\sim_t$ on V_{t_i}. Then we relabel V_{t_1} according to the labeling of V_{t_2} so that the resulting labeling lab_t on V_t satisfies that

- for $v, w \in V_t$, if v and w are contained in distinct classes of $V_t/\sim_t$, then $\text{lab}_t(v) \neq \text{lab}_t(w)$, and
- $\{v \in V_t : \text{lab}_t(v) = 1\}$ is exactly the set of vertices in V_t having no in-neighborhood and no out-neighborhood in $V \setminus V_t$.

This concludes the proof. □

9.9.4 Algorithmic Applications

We present algorithms based on directed clique-width or bi-rank-width. Courcelle, Makowsky and Rotics [26] showed that every problem expressible in MSO_1 logic can be solved in polynomial time on graphs of bounded directed clique-width.

Theorem 9.9.16 ([26]) *Every problem expressible in MSO_1 logic is fixed parameter tractable with respect to the parameter directed clique-width.*

For many problems, we can design a dynamic-programming algorithm with running time much better than one guaranteed by Theorem 9.9.16. For instance, the problem of finding a minimum dominating set can be solved in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ when a directed clique-width k-expression is given.

Theorem 9.9.17 *Given a digraph $D = (V, A)$ and its directed clique-width k-expression, one can compute a minimum directed dominating set of D in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$.*

We briefly present how to formulate table indices. Let ϕ be the given k-expression defining D, and let T be the labeled rooted tree induced by ϕ. For every node t of T, let D_t be the subgraph of D defined at node t, and for each $i \in \{1, \ldots, k\}$, let $D_t[i]$ be the subgraph of D_t induced on the set of vertices with label i.

The property of the constructed graph D_t at some node t is that two vertices in a same label class have same in-neighbors and out-neighbors in $V \setminus V(D_t)$. In the table of dynamic programming, we store the information that label classes that are completely dominated by some vertices of D_t, and label classes containing a vertex taken as a dominating set. We can recursively check whether a proper dominating set exists with given these information and a fixed size. This is similar to one developed for undirected case by Kobler and Rotics [67].

For many problems whose solutions can be locally checked, we can similarly design dynamic programming algorithms for problems on digraphs of bounded clique-width, which runs in FPT time. However, it becomes different when a solution requires some global property such as connectivity. For instance, Fomin, Golovach, Lokshtanov and Saurabh [36] proved that the problem of testing whether there is a hamiltonian cycle is $W[1]$-hard parameterized by clique-width, and later the same authors proved that this problem does not admit an algorithm with running time $n^{o(k)}$ under the ETH assumption [35]. On the other hand, it can be solved in time $n^{\mathcal{O}(k^2)}$, similar to the undirected case [32].

Theorem 9.9.18 *Given a digraph $D = (V, A)$ and its directed clique-width k-expression, one can test whether D contains a hamiltonian cycle in time $n^{\mathcal{O}(k^2)}$.*

We briefly explain the idea of Theorem 9.9.18. If D contains a hamiltonian cycle, then its restriction on D_t forms a partition of D_t into vertex-disjoint paths unless $D_t \neq D$. Thus, if we have all possible partitions of D_t into vertex-disjoint paths for each node t, then at the last node, we can test whether there is a hamiltonian cycle. One could observe that if there are two partitions into paths where for every pair (i, j) of integers in $\{1, 2, \dots, k\}$, the number of paths from $D_t[i]$ to $D_t[j]$ is equal, then they have the same role in generating a hamiltonian cycle. Thus, in the indices of tables, we are given some integer for every pair of integers, and we check whether there is a partition into paths meeting this condition. Using this table scheme, we can solve it in time $n^{\mathcal{O}(k^2)}$. Bergounoux, Kanté and Kwon [10] announced that the running time can be further improved to $n^{\mathcal{O}(k)}$.

There are more interesting problems that can be solved in FPT or XP time parameterized by clique-width. For instance, Parity Game can be solved in polynomial time on digraphs of constant directed clique-width [76]. We refer to [41] for more examples.

One issue of using directed clique-width is that if we approximate directed clique-width using Lemma 9.9.15 from obtained rank-decomposition, it is unavoidable single-exponential blow-up on the parameter. Thus, designing an algorithm directly using branch-decompositions of small bi-rank-width is an interesting problem. Ganian, Hliněný and Obdržálek [44] used parsing trees for rank-width to design XP algorithms for several problems such as Graph Coloring, Chromatic Polynomial, and Hamiltonian Path problems. More examples can be found in [42, 45].

9.9.5 Vertex-Minors and Pivot-Minors

We introduce **pivot-minor** and **vertex-minor** relations in digraphs. These containment relations are defined using graph operations **pivoting** and **local complementation**, respectively. In undirected graphs, local complementation at a vertex v is an operation to replace the neighborhood of v with its complement. Local complementation was introduced in the study of circle graphs [19], 2-regular Eulerian digraphs and isotropic systems [17, 18] by Bouchet. Pivoting also came up in the study of graphic representations of isotropic systems [17], and it is represented as three successive local complementations at v, w, v on two adjacent vertices v and w. Bouchet [16] observed that the cut-rank function does not change when applying local complementation [17], and based on this property, Oum [80, 81] investigated several structural results related to rank-width. Later, Kanté and Rao [59] extended the notion of local complementation and pivoting to digraphs.

We introduce here the pivoting operation in a digraph. Let M be a $V \times V$-matrix on $\mathbb{F}_4$, and let x, y be distinct elements in V such that $M[x, y] \neq 0$. The matrix $M * (x, y)$ is a $V \times V$-matrix such that $(M * (x, y))[z, z] := 0$ for all $z \in V$, and for all $s, t \in V \setminus \{x, y\}$ with $s \neq t$,

- $(M*(x,y))[s,t] := M[s,t] - \frac{M[s,x]\cdot M[y,t]}{M[y,x]} - \frac{M[s,y]\cdot M[x,t]}{M[x,y]}$,
- $(M*(x,y))[x,t] := \frac{M[y,t]}{M[y,x]}$, $(M*(x,y))[y,t] := -\frac{M[x,t]}{M[x,y]}$,
- $(M*(x,y))[s,x] := -\frac{M[s,y]}{M[x,y]}$, $(M*(x,y))[s,y] := \frac{M[s,x]}{M[y,x]}$,
- $(M*(x,y))[x,y] := -\frac{1}{M[y,x]}$, $(M*(x,y))[y,x] := -\frac{1}{M[x,y]}$,

where all equations are computed over $\mathbb{F}_4$. For an arc (v,w) of a digraph D, a digraph obtained by **pivoting** vw is defined as the digraph whose $\mathbb{F}_4$-adjacency matrix is $M_D^4 * (v,w)$, and it is denoted by $D \wedge vw$. A digraph H is a **pivot-minor** of a digraph D if H can be obtained from D by a sequence of pivotings and vertex deletions. We observe that pivot operations do not change the function cutrk_D^4.

Lemma 9.9.19 ([59]) *Let $D=(V,A)$ be a digraph. Every pivot operation does not change the function cutrk_D^4, and thus, if a digraph H is a pivot-minor of D, then $\mathrm{rw}^4(H) \le \mathrm{rw}^4(D)$.*

Proof. Let (x,y) be an arc of D, and let $X \subseteq V$ and $Y = V \setminus X$. It is enough to prove that $\mathrm{cutrk}_D^4(X) = \mathrm{cutrk}_{D\wedge xy}^4(X)$. Without loss of generality, we may assume $x \in X$. We divide cases depending on whether $y \in X$ or not. First assume that $y \in X$, and let $X' := X \setminus \{x,y\}$. In this case, we have

$$
\begin{aligned}
&\mathbb{F}_4\text{-rank}(M^4_{D\wedge xy}[X,Y]) \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} \frac{1}{M_D^4[y,x]}\cdot M_D^4[y,Y] \\ \frac{-1}{M_D^4[x,y]}\cdot M_D^4[x,Y] \\ M_D^4[X',Y] - \frac{M_D^4[X',x]\cdot M_D^4[y,Y]}{M_D^4[y,x]} - \frac{M_D^4[X',y]\cdot M_D^4[x,Y]}{M_D^4[x,y]} \end{pmatrix} \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} \frac{1}{M_D^4[y,x]}\cdot M_D^4[y,Y] \\ \frac{-1}{M_D^4[x,y]}\cdot M_D^4[x,Y] \\ M_D^4[X',Y] \end{pmatrix} = \mathbb{F}_4\text{-rank}(M_D^4[X,Y]).
\end{aligned}
$$

Now, we assume that $y \notin X$, and let $X' := X \setminus \{x\}$ and $Y' := Y \setminus \{y\}$. Then we have

$$
\begin{aligned}
&\mathbb{F}_4\text{-rank}(M^4_{D\wedge xy}[X,Y]) \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} -\frac{1}{M^4_D[y,x]} & \frac{-1}{M^4_D[y,Y']}\cdot M^4_D[y,x] \\ \frac{M^4_D[X',x]}{M^4_D[y,x]} & M^4_D[X',Y'] - \frac{M^4_D[X',x]\cdot M^4_D[y,Y']}{M^4_D[y,x]} - \frac{M^4_D[X',y]\cdot M^4_D[x,Y']}{M^4_D[x,y]} \end{pmatrix} \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} -\frac{1}{M^4_D[y,x]} & \frac{-1}{M^4_D[y,Y']}\cdot M^4_D[y,x] \\ 0 & M^4_D[X',Y'] - \frac{M^4_D[X',y]\cdot M^4_D[x,Y']}{M^4_D[x,y]} \end{pmatrix} \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} -\frac{1}{M^4_D[y,x]} & 0 \\ 0 & M^4_D[X',Y'] - \frac{M^4_D[X',y]\cdot M^4_D[x,Y']}{M^4_D[x,y]} \end{pmatrix} \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} M^4_D[x,y] & 0 \\ M^4_D[X',y] & M^4_D[X',Y'] - \frac{M^4_D[X',y]\cdot M^4_D[x,Y']}{M^4_D[x,y]} \end{pmatrix} \\
&= \mathbb{F}_4\text{-rank}\begin{pmatrix} M^4_D[x,y] & M^4_D[x,Y'] \\ M^4_D[X',y] & M^4_D[X',Y'] \end{pmatrix} = \mathbb{F}_4\text{-rank}(M^4_D[X,Y]).
\end{aligned}
$$

□

Kanté [57] showed that digraphs of bounded $\mathbb{F}_4$-rank-width are well-quasi-ordered under the pivot-minor operation. Note that the class of digraphs of bounded $\mathbb{F}_4$-rank-width is not well-quasi-ordered under the induced subdigraph operation. The set of all directed cycles is such an example.

Theorem 9.9.20 ([57]) *Every pivot-minor closed class of digraphs of $\mathbb{F}_4$-rank-width at most k is well-quasi-ordered under the pivot-minor relation.*

In undirected case, it is an open problem whether graphs are well-quasi-ordered under the undirected version of pivot-minor relation. If this holds, then it would imply the graph minor theorem which say that graphs are well-quasi-ordered under the minor relation. We ask the same question for directed graphs.

Problem 9.9.21 *Is the set of digraphs well-quasi-ordered under the pivot-minor relation?*

It is open whether we can check whether a fixed graph H is a pivot-minor of a graph G for undirected graphs. Courcelle and Oum [27] proved that this problem is solvable in polynomial time when underlying graphs have bounded rank-width. Results from [57] imply that the same question for directed graphs is solvable in polynomial time when underlying digraphs have bounded $\mathbb{F}_4$-rank-width. We ask a question for general digraphs, as for undirected graphs.

Problem 9.9.22 *For every fixed digraph H, is there a polynomial time algorithm testing whether a digraph G contains H as a pivot-minor?*

For a vertex v in a digraph $D = (V, A)$, the $\mathbb{F}_4$**-local complementation** at v, denote by $D * v$, is the operation to take the digraph with the $\mathbb{F}_4$-adjacency matrix M' where

- for $x, y \in V$ with $x \neq y$, $M'[x, y] = M_D^4[x, y] + M_D^4[x, z]M_D^4[z, y]$,
- for $x \in V$, $M'[x, x] = 0$.

A digraph H is an $\mathbb{F}_4$**-vertex-minor** of D if H can be obtained from D by a sequence of local complementations and vertex deletions. Note that as in the undirected case, it is satisfied that $D \wedge vw = D * v * w * v$ [59].

Lemma 9.9.23 ([59]) *Let $D = (V, A)$ be a digraph. Every $\mathbb{F}_4$-local complementation does not change the function* cutrk_D^4, *and thus if H is an $\mathbb{F}_4$-vertex-minor of a digraph D, then* $\mathrm{rw}^4(H) \leq \mathrm{rw}^4(D)$.

Proof. Let $D = (V, A)$ be a digraph and x be a vertex of D. Let $X \subseteq V$. We may assume that $x \in X$ as $\mathrm{cutrk}_D^4(X) = \mathrm{cutrk}_D^4(V \setminus X)$. For each $y \in X$, the $\mathbb{F}_4$-local complementation at x results in adding a multiple of the row indexed by x to the row indexed by y. Therefore, we have $\mathrm{cutrk}_{D*x}^4(X) = \mathrm{cutrk}_D^4(X)$. □

Kanté and Rao [59] proved that the size of a minimal vertex-minor or pivot-minor obstruction for digraphs of $\mathbb{F}_4$-rank-width at most k is bounded by a function of k.

Theorem 9.9.24 ([59])

1. *For each positive integer k, there is a set $\mathcal{C}_k^v$ of directed graphs each having at most $(6^{k+1} - 1)/5$ vertices, such that a digraph has $\mathbb{F}_4$-rank-width at most k if and only if it has no $\mathbb{F}_4$-vertex-minor isomorphic to digraphs in $\mathcal{C}_k^v$.*
2. *For each positive integer k, there is a set $\mathcal{C}_k^p$ of directed graphs each having at most $(6^{k+1} - 1)/5$ vertices, such that a digraph has $\mathbb{F}_4$-rank-width at most k if and only if it has no pivot-minor isomorphic to digraphs in $\mathcal{C}_k^p$.*

A similar variant of local complementation can be defined in a way that it preserves the bi-rank-width of a digraph. For a vertex v in a digraph $D = (V, A)$, the $\mathbb{F}_2$**-local complementation** at v, denote by $D *_2 v$, is the operation to take the digraph with the out-neighborhood matrix M' where

- for $x, y \in V$ with $x \neq y$, $M'[x, y] = M_D^+[x, y] + M_D^+[x, z]M_D^+[z, y]$,
- for $x \in V$, $M'[x, x] = 0$.

A digraph H is an $\mathbb{F}_2$**-vertex-minor** of D if H can be obtained from D by a sequence of local complementations and vertex deletions.

Lemma 9.9.25 ([59]) *Let $D = (V, A)$ be a digraph. Every $\mathbb{F}_2$-local complementation does not change the function* $\mathrm{bicutrk}_D$, *and thus if H is an $\mathbb{F}_2$-vertex-minor of a digraph D, then* $\mathrm{birw}(H) \leq \mathrm{birw}(D)$.

Proof. Let $D = (V, A)$ be a digraph and x be a vertex of D. Let $X \subseteq V$. We may assume that $x \in X$. In the matrix $M_D^+[X, V \setminus X]$, for each $y \in X \setminus \{x\}$, the $\mathbb{F}_2$-local complementation at x results in adding a multiple of the row indexed by x to the row indexed by y. Therefore, we have $\text{bicutrk}_{D*x}(X) = \text{bicutrk}_D(X)$. □

Kanté and Rao [59] discussed that their generalization of pivot operation for edge-colored graphs does not fit to bi-rank-width. Also, they observed that digraphs of bounded bi-rank-width are not well-quasi-ordered under the $\mathbb{F}_2$-vertex-minor relation. The set of digraphs whose underlying graphs are even cycles such that each vertex has either in-degree 2 or out-degree 2 has bounded bi-rank-width and is not well-quasi-ordered by the $\mathbb{F}_2$-vertex-minor relation. Any $\mathbb{F}_2$-local complementation at a vertex of such cycle does not create any new arc, and thus, it is implied by the observation that such cycles are not well-quasi-ordered under the induced subdigraph relation. Furthermore, we can observe that all of such cycles are $\mathbb{F}_2$-vertex-minor obstructions for digraphs of bi-rank-width at most 1. Thus, we could not expect an upper bound on the size of $\mathbb{F}_2$-vertex-minor obstructions for digraphs of bounded bi-rank-width as in Theorem 9.9.24.

References

1. I. Adler. Directed tree-width examples. *J. Combin. Theory Ser. B*, 97(5):718–725, 2007.
2. B. Alspach. Searching and sweeping graphs: a brief survey. *Matematiche (Catania)*, 59:5–37, 2006.
3. S. Amiri, L. Kaiser, S. Kreutzer, R. Rabinovich, and S. Siebertz. Graph searching games and width measures for directed graphs. In *STACS 2015: 32nd International Symposium on Theoretical Aspects of Computer Science*, volume 30 of *LIPIcs*, pages 34–47, 2015.
4. S. Amiri, K. Kawarabayashi, S. Kreutzer, and P. Wollan. The Erdős-Posa property for directed graphs. *CoRR*, arXiv:1603.02504, 2016.
5. S. Arnborg, D. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Matrix Analysis Applications*, 8(2):277–284, 1987.
6. G. Bagan, A. Bonifati, and B. Groz. A trichotomy for regular simple path queries on graphs. In *PODS 2013: 32nd Symposium on Principles of Database Systems*, pages 261–272, New York, NY, USA, 2013. ACM.
7. J. Bang-Jensen and T.M. Larsen. DAG-width and circumference of digraphs. *J. Graph Theory*, 82:194–206, 2016.
8. J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs Combin.*, 22(2):161–172, 2006.
9. J. Barat, P. Hajnal, Y. Lin, and A. Yang. On the structure of graphs with path-width at most two. *Studia Scientiarum Mathematicarum Hungarica*, 49(2):211–222, 2012.
10. B. Bergougnoux, M.M. Kanté, and O. Kwon. An optimal XP algorithm for hamiltonian cycle on graphs of bounded clique-width. In *WADS 2017: 15th International Symposium on Algorithms and Data Structures*, volume 10389 of *Lect. Notes Comput. Sci.*, pages 121–132, 2017.

11. D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-width and parity games. In *STACS 2006: Symp. on Theoretical Aspects of Computer Science*, volume 3884 of *Lect. Notes Comput. Sci.*, pages 524–536. Springer, 2006.
12. D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, and J. Obržálek. DAG-width and parity games. *J. Combin. Theory Ser. B*, 102(4):900–923, 2012.
13. D. Berwanger and E. Grädel. Entanglement – a measure for the complexity of directed graphs with applications to logic and games. In *LPAR 2005: Logic for Programming, Artificial Intelligence, and Reasoning*, Lect. Notes Artif. Intel., pages 209–223, 2005.
14. D. Berwanger, E. Grädel, L. Kaiser, and R. Rabinovich. Entanglement and the complexity of directed graphs. *Theor. Comput. Sci.*, 463:2–25, 2012.
15. D. Bienstock and P.D. Seymour. Monotonicity in graph searching. *J. Algor.*, 12(2):239–245, 1991.
16. A. Bouchet. Connectivity of isotropic systems. In *Combinatorial Mathematics: 3rd International Conference (New York, 1985)*, volume 555 of *Ann. New York Acad. Sci.*, pages 81–93, New York, 1989. New York Acad. Sci.
17. A. Bouchet. Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B*, 45(1):58–76, 1988.
18. A. Bouchet. Isotropic systems. *European J. Combin.*, 8(3):231–244, 1987.
19. A. Bouchet. Reducing prime graphs and recognizing circle graphs. *Combinatorica*, 7(3):243–254, 1987.
20. M. Chudnovsky, A. Scott, and P.D. Seymour. Disjoint paths in tournaments. *Adv. Math.*, 270:582–597, 2015.
21. D.G. Corneil, M. Habib, J. Lanlignel, B. Reed, and U. Rotics. Polynomial-time recognition of clique-width ≤ 3 graphs. *Discrete Appl. Math.*, 160(6):834–865, 2012.
22. D.G. Corneil, H. Lerchs, and L.S. Burlingham. Complement reducible graphs. *Discrete Appl. Math.*, 3(3):163–174, 1981.
23. B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, pages 194 – 242. Elsevier, 1990.
24. B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic*. Cambridge Univ. Press, Cambridge, 2012. A language-theoretic approach, With a foreword by M. Nivat.
25. B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. System Sci.*, 46(2):218–270, 1993.
26. B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.
27. B. Courcelle and S. Oum. Vertex-minors, monadic second-order logic, and a conjecture by Seese. *J. Combin. Theory Ser. B*, 97(1):91–126, 2007.
28. W.H. Cunningham. Decomposition of directed graphs. *SIAM J. Algebraic Discrete Methods*, 3(2):214–228, 1982.
29. M. de Oliveira Oliveira. An algorithmic metatheorem for directed treewidth. *Discrete Appl. Math.*, 204:49–76, 2016.
30. Z. Dvořák. Constant-factor approximation of the domination number in sparse graphs. *European J. Combin.*, 34(5):833–840, 2013.
31. K. Edwards, I. Muzi, and P. Wollan. Half-integral linkages in highly connected directed graphs. *CoRR*, arXiv:1611.01004, 2016.
32. W. Espelage, F. Gurski, and E. Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *WG 2001: Graph-theoretic concepts in computer science*, volume 2204 of *Lect. Notes Comput. Sci.*, pages 117–128. Springer, Berlin, 2001.

33. H. Fernau and D. Meister. Digraphs of bounded elimination width. *Discrete Appl. Math.*, 168:78–87, 2014.
34. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. ISBN 3-54-029952-1.
35. F.V. Fomin, P.A. Golovach, D. Lokshtanov, and S. Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM J. Comput.*, 43(5):1541–1563, 2014.
36. F.V. Fomin, P.A. Golovach, D. Lokshtanov, and S. Saurabh. Intractability of clique-width parameterizations. *SIAM J. Comput.*, 39(5):1941–1956, 2010.
37. F.V. Fomin and M. Pilipczuk. Jungles, bundles and fixed-parameter tractability. In *SODA 2013: 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 396–413, 2013.
38. F.V. Fomin and D.M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.
39. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
40. A. Fradkin and P.D. Seymour. Edge-disjoint paths in digraphs with bounded independence number. *J. Combin. Theory Ser. B*, 110:19–46, 2015.
41. R. Ganian, P. Hliněný, J. Kneis, A. Langer, J. Obdržálek, and P. Rossmanith. Digraph width measures in parameterized algorithmics. *Discrete Appl. Math.*, 168:88–107, 2014.
42. R. Ganian, P. Hliněný, J. Kneis, A. Langer, J. Obdržálek, and P. Rossmanith. On digraph width measures in parameterized algorithmics. In *IWPEC 2009: Parameterized and Exact Computation*, volume 5917 of *Lect. Notes Comput. Sci.*, pages 185–197. Springer, Berlin, 2009.
43. R. Ganian, P. Hliněný, J. Kneis, D. Meister, J. Obdržálek, P. Rossmanith, and S. Sikdar. Are there any good digraph width measures? *J. Combin. Theory Ser. B*, 116:250–286, 2016.
44. R. Ganian, P. Hliněný, and J. Obdržálek. A unified approach to polynomial algorithms on graphs of bounded (bi-)rank-width. *European J. Combin.*, 34(3):680–701, 2013.
45. R. Ganian, P. Hliněný, and J. Obdržálek. Better algorithms for satisfiability problems for formulas of bounded rank-width. *Fund. Inform.*, 123(1):59–76, 2013.
46. M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. In *STOC 2014: 46th Annual ACM Symposium on Theory of Computing*, pages 89–98. ACM, 2014.
47. F. Gurski, E. Wanke, and E. Yilmaz. Directed NLC-width. *Theor. Comput. Sci.*, 616:1–17, 2016.
48. P. Hliněný. A parametrized algorithm for matroid branch-width. *SIAM J. Comput.*, 35(2):259–277, loose erratum, 2005.
49. P. Hliněný and S. Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008.
50. P. Hunter. Losing the +1: Directed path-width games are monotone. Unpublished note, available from Feb. 2017 at www.cs.ox.ac.uk/paul.hunter/papers/losing.pdf, 2006.
51. P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and ordering. *Theor. Comput. Sci.*, 399(3):206–219, 2008.
52. S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. Assoc. Comput. Mach.*, 48:761–777, 2001.
53. V. Jelínek. The rank-width of the square grid. *Discrete Appl. Math.*, 158(7):841–850, 2010.

54. T. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *J. Combin. Theory Ser. B*, 82(1):138–154, 2001.
55. L. Kaiser, S. Kreutzer, R. Rabinovich, and S. Siebertz. Directed width measures and monotonicity of directed graph searching. *CoRR*, arXiv:1408.4745, 2014.
56. M.M. Kanté. The rank-width of directed graphs. arXiv:0709.1433, September 2007.
57. M.M. Kanté. Well-quasi-ordering of matrices under Schur complement and applications to directed graphs. *European J. Combin.*, 33(8):1820–1841, 2012.
58. M.M. Kanté and M. Rao. Directed rank-width and displit decomposition. In C. Paul and M. Habib, editors, *WG 2009: Graph-Theoretic Concepts in Computer Science*, volume 5911 of *Lect. Notes Comput. Sci.*, pages 214–225. Springer Berlin Heidelberg, 2010.
59. M.M. Kanté and M. Rao. The rank-width of edge-coloured graphs. *Theory Comput. Syst.*, 52(4):599–644, 2013.
60. K. Kawarabayashi, Y. Kobayashi, and S. Kreutzer. An excluded half-integral grid theorem for digraphs and the directed disjoint paths problem. In *STOC 2014: The ACM Symposium on Theory of Computing*, pages 70–78, New York, NY, USA, 2014. ACM.
61. K. Kawarabayashi and S. Kreutzer. The directed grid theorem. In *STOC 2015: 47th Annual ACM on Symposium on Theory of Computing*, pages 655–664. ACM, 2015.
62. H.A. Kierstead and D. Yang. Orderings on graphs and game coloring number. *Order*, 20:255–264, 2003.
63. S. Kintali. Directed width parameters and circumference of digraphs. *Theor. Comput. Sci.*, 659:83–87, 2017.
64. S. Kintali, N. Kothari, and A. Kumar. Approximation algorithms for digraph width parameters. *Theor. Comput. Sci.*, 562:365 – 376, 2015.
65. S. Kintali and Q. Zhang. Forbidden directed minors and kelly-width. *Theor. Comput. Sci.*, 662:40 – 47, 2017.
66. K. Kitsunai, Y. Kobayashi, K. Komuro, H. Tamaki, and T. Tano. Computing directed pathwidth in $O(1.89^n)$ time. *Algorithmica*, 75(1):138–157, 2016.
67. D. Kobler and U. Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Appl. Math.*, 126(2-3):197–221, 2003.
68. S. Kreutzer. Graph searching games. In K.R. Apt and E. Grädel, editors, *Lectures in Game Theory for Computer Scientists*, chapter 7, pages 213–263. Cambridge University Press, 2011.
69. S. Kreutzer and S. Ordyniak. Digraph decompositions and monotonicity in digraph searching. *Theor. Comput. Sci.*, 412(35):4688–4703, 2011.
70. S. Kreutzer and S. Ordyniak. Width-measures for directed graphs and algorithmic applications. In *Quantitative graph theory*, Discrete Math. Appl. (Boca Raton), pages 181–231. CRC Press, Boca Raton, FL, 2015.
71. S. Kreutzer, R. Rabinovich, S. Siebertz, and G. Weberstädt. Structural properties and constant factor-approximation of strong distance-r dominating sets in sparse directed graphs. In *STACS 2017: 34th International Symposium on Theoretical Aspects of Computer Science*, 2017.
72. S. Kreutzer and S. Tazari. Directed nowhere dense classes of graphs. In *SODA 2012: 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1552–1562, 2012.
73. D. Meister, J.A. Telle, and M. Vatshelle. Recognizing digraphs of Kelly-width 2. *Discrete Appl. Math.*, 158(7):741–746, 2010.

74. J. Nešetřil and P.O. de Mendez. Grad and classes with bounded expansion I–III. *European J. Combin.*, 29, 2008. Series of 3 papers appearing in volumes (3) and (4).
75. J. Nešetřil and P.O. de Mendez. On nowhere dense graphs. *European J. Combin.*, 32(4):600–617, 2011.
76. J. Obdržálek. Clique-width and parity games. In *CSL 2007/EACSL 2007: 21st International Conference, and the 16th Annuall Conference on Computer Science Logic*, volume 4646 of *Lect. Notes Comput. Sci.*, pages 54–68, Berlin, Heidelberg, 2007. Springer-Verlag.
77. J. Obdržálek. DAG-width: connectivity measure for directed graphs. In *SODA 2006: Symp. on Discrete Algorithms*, pages 814–821, 2006.
78. S. Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):Art. 10, 20, 2009.
79. S. Oum. Rank-width: Algorithmic and structural results. *Discrete Appl. Math.*, 2016.
80. S. Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005.
81. S. Oum. Rank-width and well-quasi-ordering. *SIAM J. Discrete Math.*, 22(2):666–682, 2008.
82. S. Oum and P.D. Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006.
83. R. Rabinovich. *Graph Complexity Measures and Monotonicity*. PhD thesis, RWTH Aachen University, 2013.
84. B. Reed. Introducing directed tree-width. *Elec. Notes Discrete Math.*, 3:222 – 229, 1999.
85. B. Reed. Tree width and tangles: A new connectivity measure and some applications. In R.A. Bailey, editor, *Surveys in Combinatorics*, volume 241, pages 87–162. Cambridge University Press, 1997.
86. B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
87. F. Reidl, F.S. Villaamil, and K. Stavropoulos. Characterising bounded expansion by neighbourhood complexity. arXiv:1603.09532, 2016.
88. N. Robertson and P.D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.
89. N. Robertson and P.D. Seymour. Graph minors V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986.
90. M.A. Safari. D-width: A more natural measure for directed tree width. In *MFCS 2005: Mathematical Foundations of Computer Science*, number 3618 in Lect. Notes Comput. Sci., pages 745 – 756, 2005.
91. N. Sauer. On the density of families of sets. *J. Combin. Theory Ser. A*, 13(1):145–147, 1972.
92. P.D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *J. Combin. Theory Ser. B*, 58(1):22–33, 1993.
93. S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific J. Math.*, 41(1):247–261, 1972.
94. B. Sheppard. DNA sequencing by hybridization. Master's thesis, Memorial University of Newfoundland, 2014.
95. A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. *SIAM J. Discrete Math.*, 24(1):146–157, 2010.
96. H. Tamaki. A directed path-decomposition approach to exactly identifying attractors of boolean networks. In *ISCIT 2010: International Symposium on Communications and Information Technologies*, pages 844–849, Oct 2010.

97. H. Tamaki. A polynomial time algorithm for bounded directed pathwidth. In *Graph-Theoretic Concepts in Computer Science*, volume 6986 of *Lecture Notes in Computer Science*, pages 331–342. Springer Berlin Heidelberg, 2011.
98. K. Truemper. *Matroid decomposition.* Academic Press, Inc., Boston, MA, 1992.
99. V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In V. Vovk, H. Papadopoulos, and A. Gammerman, editors, *Measures of Complexity*, pages 11–30. Springer, 2015.
100. E. Wanke. k-NLC graphs and polynomial algorithms. *Discrete Appl. Math.*, 54(2):251 – 266, 1994.
101. D.H. Younger. Graphs with interlinked directed circuits,. In *Midwest symposium on circuit theory 2*, pages XVI 2.1–XVI 2.7, 1973.
102. X. Zhu. Colouring graphs with bounded generalized colouring number. *Discrete Math.*, 309(18):5562–5568, 2009.