

Directed Width Measures and Monotonicity of Directed Graph Searching

Lukasz Kaiser

*CNRS & Université Sorbonne Paris Cité**,
lukaszkaiser@google.com

Stephan Kreutzer, Roman Rabinovich[†] and Sebastian Siebertz

Logic and Semantics, Technical University Berlin
 {stephan.kreutzer, roman.rabinovich, sebastian.siebertz}@tu-berlin.de

June 19, 2018

We consider generalisations of tree width to directed graphs, that attracted much attention in the last fifteen years. About their relative strength with respect to “bounded width in one measure implies bounded width in the other” many problems remain unsolved. Only some results separating directed width measures are known. We give an almost complete picture of this relation.

For this, we consider the cops and robber games characterising DAG-width and directed tree width (up to a constant factor). For DAG-width games, it is an open question whether the robber-monotonicity cost (the difference between the minimal numbers of cops capturing the robber in the general and in the monotone case) can be bounded by any function. Examples show that this function (if it exists) is at least $f(k) > 4k/3$ [KO08]. We approach a solution by defining *weak monotonicity* and showing that if k cops win weakly monotonically, then $O(k^2)$ cops win monotonically. It follows that bounded Kelly-width implies bounded DAG-width, which has been open since the definition of Kelly-width [HK08].

For directed tree width games we show that, unexpectedly, the cop-monotonicity cost (no cop revisits any vertex) is not bounded by any function. This separates directed tree width from D-width defined in [Saf05], refuting a conjecture in [Saf05].

1 Introduction

In the study of hard algorithmic problems on graphs, methods derived from structural graph theory have proved to be a valuable tool. The rich theory of special classes of graphs developed in this area

*Currently at Google Inc.

[†]This work was partially supported by the projects *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* and *Logic for Interaction (LINT)* of the *European Science Foundation*.

has been used to identify classes of graphs, such as classes of bounded tree width or clique-width, on which many computationally hard problems can be solved efficiently. Most of these classes are defined by some structural property, such as having a tree decomposition of low width, and this structural information can be exploited algorithmically.

Structural parameters such as tree width, clique-width, classes of graphs defined by excluded minors etc. studied in this context relate to undirected graphs. However, in various applications in computer science, directed graphs are a more natural model. Given the enormous success width parameters had for problems defined on undirected graphs, it is natural to ask whether they can also be used to analyse the complexity of hard algorithmic problems on digraphs. While in principle it is possible to apply the structure theory for undirected graphs to directed graphs by ignoring the direction of edges, this implies a significant information loss. Hence, for computational problems whose instances are directed graphs, methods based on the structure theory for undirected graphs may be less useful.

Reed [Ree99] and Johnson, Robertson, Seymour and Thomas [JRST01] initiated the development of a decomposition theory for directed graphs with the aim of defining a directed analogue of undirected tree width. They introduced the concept of *directed tree width* and showed that the k -disjoint paths problem and more general linkage problems can be solved in polynomial-time on classes of digraphs of bounded directed tree width. Following this initial proposal, several alternative notions of width measures for sparse classes of digraphs have been introduced, for instance *directed path width* (see [Bar06], initially proposed by Robertson, Seymour and Thomas), *D-width* [Saf05], *DAG-width* [BDH⁺12] and *Kelly-width* [HK08]. For each of these, algorithmic applications were given, for example in relation to linkage problems or a form of combinatorial games known as *parity games*. On the other hand, some other standard graph theoretical problems such as directed dominating set remain intractable on classes of digraphs of small width with respect to these measures. More recently, directed width parameters have been used successfully in areas outside core graph algorithmics, for instance in Boolean network analysis [Tam10], in the evaluation of simple regular path queries [BBG13], in the theory of verification in form of μ -calculus model-checking and solving parity games [BDH⁺12, HK08, BG05].

Despite the considerable interest these parameters have generated, not much is known about the relation between them. It is known that classes of bounded DAG-width, Kelly-width or D-width also have bounded directed tree width, making directed tree width the most general of these parameters. On the other hand, classes of digraphs of bounded directed path width also have bounded width in the other measures. However, it is still an open problem how DAG-width, Kelly-width or D-width relate to each other. The main structural contribution of this paper is to give an almost complete picture of the relationship between these width parameters with strict inequalities in most cases.

Digraph parameters such as directed tree width, DAG-width or Kelly-width are closely related to graph searching games, also called cops and robber games in this case. In a graph searching game, a number of cops tries to capture a robber on a graph or digraph. The robber occupies a vertex of the graph and so does each of the cops. The game is played in rounds where in each round the cops first announce their new position and then the robber can move to a different vertex of the graph to avoid capture. See below for details and see [FT08, Kre11] for recent surveys.

Variations of the game are obtained by restricting the moves of the cops and the robber in several ways. On every graph or digraph, the cops have a winning strategy that guarantees capturing the robber by using sufficiently many cops. The minimal number of cops on a digraph G that guarantees to capture the robber is a natural graph parameter and it turns out that the width measures discussed above are closely related to these parameters defined by suitable graph searching games.

An important concept in the context of graph searching games is *monotonicity*. Monotonicity

is a restriction on the strategies employed by the cops. We distinguish between cop- and robber-monotone cop strategies. Roughly speaking, a strategy is *cop-monotone* if the cops never revisit a vertex where they have been before, and it is *robber-monotone* if the set of vertices that the robber can occupy never increases during a play. Usually, monotone variants of graph searching games yield nice decompositions corresponding to directed or undirected width measures. For instance, a tree decomposition corresponds exactly to a cop-monotone winning strategy for the cops in a particular type of graph searching games.

The (*cop- or robber-*) *monotonicity problem* for variants of graph searching games—i.e. the problem whether on every graph or digraph the number of cops required to capture a robber with a cop- or robber-monotone strategy is the same as the number of cops required with an unrestricted strategy—has intensively been studied in the literature. For games that are not monotone, we call the number of extra cops required for a monotone strategy the *monotonicity cost* of the game variant. For graph searching games on undirected graphs this problem has been solved for most commonly used game variants and usually the games are monotone. For directed graphs, however, the situation is much less understood. It was shown in [JRST01, Adl07] that the games corresponding to directed tree width are not monotone. In [KO08] it was shown that also the games corresponding to Kelly- and DAG-width are non-monotone. More precisely, in [KO08] examples are exhibited where monotone strategies require at least $\frac{4}{3}k$ cops, but k cops suffice for an unrestricted strategy. However, all attempts to use the tricks facilitated in these examples to show that the monotonicity cost is in fact unbounded have failed so far.

Among the most important open problems in the area of cops and robber games at the moment is the question whether the monotonicity cost for the games corresponding to directed tree width, Kelly-width or DAG-width can be bounded by a constant factor, or by any function at all. This question is particularly interesting for DAG-width and Kelly-width games, as it was shown in [HK08] that bounding the monotonicity cost of these games would imply that DAG-width and Kelly-width are bounded by each other, i.e. a class of digraphs has bounded Kelly-width if, and only if, it has bounded DAG-width. The proof relies on translating monotone strategies in one type of game into (non-monotone) strategies of the other type of game.

For directed tree width games and robber-monotone strategies, the monotonicity question was answered in the affirmative in [JRST01]. It has been conjectured ([Saf05, Page 750]¹) that the cop-monotonicity cost should also be bounded for directed tree width games. Whether the monotonicity cost for DAG- and Kelly-width games is bounded is still open as well, despite considerable efforts in the community. These monotonicity problems are arguably the most important open problems in cops and robber games.

In this paper we give a negative answer to the cop-monotonicity problem for directed tree width games. We show that there is a class of digraphs where 4 cops have a winning strategy in the directed tree width game, but the number of cops required to win with a cop-monotone strategy is unbounded. We also make progress on the problem for DAG-width games. We introduce a weaker form of monotonicity, called *weak monotonicity*, and show that any weakly monotone strategy for k cops can be transformed into a robber-monotone strategy for k^2 cops. While this does not settle the monotonicity problem for DAG-width games completely, it constitutes significant progress towards this longstanding open problem in the following sense: in the known examples for non-monotonicity of DAG-width games, for instance in [KO08], the (unrestricted) strategies used by cops to win the game are actually weakly monotone in our sense. Hence, our result implies that these tricks cannot be used to show that there is no bound on the monotonicity cost for DAG-width games. Furthermore, as explained above, in [HK08] it is shown that monotone strategies in the DAG-width

¹Safari actually conjectures that D-width equals directed tree width which would imply cop-monotonicity.

or Kelly-width game can be translated into (non-monotone) strategies in the other type of games (with roughly the same number of cops). It turns out that the translation from Kelly-width games into DAG-width games actually translates a Kelly-strategy into a weakly monotone DAG-strategy and hence, by our result, this strategy can further be translated into a monotone strategy (with a quadratic number of cops). As a consequence, bounded Kelly-width implies bounded DAG-width, settling one of the open problems in the relation between different width measures. Finally, a winning cop strategy in weakly monotone DAG-width game induces a decomposition of the graph of small width, similar to tree width, DAG-width etc. In contrast to DAG decompositions, for which we do not know whether there exist “small” decompositions (i.e. of size polynomial both in $|G|$ and in the DAG-width of G), the new decompositions are essentially tress (rather than DAGs) of size in $O(|G|^2)$. Having a simpler structure than DAG decompositions they may be interesting by themselves both for algorithmical applications and for theoretical research on DAG-width. We remark that such a decomposition encodes in a compact way a DAG decomposition of width at most quadratically larger than the optimal one.

Organisation. The paper is organised as follows. In Section 2, after fixing some basic notation, we introduce graph searching games and prove our first main result, that the cop-monotonicity cost for directed tree width games are unbounded (Theorem 3.2). As a consequence, we separate directed tree width from D-width. Our monotonicity results for DAG-width are presented in Section 4 (see Theorem 4.32). In Section 5, we compare the various directed width measures with respect to the question whether classes of digraphs of bounded width in one measure have bounded width in another measure.

2 Preliminaries

We assume familiarity with basic concepts of directed graph theory and refer to [Die12] for background. The first part of this section serves to review and fix notation and terminology.

We denote the set of positive integers by \mathbb{N} and for $n \in \mathbb{N}$ we write $[n]$ for the set $\{1, \dots, n\}$. The prefix relation on words over some alphabet Σ is denoted by \sqsubseteq and its irreflexive version by \sqsubset . The lexicographical order is denoted by \preceq and its irreflexive variant by \prec . We write $\Sigma^{\leq n}$ for the set of words over Σ of length at most n .

All graphs in this paper are finite, directed and simple, i.e. they do not have loops or multiple edges between the same pair of vertices. Undirected graphs are directed graphs with symmetric edge relation, but we write $\{v, w\}$ for the undirected edge between v and w , i.e. for the pair of edges (v, w) and (w, v) . If G is a graph, then $V(G)$ denotes its set of vertices and $E(G)$ its set of edges. For a set $X \subseteq V(G)$ we write $G[X]$ for the subgraph of G induced by X and $G - X$ for $G[V(G) \setminus X]$. The set of vertices reachable from a vertex $v \in V(G)$ in G is denoted by $\text{Reach}_G(v)$. For a set $X \subseteq V(G)$ we write $\text{Reach}_G(X)$ for the set $\{w \in V(G) : \text{there is } v \in X \text{ such that } w \in \text{Reach}_G(v)\}$. For vertices $v, w \in V(G)$ we write $v \geq w$ (or $w \leq v$) if $w \in \text{Reach}_G(v)$ and $v > w$ (or $w < v$) if, additionally, $v \neq w$. A path is a sequence of vertices v_1, v_2, \dots with $(v_i, v_{i+1}) \in E(G)$ for all $i \geq 0$. A *strongly connected component* of a digraph G is a maximal subgraph C of G which is strongly connected, i.e. for all $u, v \in V(C)$ we have $u \in \text{Reach}_C(v)$ and $v \in \text{Reach}_C(u)$. All components considered in this paper will be strong and hence we simply write *component*. We write $e \sim v$ if vertex v is incident with edge e .

We write \bar{G} for the underlying undirected graph of G . The depth of an undirected, rooted tree is the the maximum number of vertices on a path from the root to a leaf of the tree. We write T_ℓ^d for the complete undirected tree of branching degree d and depth ℓ . We assume that the vertices of a rooted undirected tree of maximum branching degree d are words over $\{0, \dots, d - 1\}$, hence

the vertex set of T_ℓ^d is $\{0, \dots, d-1\}^{\leq \ell}$.

Graph searching games A graph searching game is played on a graph G by a team of cops and a robber. The robber and each cop occupy a vertex of G . Hence, a current game position can be described by a pair (C, v) , where C is the set of vertices occupied by cops and v is the current robber position. At the beginning the robber chooses an arbitrary vertex v and the game starts at position (\emptyset, v) . The game is played in rounds. In each round, from a position (C, v) the cops first announce their next move, i.e. the set $C' \subseteq V(G)$ of vertices that they will occupy next. Based on the triple (C, C', v) the robber chooses his new vertex v' . This completes a round and the play continues at position (C', v') . Variations of graph searching games are obtained by restricting the moves allowed for the cops and the robber. In all game variants considered here, from a position (C, C', v) , i.e. where the cops move from their current position C to C' and the robber is on v , the robber would have exactly the same choice of possible moves from any vertex in the component of $G - C$ containing v . We will therefore describe game positions by a pair (C, R) , or a triple (C, C', R) , where C, C' are as before and R is a component of $G - C$. We call R the *robber component*.

Formally a graph searching game on a graph G is specified by a tuple $\mathcal{G} = (\text{Pos}(G), \text{Mvs}(G), \text{Mon})$, where $\text{Pos}(G)$ describes the set of possible positions, $\text{Mvs}(G)$ the set of legal moves and Mon specifies the monotonicity criteria used. In all game variants considered here, the set $\text{Pos}(G)$ of positions is $\text{Pos}_c \cup \text{Pos}_r$ where $\text{Pos}_c = \{(C, R) : C \subseteq V(G), R \subseteq V(G) \text{ is a component of } G - C\}$ are cop positions and $\text{Pos}_r = \{(C, C', R) : C, C' \subseteq V(G) \text{ and } R \subseteq V(G) \text{ is a component of } G - C\}$ are robber positions.

As far as legal moves are concerned, we distinguish between two different types of games, called *reachability* and *component* games. In both cases the cops moves are $\text{Mvs}_c(G) := \{(C, R), (C, C', R) : (C, R) \in \text{Pos}_c, (C, C', R) \in \text{Pos}_r\}$. The difference is in the definition of the set of possible robber moves.

Reachability game In the *reachability game*, $\text{Mvs}(G)$ are defined as $\text{ReachMvs}(G)$, where $\text{ReachMvs}(G) := \text{Mvs}_c(G) \cup \{(C, C', R), (C', R') : (C, C', R) \in \text{Pos}_r, (C', R') \in \text{Pos}_c \text{ and } R' \text{ is a component of } G - C' \text{ reachable from a vertex in } R \text{ by a directed path in } G - (C \cap C')\}$. I.e. the robber can run along any directed path in the digraph which does not contain a cop from $C \cap C'$ (i.e. one that remains on the board).

Component game In the *component game*, we define $\text{Mvs}(G)$ as $\text{CompMvs}(G)$, where $\text{CompMvs}(G) := \text{Mvs}_c(G) \cup \{(C, C', R), (C', R') : (C, C', R) \in \text{Pos}_r, (C', R') \in \text{Pos}_c \text{ and } R' \text{ is a component of } G - C' \text{ such that } R, R' \text{ are contained in the same component of } G - (C \cap C')\}$. I.e. in the component game, the robber can only run to a new vertex within the strongly connected component of $G - (C \cap C')$ that contains his current position.

Monotonicity The component Mon describes the monotonicity condition and is a set of finite plays. The cops win all plays $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \dots$ in Mon where $R_i = \emptyset$ for some i and the robber wins all other plays. Usually Mon describes cop- or robber-monotonicity; the latter is defined differently in the component game and in the reachability game: $\text{Mon} \subseteq \text{cm}(G) \cup \text{rm}_{\text{comp}}(G) \cup \text{rm}_{\text{reach}}(G)$. A play $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \dots$ is in $\text{cm}_{\text{reach}}(G)$, called *cop-monotone*, if for all $i, j, k \geq 0$ with $i < j < k$ we have $C_i \cap C_k \subseteq C_j$, i.e. cop-monotonicity means that the cops never reoccupy vertices. A play $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \dots$ is in $\text{rm}_{\text{reach}}(G)$, called *robber-monotone*, if for all i , $\text{Reach}_{G-(C_i \cap C_{i+1})}(R_{i+1}) \subseteq \text{Reach}_{G-(C_i \cap C_{i+1})}(R_i)$, i.e. once the

robber cannot reach a vertex, he won't be able to reach it forever. Finally, a play is in $\text{rm}_{\text{comp}}(G)$, also called *robber-monotone*, if $R_{i+1} \subseteq R_i$ for all i .

A strategy for the cops is *cop-* or *robber-monotone* if all plays consistent with that strategy are cop- or robber-monotone, respectively.

By combining reachability or component games with monotonicity conditions we obtain a range of different graph searching games. It follows immediately from the definition that on every digraph the cops have a winning strategy in each of the graph searching games defined above by simply placing a cop on every vertex. For a given digraph G , we are therefore interested in the minimal number k such that the cops have a winning strategy in which no cop position C_i contains more than k vertices.

Definition 2.1. For any digraph G , we define

- $cn_G(\text{dtw})$ as the minimal number of cops needed to win $(\text{Pos}(G), \text{CompMvs}(G), \text{Mon})$ where Mon is the set of all finite plays,
- $cn_G(\text{cmdtw})$ as the minimal number of cops to win $(\text{Pos}(G), \text{CompMvs}(G), \text{Mon} = \text{cm}_{\text{comp}}(G))$,
- $cn_G(\text{DAG})$ as the minimal number of cops needed to win $(\text{Pos}(G), \text{ReachMvs}(G), \text{Mon} = \text{cm}_{\text{reach}}(G))$.

It follows immediately from the definitions that, for all digraphs G , $cn_G(\text{dtw}) \leq cn_G(\text{DAG})$ and $cn_G(\text{dtw}) \leq cn_G(\text{cmdtw})$. The number $cn_G(\text{cmdtw}) - cn_G(\text{dtw})$ is called the *cop-monotonicity cost* for the component game on G . Robber-monotonicity cost as well as monotonicity cost for other game variants are defined analogously.

3 Strong non-cop-monotonicity of directed tree width

Directed tree width can be characterised up to a constant factor by the directed tree width game.

Theorem 3.1 ([JRST01]). *The directed tree width of a graph G and $cn_G(\text{dtw})$ are within a constant factor of each other.*

Directed tree width is defined by directed tree decompositions (in [JRST01] called *arboreal decompositions*), see ?? for a definition. Such a decomposition can be viewed as a description of a robber-monotone winning strategy for the cops. The proof of Theorem 3.1 essentially shows that a winning strategy for k cops can be transferred in a directed tree decomposition of width, roughly, at most $3k$ and hence in a robber-monotone winning strategy for approximately $3k$ cops. It follows that the robber-monotonicity cost for directed tree width is bounded by a constant factor.

One would expect that the cop-monotonicity cost can be bounded similarly by a slowly growing function. However, the following theorem shows that the cop-monotonicity cost for directed tree width cannot be bounded by any function at all.

Theorem 3.2. *There is a class $\{G_n : n > 2\}$ of graphs such that for all n , $cn_{G_n}(\text{dtw}) \leq 4$ and $cn_{G_n}(\text{cmdtw}) \geq n$.*

Proof. Let $n > 2$. We inductively define a sequence of graphs G_n^m and sets marked vertices $M(G_n^m) \subseteq V(G_n^m)$ for $m \in \{1, \dots, n+1\}$. We then define G_n as G_n^{n+1} .

First G_n^1 is an edgeless graph with a single vertex and $M(G_n^1) = V(G_n^1)$, i.e. the vertex of G_n^1 is marked. Assume that $(G_n^m, M(G_n^m))$ has been constructed. Recall that T_ℓ^d denotes a complete undirected tree of branching degree d and depth ℓ . One part of G_n^{m+1} is a copy of T_{n+2}^{n+1} , which has $(n+1)^{n+2}$ leaves v_s for $s \in \{1, \dots, (n+1)^{n+2}\}$. The graph G_n^{m+1} is the disjoint union of T_{n+2}^{n+1} and

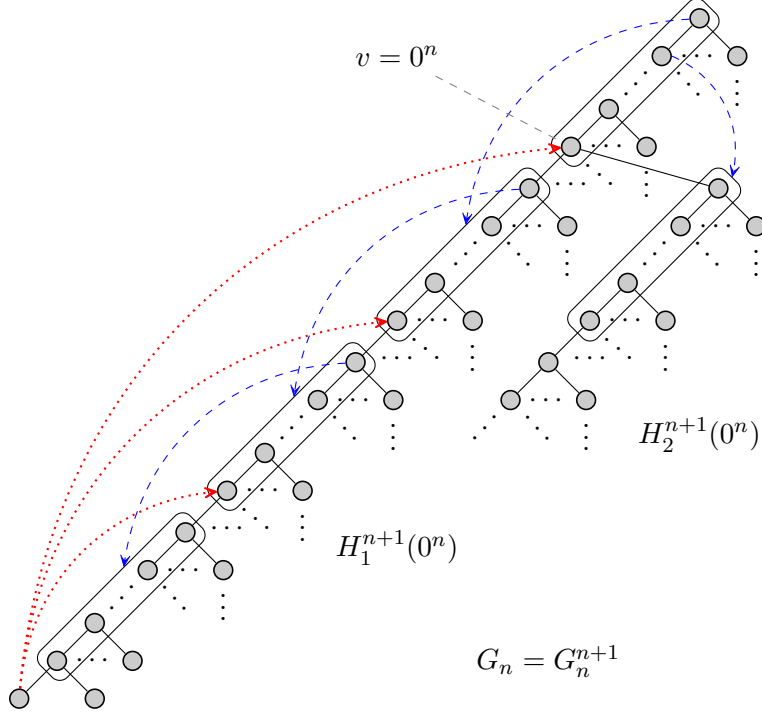


Figure 1: $cn_{G_n}(\text{dtw}) = 4$, but the robber wins against n cop-monotone cops. Only the left-most branch of G_n and the upper part of the left-most branch of $G_2^n(0^n)$ is shown.

$n \cdot (n+1)^{n+2}$ copies $H_j^{m+1}(v_s)$ of G_n^m where $j \in \{1, \dots, n\}$ and $s \in \{1, \dots, (n+1)^{n+2}\}$ plus some additional edges which we describe next. We denote the subgraph T_{n+2}^{n+1} of G_n^{m+1} by $T(G_n^{m+1})$.

For every leaf $v \in \{v_s : 1 \leq s \leq (n+1)^{n+2}\}$ there is an undirected edge from v to the root of $H_i^{m+1}(v)$. Let $x_i^{m+1}(v)$ be the i th vertex on the path from the root of $T(G_n^{m+1})$ to v . For all leaves v of $T(G_n^{m+1})$ and all $1 \leq i \leq n$ we add directed edges from $x_i^{m+1}(v)$ to all marked vertices $M(H_i^{m+1}(v))$ of $H_i^{m+1}(v)$. Finally, for all leaves v of $T(G_n^{m+1})$ and all leaves of $H_i^{m+1}(v)$ we add a directed edge to v . We define $M(G_n^{m+1}) := V(T(G_n^{m+1}))$. The graph G_n is schematically shown in ?? (edges without arrows mean edges in both directions).

Let us describe a non-cop-monotone winning strategy for 4 cops on G_n . Observe that $G_n = G_n^{n+1}$ is an undirected tree with additional edges that connect only vertices of the same branch. In particular, for each subgraph $H_j^i(v)$, if the robber is in $H_j^i(v)$ and the cops block the root of $T(H_j^i(v))$ and $x_j^{i+1}(v)$, then the robber may not leave $H_j^i(v)$ as he cannot reenter $H_j^i(v)$.

We show that in each play of the game there is a unique sequence

$$G_n^{n+1}, H_{j(n)}^n(v_n), H_{j(n-1)}^{n-1}(v_{n-1}), \dots, H_{j(1)}^1(v_1)$$

of subgraphs in which the cops are placed and such that the robber is captured on the unique vertex of $H_{j(1)}^1(v_1)$.

Assume that the root of $T(G_n^{n+1})$ is occupied by a cop. Then two additional cops can play in a top-down manner in $T(G_n^{n+1})$ following the robber to his tree branch until the robber is forced out of $T(G_n^{n+1})$ into some $H_{j(n)}^n(v)$ for some leaf v of $T(G_n^{n+1})$. Define $v_n := v$. The cops now occupy in a first step v_n , the root of $T(H_{j(n)}^n(v_n))$ and $x_{j(n)}^{n+1}(v_n)$. In a second step, they release the cop from v_n and from the root of $T(G_n^{n+1})$, as these vertices are no longer available for the robber.

Similarly, assume that the root of $T(H_{j(i)}^n(v_i))$ and $x_{j(i)}^{i+1}(v_i)$ are occupied by cops. As above, two additional cops can play in a top-down manner in $T(H_{j(i)}^n(v_i))$ following the robber to his tree branch until the robber is forced out of $T(H_{j(i)}^n(v_i))$ into some $H_{j(i-1)}^{i-1}(v)$ for some leaf v of $T(H_{j(i)}^n(v_i))$. Define $v_{i-1} := v$. At this point of time, three cops are placed on the graph, one on v_{i-1} , one on the root of $T(H_{j(i)}^n(v_i))$ and one on $x_{j(i)}^{i+1}(v_i)$. The cops now first occupy with an additional cop the root of $T(H_{j(i-1)}^n(v_{i-1}))$. They can now release the cop from v_{i-1} which they place on $x_{j(i-1)}^i(v_{i-1})$. Finally they may release the cop from $x_{j(i)}^{i+1}(v_i)$ and thereby establish the induction hypothesis for $i - 1$.

In this way the robber is captured at the latest on the single vertex of $H_{j(1)}^1(v_1)$.

Now we construct a robber strategy that wins against all cop-monotone strategies for n cops if $n > 2$. For a vertex v and subtree T of G_n we say that T is a subtree of v if the root of T is a direct successor of v . The robber resides on a vertex of $T(G_n)$ that has the least distance to the root of G_n as long as this is possible. When a cop occupies his vertex v the robber proceeds to a directed successor of v such that the subtree of v is cop free. Such a successor always exists due to the high branching degree of G_n . When the robber reaches a leaf w_n of $T(G_n)$, every vertex on the path from the root of G_n to w_n has been occupied by a cop. As the length of the path is greater than the number of cops, there is a vertex $x_{i_n}^n(w_n)$ that has been left by a cop. When a cop occupies w_n , the robber goes to $G_{i_n}^n(w_n)$. Now on $G_{i_n}^n(w_n)$ (which is isomorphic to G_n^{n-1}) the robber plays in the same way as on G_n and so on recursively for each m on $G_{i_m}^m(w_m)$. Note that until the robber is captured, there is a path from this vertex to a leaf of G_n and then to all already chosen w_j .

Consider a position when the robber arrives at a leaf v of G_n and a cop is landing on this vertex. Then at most $n - 1$ cops are on the graph and there is some j such that there is no cop in $T(G_{i_j}^j(w_j))$. Thus there is a cop free path from v to w_j , then to $x_{i_j}^j(w_j)$ within $T(G_{i_j}^j(w_j))$ and then via $x_{i_{j-1}}^{j-1}(w_{j-1})$, $x_{i_{j-2}}^{j-2}(w_{j-2})$, \dots , $x_{i_2}^2(w_2)$ back to v . Note that all those x -vertices are not occupied by cops by construction. Thus the robber can return to w_j and play from w_j as before. In this way the robber will never be captured. \square

4 Towards monotonicity of the DAG-width game

DAG-width is usually defined by means of DAG decompositions, similar to tree decompositions. For our purposes a game theoretic characterisation of the DAG-width of a graph G as $cn_G(\text{DAG})$ is more useful, and we take it as a definition and refer to the corresponding game as DAG-width game. See [BDH⁺12] for details.

As explained in the introduction, one of the most important open problems in graph searching is the question whether cop- and robber-monotonicity cost of DAG-width games is bounded by any function. Towards this goal, we introduce two new constraints for the DAG-width game, *weak monotonicity* and a technical notion of *shyness*.

Weak monotonicity relaxes the winning condition for the cops, so that they win more plays. For a digraph G we define $\text{wm}(G)$ as the set of all finite plays $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \dots$ such that the following condition is satisfied. For all i let $c(i) := C_{i+1} \cap R_i$ be the cops which move into the component of $G - C_i$ currently used by the robber. We call these cops the *chasers*. All other cops being placed, i.e. the cops in $(C_{i+1} \setminus C_i) \setminus c(i)$ are *guards*. The play $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \dots$ is *weakly monotone* if for all i and all j with $j < i$, no vertex in $c(j)$ is reachable by a directed path from any vertex in R_i in $G - (C_i \cap C_{i+1})$. That is, for weak monotonicity we only require monotonicity in the cops that are used to shrink the robber space but not in the cops placed outside of the component to block the paths to previous cop positions. The set $\text{wm}(G)$ is the set of all

weakly monotone plays on G .

In a shy robber game, the robber can never leave his strong component and therefore has the same set of possible moves as in the directed tree width game. However, and this is crucial, the monotonicity conditions are defined based on directed reachability. I.e. the robber can destroy monotonicity if there is a directed path from his current position to a forbidden vertex. We use the shy robber games to consider the case in the DAG-width game when the robber decides never to change his component, even if he could do this: we just enforce him to stay in his component. Of course, this does not restrict his ability to incur non-monotonicity outside of his component.

Based on weak monotonicity we can now define the following variants of the DAG-width game.

- The *weakly monotone game* is the game defined by $(\text{Pos}(G), \text{ReachMvs}(G), \text{Mon} = \text{wm}(G))$.
- The *weakly monotone shy (robber) game* is the game $(\text{Pos}(G), \text{CompMvs}(G), \text{Mon} = \text{wm}(G))$.
- Finally, the *strongly monotone shy (robber) game* is the game $(\text{Pos}(G), \text{CompMvs}(G), \text{Mon} = \text{cm}_{\text{reach}}(G))$.

We write $cn_G(\text{shyDAG})$, $cn_G(\text{wmDAG})$ and $cn_G(\text{wmshyDAG})$ for the minimal number k of cops needed to win the corresponding game. The following inequalities are immediate consequences of the definitions:

$$\begin{aligned} cn_G(\text{wmshyDAG}) &\leq cn_G(\text{shyDAG}) \leq cn_G(\text{DAG}); \\ cn_G(\text{wmshyDAG}) &\leq cn_G(\text{wmDAG}) \leq cn_G(\text{DAG}). \end{aligned}$$

The following theorem is our main result in this section.

Theorem 4.1. *If k cops capture a shy robber in a weakly monotone way, then $18k^2 + 3k$ cops capture a non-shy robber in a strongly monotone way.*

Hence the weak monotonicity cost is bounded by a quadratic function. To prove that the (strong) monotonicity cost is bounded, it suffices to show that for some function $f: \mathbb{N} \rightarrow \mathbb{N}$ a winning strategy for k cops in the DAG-width game without any monotonicity constraints induces a winning strategy for $f(k)$ cops against a shy robber in the weakly monotone game. If this is not true, Theorem 4.32 shows that the examples from [KO11] cannot be used to prove this, as the winning strategies used there are weakly monotone.

4.1 Blocking and the blocking order

Considering the robber whose influence reaches further than his current component (either because he can leave it or by the weak monotonicity) we study the properties of cops blocking certain positions from the robber. This is crucial for placing the guards. To make this formal, we define blocking sets and an order on them, so that we can speak about minimal blocking sets.

Definition 4.2 (Blocking). Let R , M and X be sets of vertices of a graph G . We say that X *blocks* $R \rightarrow M$ if $X \cap R = \emptyset$ and every path from R to M in G contains a vertex in X . When R and M are clear from the context, we simply say that X is a *blocker*.

Below, we formulate a few basic properties of blocking. Note that the graph G can, of course, have cycles. For some $X \subseteq V(G)$ and a path P we say that P is X -free if $X \cap P = \emptyset$.

Lemma 4.3. *If X blocks $R \rightarrow M$ and Y blocks $R \rightarrow X$, then Y blocks $R \rightarrow M$.*

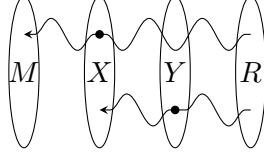


Figure 2: Illustration for Lemma 4.3

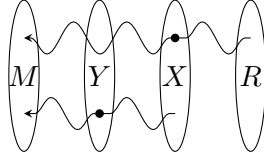


Figure 3: Illustration for Lemma 4.4

Proof. Assume to the contrary that Y does not block $R \rightarrow M$, so there is a Y -free path P from R to M , see Figure 2. Since X blocks $R \rightarrow M$, there is a vertex v on this path which is in X . But then the prefix up to v of the path P is a Y -free path from R to X , a contradiction to the assumption that Y blocks $R \rightarrow X$. \square

Lemma 4.4. *If X blocks $R \rightarrow M$ and Y blocks $X \rightarrow M$, then Y blocks $R \rightarrow M$.*

Proof. Assume to the contrary that Y does not block $R \rightarrow M$, so there is a Y -free path P from R to M , see Figure 3. Since X blocks $R \rightarrow M$, there is a vertex v on this path which is in X . But since Y blocks $X \rightarrow M$, there must be a vertex in Y on the suffix of P starting from v . This is a contradiction, as P was assumed to be Y -free. \square

The following lemma is not used directly in further proofs, but serves as an illustration of the techniques that will be used later.

Lemma 4.5. *If A_1 blocks $X \rightarrow M$ and X blocks $A_2 \rightarrow M$, then $A_1 \setminus A_2$ blocks $X \rightarrow M$.*

Proof. The situation is illustrated in Figure 4. Let $A = A_1 \setminus A_2$. Assume to the contrary that A does not block $X \rightarrow M$, so there is an A -free path P from X to M . Since A_1 blocks $X \rightarrow M$, there must be a vertex on this path which is in A_1 . Let w be the last such vertex on P and note that $w \in A_1 \cap A_2$ since P is $(A_1 \setminus A_2)$ -free. But, as X blocks $A_2 \rightarrow M$, there must be a vertex $u \in X$ on the part of P strictly after w . The suffix of P starting from u is then a path connecting X with M and avoiding A_1 (by the choice of w), a contradiction to our assumption that A_1 blocks $X \rightarrow M$. \square

We formulate the following simple observation as a lemma.

Lemma 4.6. *Let X be an inclusion-minimal set that blocks $R \rightarrow M$. Then for each $v \in X$ there is a path P from R to M such that $P \cap X = \{v\}$.*

The following is our main technical lemma on blocking.

Lemma 4.7. *Let A and B block $R \rightarrow M$. Then*

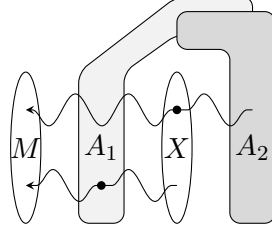


Figure 4: Illustration for Lemma 4.5

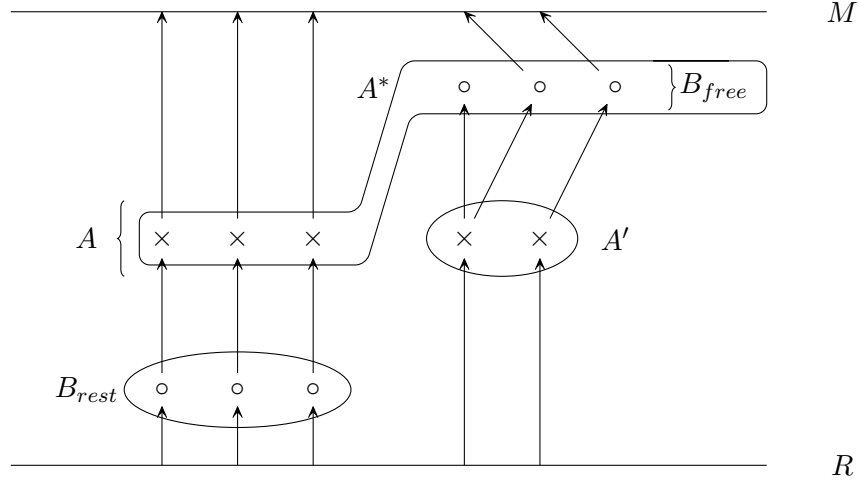


Figure 5: Situation in the proof of Lemma 4.7.

- (1) A blocks $B \rightarrow M$ or
- (2) there exists a set $B^* \subseteq A \cup B$ with $|B^*| < |B|$ which blocks $R \rightarrow B, M$, or
- (3) there exists a set $A^* \subseteq A \cup B$ with $|A^*| \leq |A|$ which blocks $A, B, R \rightarrow M$.

Proof. We partition the set B into elements B_{free} from which M is reachable via paths which avoid A and the rest, called B_{rest} , so A blocks $B_{rest} \rightarrow M$. Moreover, we let A' be any *inclusion-minimal* subset of A such that the set $A' \cup B_{rest}$ blocks $R \rightarrow B_{free}$. Observe that if $A' = \emptyset$, then either $B_{free} = \emptyset$, in which case A already blocks $B \rightarrow M$ and we are done by ??, or B_{rest} blocks $R \rightarrow B_{free} \neq \emptyset$ and thus $R \rightarrow M$, as B blocks $R \rightarrow M$, in which case $B^* = B_{rest}$ is the set we require in ??. We will now consider the case when $A' \neq \emptyset$. This situation is depicted in Figure 5. First observe a simple fact.

Claim 1. *For every $a \in A'$ there is a $B \cup (A' \setminus \{a\})$ -free path from R to a .*

Proof. As $B_{free} \cup A'$ blocks $B_{free} \rightarrow M$, by Lemma 4.6, there is a path P from B_{free} to M with $P \cap (B_{rest} \cup A') = \{a\}$. The suffix of P from the last occurrence of a is a path with the desired properties: it never visits $B_{rest} \cup A'$ and thus also never visits B_{free} , as $B_{rest} \cup A'$ blocks $B_{free} \rightarrow M$. \dashv

We will consider two cases.

Case (i): $|A'| < |B_{free}|$.

Define $B^* = A' \cup B_{rest}$ – it is smaller than B and blocks $B \rightarrow M$, which is ??.

Case (ii): $|A'| \geq |B_{free}|$.

Define $A^* = B_{free} \cup (A \setminus A')$. We claim that A^* blocks $R \rightarrow M$. Assume to the contrary that there is a path P from R to M which avoids A^* . Since it avoids B_{free} and B blocks $R \rightarrow M$, this path must go through B_{rest} . But, since A blocks $B_{rest} \rightarrow M$, it must visit A after each visit of B_{rest} . Let $a \in A$ be the last such vertex. Since the path omits A^* , we have $a \in A'$. By Claim 1, there is a $B \cup (A' \setminus \{a\})$ -free path P' from R to a . Concatenating the P' and the suffix of P from a we get a B -free path from R to M , which contradicts the fact that B blocks $R \rightarrow M$. Thus A^* blocks M from R .

Now we show that A^* blocks $A, B \rightarrow M$. First, A^* blocks $A' \rightarrow M$, otherwise there is a B_{free} -free path P_0 from A' to M . Let a' be the last vertex from A' on P_0 . According to Claim 1, there is a B -free path P_1 from R to a' . The concatenation of P_1 and the suffix of P_0 from a' is B -free path from M to R , which contradicts the assumption that B blocks $R \rightarrow M$. It follows that A^* blocks $R \rightarrow A$.

To see that A^* also blocks $R \rightarrow B$, note that A blocks $B_{rest} \rightarrow M$ and $B_{free} \subseteq A^*$. Finally, $|A^*| \leq |A|$ since B_{free} is disjoint with A by its definition. \square

A preorder on blocking sets

The blocking relation induces a partial preorder on sets blocking $R \rightarrow M$.

Definition 4.8. Let A and B block $R \rightarrow M$ in G . We write $A \prec_M^R B$ if either $|A| < |B|$, or $|A| = |B|$ and A blocks $B \rightarrow M$.

Intuitively, the second condition for $A \prec_M^R B$ means that A blocks from R as few vertices in addition to M as possible. From Lemma 4.7 we immediately obtain the following

Corollary 4.9. *If A is \prec_M^R -minimal and B blocks $R \rightarrow M$, then*

- (1) A blocks $B \rightarrow M$ or
- (2) there exists a set B^* with $|B^*| < |B|$ which blocks $R \rightarrow B$ and $R \rightarrow M$.

Proof. Assume that the case (3) from Lemma 4.7 holds. Let A^* be a set with $|A^*| \leq |A|$ that blocks $A \rightarrow M$, $B \rightarrow M$, and $R \rightarrow M$. In particular, A^* blocks $A \rightarrow M$, so A is not \prec_M^R -minimal. \square

From Lemma 4.4 we obtain that \prec_M^R is transitive, so it is a preorder. Moreover, Corollary 4.9 allows us to show the following lemma.

Lemma 4.10. *There is a unique minimal element with respect to \prec_M^R .*

Proof. Assume that there exist two distinct \prec_M^R -minimal sets A and B that block $R \rightarrow M$, then neither $A \prec_M^R B$ nor $B \prec_M^R A$. That means, $|A| = |B|$. Consider the cases given by Lemma 4.7. In ??, A blocks $B \rightarrow M$, so $A \prec_M^R B$ and B is not minimal. In ??, $B^* \prec_M^R B$, so B is not minimal as well. In ??, $A^* \prec_M^R A$, so A is not minimal. \square

We will denote the minimal element with respect to \prec_M^R , the *minimal blocker* of $R \rightarrow M$, by $\text{mb}(R, M)$.

During the game, it is important to us how minimal blocking sets behave when R becomes smaller or M becomes bigger, especially in comparison to possible previous blocking sets. The next lemma allows to compare a minimal set to a possibly non-minimal one.

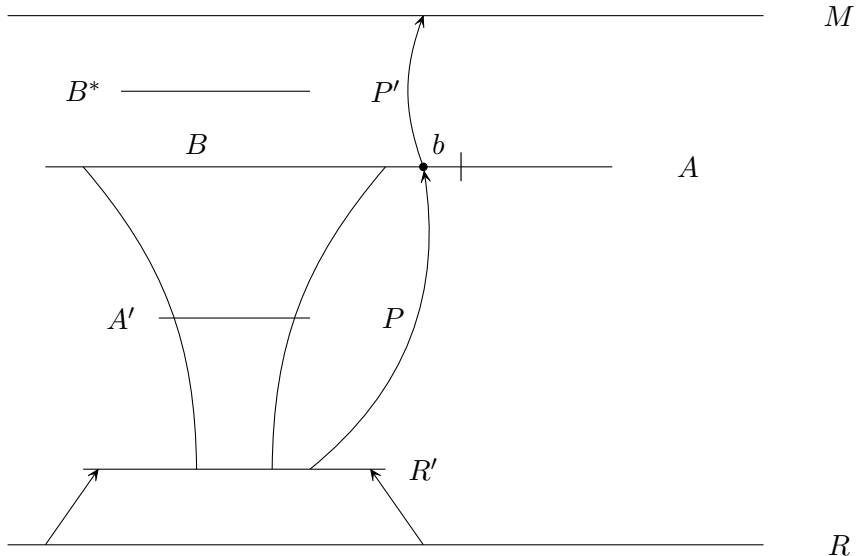


Figure 6: Illustration to Lemma 4.11.

Lemma 4.11. *Let A be $\text{mb}(R, M)$, let $R' \subseteq \text{Reach}_{G-A}(R)$ and, for the new R' , let A' be $\text{mb}(R', M)$. Then A' blocks $R' \rightarrow A$.*

Proof. Let $B = \text{Reach}_G(R') \cap A$. It suffices to prove that A' blocks $R' \rightarrow B$. Apply Lemma 4.7 with Corollary 4.9 to B (as A), A' (as B), R' (as R) and M (as M). Consider $??$. Assume that there is a path P from R' to a vertex $b \in B$ that avoids A' . By Lemma 4.6 there is an $A \setminus \{b\}$ -free path P' from b to M . Concatenating P with the suffix P' from the last occurrence of b in that path we get a path from R' to M . As A' blocks $R' \rightarrow M$, this path goes through A' . As P does not, there is some $a' \in P' \cap A'$. As B blocks $A' \rightarrow M$, P' visits B after a' . As $P' \cap B = \{b\}$, P' visits b after a' , but by definition of P' , it contains b only as the first vertex, which is not $a' \in A'$, because $a' \in P$ and P is A' -free.

In Case (3), some set B^* with $|B^*| < |B|$ blocks $A', B, R' \rightarrow M$, but A is minimal, so if $B^* \neq B$, we can replace B by B^* in A to get a blocker $R' \rightarrow M$ (because B^* blocks $B \rightarrow M$) with $B^* \cup (A \setminus B) \prec_{M'}^{R'} A$. This is impossible, since A is minimal, so $B^* = B$. Thus B blocks $A' \rightarrow M$ and we have Case (1). \square

A similar result is obtained for the case when M grows.

Lemma 4.12. *Let A be $\text{mb}(R, M)$, let $M' \supseteq M$ and, for the new M' , let A' be $\text{mb}(R, M')$. Then A' blocks $R \rightarrow A$ and A blocks $A' \rightarrow M$.*

Proof. Consider $B = \{a \in A' \mid M \cap \text{Reach}_G(a) \neq \emptyset\}$ (i.e. B is the part of A' from which M is reachable) and apply Lemma 4.7 with Corollary 4.9 to A and B . Case (2) is impossible (replace B by B^* in A' , then $B^* \cup (A' \setminus B)$ blocks $R \rightarrow M'$ and $B^* \cup (A' \setminus B) \prec_{M'}^R A'$, but A' is $\prec_{M'}^R$ -minimal), so we have Case (1), i.e. A blocks $B \rightarrow M$. Then A blocks $A' \rightarrow M$, which shows the second statement.

Assume that there exists a path P' from R to A that avoids B (and thus A'). Let a be the last vertex on this path. By Lemma 4.6, there is a path P from R to M whose intersection with A is $\{a\}$. Consider the suffix S of P from the last appearance of a . It does not visit B , as A blocks $A' \rightarrow M$ and thus also $B \rightarrow M$, so after each visit of B , S would visit A , but P intersects A only

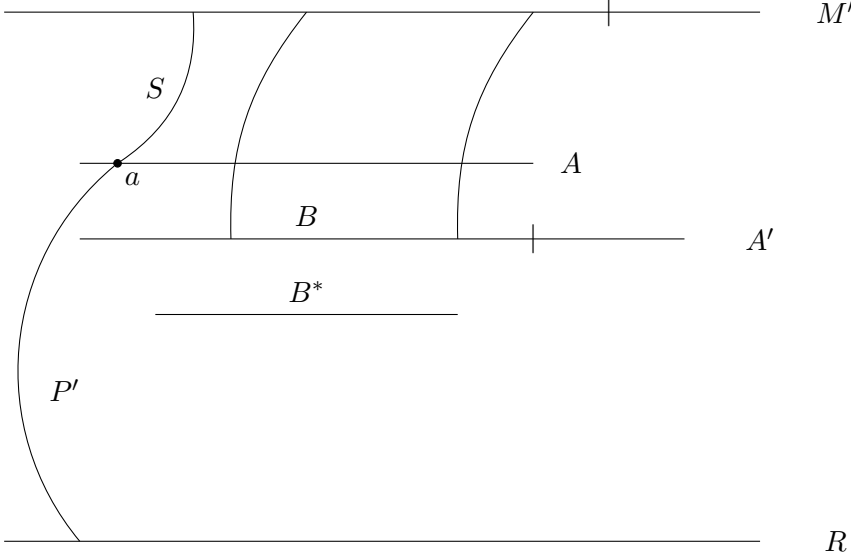


Figure 7: Illustration to Lemma 4.12.

in a and S does not visit a by definition. Concatenating P' with S we get a path from R to M that avoids A' , which is impossible, as A' blocks $R \rightarrow M$. \square

4.2 Minimally blocking strategies

In this section we concentrate on a specific kind of strategies for the cops in the shy weakly monotone game, namely ones that move chasers in the same way, but whose guarding moves are placing the cops on the minimal blocking set.

Let σ be a strategy for the cops in the weakly monotone shy game on G . We define the *minimally blocking strategy* σ_{mb} , derived from σ , for possibly more cops than σ , by induction on the length of play prefixes. This construction also provides a function that maps each history π consistent with σ to a history π_{mb} of the same length as π that is consistent with σ_{mb} such that the following invariants hold. Let $\pi = (C_0, R_0), (C_0, C_1, R_0), \dots, P$ where $P = (C_i, R_i)$ or $P = (C_i, C_{i+1}, R_i)$, and let $\sigma_{\text{mb}} = (C_0^{\text{mb}}, R_0^{\text{mb}}), (C_0^{\text{mb}}, C_1^{\text{mb}}, R_0^{\text{mb}}), \dots, P_{\text{mb}}$ where $P_{\text{mb}} = (C_i^{\text{mb}}, R_i^{\text{mb}})$ or $P_{\text{mb}} = (C_i^{\text{mb}}, C_{i+1}^{\text{mb}}, R_i^{\text{mb}})$.

- (i) $R_i = R_i^{\text{mb}}$ and $M(\pi) = M(\pi_{\text{mb}})$,
- (ii) after a cop move, i.e. if $P = (C_i, C_{i+1}, R_i)$, we have $C_{i+1} \cap R_i = C_{i+1}^{\text{mb}} \cap R_i^{\text{mb}}$ (the chasers are placed in the same way),
- (iii) after a cop move, $\text{mb}(M(\pi_{\text{mb}}), R_i) \subseteq C_{i+1}^{\text{mb}}$ (the cops occupy the minimal blocker).

Let $\pi[i]$ be the prefix of π up to position (C_i, R_i) . In the first move of the cops, if $\sigma(\pi([0])) = \sigma((\emptyset, R_0)) = (\emptyset, C_1, R_0)$, with chasers $C_1^c = C_1 \cap R_0$, then we set

$$\sigma_{\text{mb}}(\pi_{\text{mb}}[0]) = (\emptyset, C_1^c \cup \text{mb}(R_0, C_1^c), R_0),$$

i.e. we put the chasers and the minimal blocker. Obviously, the invariants hold.

We turn to the inductive step. If the robber is the next to move, then the last position in π has the form (C_i, C_{i+1}, R_i) and the next move is to (C_{i+1}, R_{i+1}) where R_{i+1} is a strongly connected

component of $(G - C_{i+1})$ with $R_{i+1} \cap R_i \neq \emptyset$. As the cops play according to a robber-monotone strategy, we even have $R_{i+1} \subseteq R_i$. By the inductive hypothesis, the last position in π_{mb} has the form $(C_i^{\text{mb}}, C_{i+1}^{\text{mb}}, R_i^{\text{mb}})$ and $C_{i+1}^{\text{mb}} \cap R_i^{\text{mb}} = C_{i+1} \cap R_i$. Thus, *in the shy game*, the robber has exactly the same choices for R_{i+1}^{mb} from this position as from the end of π . Therefore we can extend π_{mb} by $(C_{i+1}^{\text{mb}}, R_{i+1}^{\text{mb}})$ and the conditions (i)–(iii) are satisfied.

Consider now the case that the cops are to move at the end of π , i.e. the last position in π has the form (C_i, R_i) . Let $\sigma(\pi[i]) = (C_i, C_{i+1}, R_i)$, with chasers $C_{i+1}^c = C_{i+1} \cap R_i$. We set

$$\sigma_{\text{mb}}(\pi[i]) = (C_i^{\text{mb}}, C_{i+1}^c \cup \text{mb}(R_i^{\text{mb}}, M(\pi[i])) \cup \text{mb}(R_{i-1}^{\text{mb}}, M(\pi[i-1])), R_i^{\text{mb}}).$$

Intuitively, we place the same chasers as σ , occupy the current minimal blocker and additionally the previous minimal blocker. It is straightforward to see that all conditions (i)–(iii) are satisfied.

The construction above defines the strategy σ_{mb} and the corresponding plays, but we are, of course, interested in strategies that are still weakly robber-monotone. Strategy σ_{mb} is even strongly robber-monotone.

Lemma 4.13. *Let σ be a strategy for cops in $cn_G(\text{wmshyDAG})$. Then the strategy σ_{mb} is robber-monotone.*

Proof. Every blocking set $\text{mb}(R_i, M(\pi[i]))$ blocks $R_i \rightarrow M(\pi[i])$, so vertices that have been occupied by the chasers are not available for the robber. By Lemma 4.11 and Lemma 4.12 previous blocking sets are blocked by later blocking sets as R_i becomes smaller and $M(\pi[i])$ becomes bigger. \square

Let us calculate the number of cops used by σ_{mb} .

Lemma 4.14. *Let σ be a winning strategy for k cops in the weakly monotone shy game on G . Then σ_{mb} is a winning strategy for $3k$ cops in the (strongly monotone) shy game on G .*

Proof. The strategy σ_{mb} is monotone by the previous lemma, and by property (i) of the definition, the components available for the robber correspond to those in plays consistent with σ , thus σ_{mb} is winning for the cops. To calculate the number of cops used by σ_{mb} , recall that the set of cops placed in step i is $C_{i+1}^{\text{mb}} = C_{i+1}^c \cup \text{mb}(R_i^{\text{mb}}, M(\pi[i])) \cup \text{mb}(R_{i-1}^{\text{mb}}, M(\pi[i-1])), R_i^{\text{mb}}$, where C_{i+1}^c were the chasers placed by σ , i.e. the set $C_i \cap R_i$, where C_i are all cops placed by σ in the corresponding position. Since σ was a weakly monotone strategy, the set C_i blocks $R_i \rightarrow M(\pi[i])$, and the previous C_{i-1} blocked $R_{i-1} \rightarrow M(\pi[i-1])$. Thus $|\text{mb}(R_i, M_i)| \leq |C_i| \leq k$ and $|\text{mb}(R_{i-1}, M_{i-1})| \leq |C_{i-1}| \leq k$, and, of course, $|C_{i+1}^c| \leq k$. Therefore $|C_{i+1}^{\text{mb}}| \leq 3k$. \square

Corollary 4.15. $cn_G(\text{wmshyDAG}) \leq cn_G(\text{shyDAG}) \leq 3 \cdot cn_G(\text{wmshyDAG})$

To convince oneself that these inequalities are not trivial, and that blocking minimally makes a difference, consider the following lemma.

Lemma 4.16. *There are graphs on which the cops have to make more than one guarding moves successively in order to win in the weakly monotone shy game, respectively in the strongly monotone game with the least possible number of cops.*

Proof. Let $n \geq 6$. Consider the graph G_n depicted in Figure 8 (recall that edges without arrows denote edges in both directions). Arrows that connect parts of the graph enclosed in a rectangle lead to or from all vertices of the graph. The graph consists of a vertex c_0 and n parts A_i that are isomorphic to each other and connected only to c_0 and in the same way. Every A_i consists of

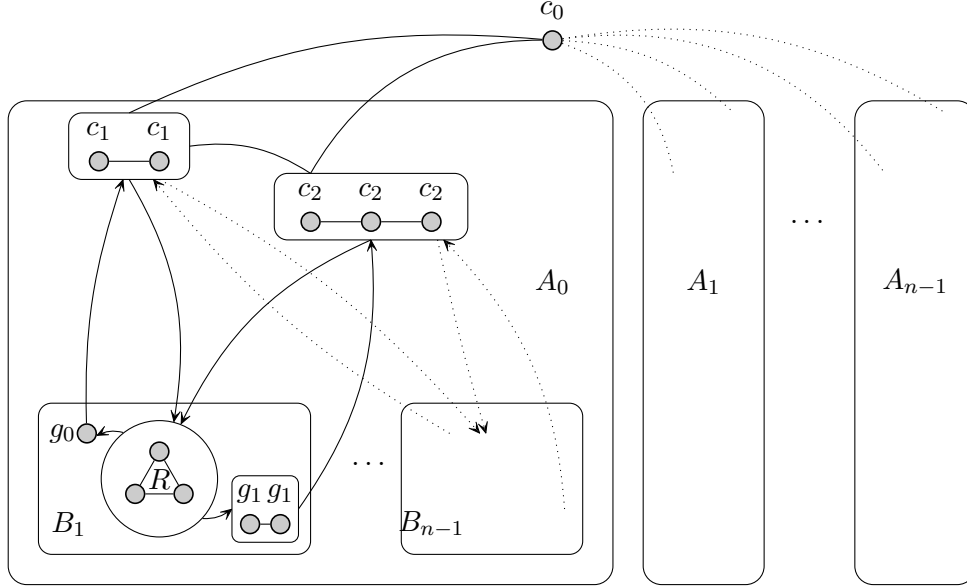


Figure 8: The cops need more than one guarding move in a row.

a 2-clique with vertices labeled in the picture with c_1 and a 3-clique with vertices labeled in the picture with c_2 that are connected to each other and to c_0 . Further, A_i contains n parts B_j . Each B_j contains a 3-clique R , a single vertex labeled with g_0 and a 2-clique with vertices labeled with g_1 . The connections are shown in the figure.

If the cops are allowed to make multiple guarding moves in a row, 6 of them suffice to capture the robber strongly (and thus weakly) monotonously. One cop is placed on c_0 (a chasing move), the robber chooses a component A_i . Then the cops occupy vertices c_1 and c_2 in further chasing moves, the robber chooses a part B_j in A_i . Then the cop from c_0 goes to g_0 , which is a guarding move, and then the cops from c_1 go to the both vertices g_1 . Note that if the robber remains in R , placing cops on g_1 is again a guarding move. Finally, the cops from c_2 capture the robber in R . Note that if there are 7 cops, it is possible to place this additional cop on a vertex in R instead of making the second guarding move and then win as before.

If the cops are not permitted to make two guarding moves in a row, the robber has the following winning strategy in the weakly (and thus strongly) monotone game against 6 cops. In the first move, the robber occupies c_0 and waits there until it is occupied by a cop. In that moment, there is a cop free component A_i (as there are 6 cops and 6 components A_0, \dots, A_{n-1} , but one cop occupies c_0). The robber goes to that cop free component A_i and waits on the 5-clique that is build by vertices c_1 and c_2 . When the cops occupy this clique, there is a cop free part B_j in A_i and the robber runs there. Note that cops on the clique are chasers, so the only free cop is that from c_0 . If he is placed in R , the robber stays, some other cop must move up, and the cops lose. If he is placed on one of the vertices g_1 (which is a guarding move), as no second guarding move is allowed, the only possible next move for cops is to place the one from g_1 on R – and lose as before. Hence, we can assume that the cop from c_0 is placed on g_0 , a guarding move. The next move must be chasing and the only possibility is to place a cop from c_1 in R . Now the cop on g_0 cannot be taken away, as a path to c_1 would be cop free, and the cops on c_2 are still bound as well. So is the cop in R (his move was chasing). Thus there is only one free cop (on c_1). He makes a guarding move, then a chasing move to R and the cops lose. \square

4.3 Decomposition

Our next goal is to define a decomposition of graphs in the spirit of [JRST01] for the strongly monotone shy game. Let G be a graph. A *shy-monotone tree decomposition* of G is a tuple (T, C, R) where T is a directed tree with root r and edges oriented away from the root, and $C, R : V(T) \rightarrow 2^G$ are functions with the properties listed below, which intuitively correspond to the placements of the cops and the component of the robber. For a node $t \in V(T)$ we write $ch(t)$ for the set of chasers corresponding to t , i.e. $ch(t) = C(t) \cap R(t)$, and we denote by $g(t)$ the guards, $g(t) = C(t) \setminus R(t)$. Moreover, we write $m(t)$ for the union $\bigcup_{s \preceq t} ch(s)$, i.e. for the set of all chasers from the nodes above t in T .

- (1) For the root r , $R(r) = V(G)$.
- (2) For every $(t, t') \in E(T)$, $R(t')$ is a strongly connected component of $R(t) \setminus ch(t)$.
- (3) For every $t \in V(T)$ if t_1, \dots, t_n are all direct successors of t , then

$$R(t) = ch(t) \cup \bigcup_{i=1}^n R(t_i).$$

- (4) For every $(t, t') \in E(T)$ holds:

$$m(t) \cap \text{Reach}_{G-(C(t) \cap C(t'))} R(t') = \emptyset,$$

i.e. there is no path from $R(t')$ to $m(t)$ avoiding $C(t) \cap C(t')$.

Note that from Item (1) and Item (3) it follows that every vertex of G is contained in the image of ch , $\bigcup ch(V(T)) = V(G)$. Indeed, for a node t without successors, we obtain from Item (3) that $R(t) = ch(t)$, and applying this item inductively proves that, for each node t , the component $R(t)$ is covered by $\bigcup_{t \preceq t'} ch(t')$. Since, by Item (1), in the root $R(r) = V(G)$, we have $\bigcup ch(V(T)) = G$.

The *width* of a shy-monotone tree decomposition (T, C, R) is defined as $\max \{|C(t)| : t \in V(T)\}$.

Proposition 4.17. *Let G be a graph. The following statements are equivalent.*

- (1) k cops capture the robber in the shy-monotone game on G .
- (2) There is a shy-monotone tree decomposition (T, C, R) of G of width k .

Proof.

- (1) \Rightarrow (2). Let σ be a strategy for k cops on G . We construct (T, C, R) inductively, starting with the root r with $R(r) = V(G)$ and we set $C(r)$ to the first placement of the cops chosen by σ . We continue the construction by following a play consistent with σ in each component chosen by the robber, and setting $C(t')$ to the vertices occupied by cops placed if the robber makes the respective move to $R(t')$. Items (2) and (3) follow from the general definition of the game, while Item (4) follows from the game being weakly-monotone.
- (2) \Rightarrow (1). From the decomposition (T, C, R) we construct a strategy σ_T . The first move of the cops is to $C(r)$, where r is the root of T . For each move of the robber to R' , the cops respond with the move to $C(t')$, where t' is the successor of t with $R(t') = R'$. Items (2) and (3) guarantee that this strategy is well defined, while Item (4) guarantees that it is winning. Obviously, the number needed cops is the width of the decomposition.

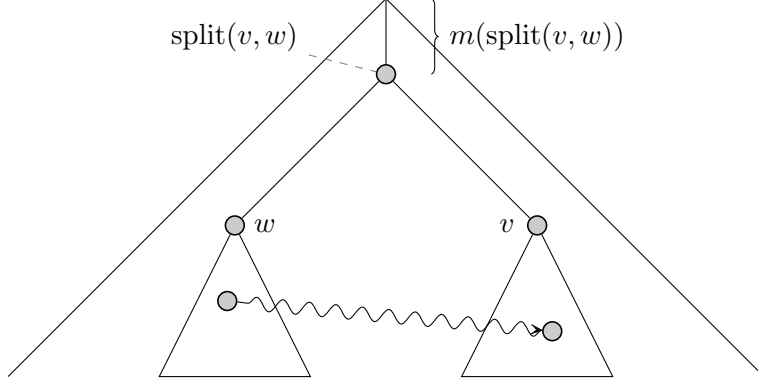


Figure 9: The order \triangleleft "to the right of".

□

We continue with an analysis of the decompositions. Let σ be a strategy for the cops in the shy-monotone game on G and T the corresponding decomposition tree. For a non-empty set of vertices A there is a unique *split vertex* $split(A)$ which is the latest common predecessor of all vertices of A . We also write $split(a, b)$ for $split(\{a, b\})$ and $split(a, A)$ for $split(\{a\} \cup A)$.

The proof of the next lemma is easy and we omit it.

Lemma 4.18. *If weak DAG-width of a graph G is k , then there is a winning strategy σ for the cops that always prescribes to place exactly one cop in a move, i.e. if $(M, C, R) \rightarrow (M', C, C', R)$ is a move according to σ , then $|C' \setminus C| = 1$.*

It follows that we can turn any shy-monotone tree decomposition into one with $|ch(t)| = 1$ for all $t \in V(T)$.

Corollary 4.19. *For every graph G with $cn_G(\text{wmshyDAG}) = k$ there is a shy-monotone tree decomposition (T, C, R) of width k where for all $t \in V(T)$, there is a vertex $w \in V(G)$ with $|ch(t)| = \{w\}$.*

We define an order on the vertices of a graph that corresponds the order in which the robber is chased in some plays which are played according to T . Let G be a graph and let T be its shy-monotone tree decomposition with $|ch(t)| = \{t\}$, for each $t \in V(T)$. Let v and w be two vertices of the graph. We say that w is *to the right of* v (and v is *to the left of* w) and write $v \triangleleft w$ if

- 1) w is on the path from the root of T to v , or
- 2) there is a path from w to v in $G - m(split(w, v))$.

In other words, w is to the right of v in G if, in a position in which a chaser occupies w , there is a cop free path from w to v . In the decomposition, we have in that case either $v \in R(w)$ and $w \notin R(v)$, or $split(v, w) \notin \{v, w\}$ and there is a path from w to v that avoids vertices above $split(v, w)$, see Figure 9 for the explanation of our terminology.

Clearly, \triangleleft is a partial order. We abuse the notation and denote any linearisation of \triangleleft also by \triangleleft .

As a next step, we show a simple, but useful property of a shy-monotone tree decomposition. Informally, the following lemma says that there is no path from left to right in the decomposition tree which avoids common predecessors of the first and the last vertices on the path.

Lemma 4.20. *If $v \triangleleft w$ and $\text{split}(v, w) \notin \{v, w\}$, then $m(\text{split}(v, w))$ blocks $v \rightarrow w$.*

Proof. Let P be a path from w to v and let P' be a path from v to w . We show that $P' \cap m(\text{split}(v, w)) \neq \emptyset$. Let $u \in G$ be a vertex in P' such that $\text{split}(u, w)$ has a minimal distance from the root of the decomposition tree. Then $P' \subseteq R(\text{split}(u, w))$. Indeed, if there is a vertex $u' \in P' \setminus R(\text{split}(u, w))$, then $\text{split}(u', R(\text{split}(u, w)))$ is nearer to the root than $\text{split}(u, w)$ (by Item (2) of the definition of a shy-monotone tree decomposition) and thus nearer than $\text{split}(u, w)$ (as $w \in R(\text{split}(u, w))$) contradicting the choice of u .

As $\text{split}(v, w) \notin \{v, w\}$ and by the definition of $\text{split}(\cdot)$, u and w are in different components of $R(\text{split}(u, w)) - \text{split}(u, w)$. As there is a path from w to u (concatenate P with with prefix of P' up to u), there is no path from u to w in $R(\text{split}(u, w)) - \text{split}(u, w)$, i.e. $\text{split}(u, w) \in P'$, so $P' \cap m(u, w) \neq \emptyset$. By the choice of u , we have $m(u, w) \subseteq m(v, w)$, so $P' \cap m(v, w) \neq \emptyset$. \square

If the robber leaves his component, he moves from the right to the left in the decomposition tree. By property (2) he can return to his left component only via $ch(t)$ for some. However $ch(t)$ are vertices where there are or have been chasers, so a winning cop strategy does not allow the robber to visit them. Thus he cannot return.

Lemma 4.21. *For every winning strategy σ in the weakly monotone game in every play $\pi = (C_0, R_0), (C_0, C_1, R_0), \dots$ consistent with σ , if the robber leaves a component R with a move $(C_i, C_{i+1}, R_i) \rightarrow (C_{i+1}, R_{i+1})$, then the cops on $C_i \cap C_{i+1}$ block $R_{i+1} \rightarrow R$. Thus the robber will never be able to enter R_i again.*

It is not known whether determining the DAG-width of a graph is solvable in non-deterministic polynomial time. For weak DAG-width, however, it is. The argument is that shy-monotone tree decompositions have polynomial size in the size of the graph.

Theorem 4.22. *Given a graph G and a natural number k , it is in NP to decide whether G has weak DAG-width at most k .*

Proof. The algorithm guesses the decomposition tree, which has size $\mathcal{O}(|G|^2)$ (because for each new chaser the guards have to be moved at most at most $|G|$ many times) and checks in polynomial time whether it is correct. \square

4.4 From shy to weakly monotone game, shy-similar strategies

First we define some conditions on the players' strategies that can be assumed without loss of generality.

Definition 4.23. A *chasing (guarding) move* of cops is a move where only chasing (guarding) cops are placed. (Note that both sorts of cops may be taken.) A cop strategy is *pure* if it consists only of guarding and chasing moves (and has no mixed moves).

Lemma 4.24. *If k cops have a winning strategy, then k cops have a pure winning strategy.*

Proof. Assume an arbitrary winning strategy f for k cops in the weakly monotone game on a graph G . At the beginning, only chasing moves are possible. Later on, instead of a mixed move $(C, v) \rightarrow (C, C', v)$, where $C_c = (C \setminus C') \cap \text{empt}(v, C)$ are the new chasers and $C_g = (C' \setminus C) \setminus \text{empt}(v, C)$ are the new guards, make first the guarding part, i.e. place cops on C_g of the move. If the robber changes his component, take the cops from G away and continue translating the strategy as if the robber went to the new component one move ago, i.e. before the cops move to C' . This is possible,

as the cops were guarding and thus did not change the component and thus the resulting position is still consistent with f . Note that the number of times the robber changes his component is finite. So assume w.o.l.g. that the robber remains in his component. Then the cops make the chasing part of their move, i.e. the cops are placed on C . It is easy to see that every robber move leads to position that is consistent with f . Further, no strong non-monotonicity occurs. Thus the new strategy is winning for the cops. \square

Lemma 4.25 (cf. [PR10], Lemma 8). *In a weakly monotone DAG-width game, if the robber has a winning strategy σ against k cops, then he also has a strategy that never prescribes to change his component if no cop was placed on a vertex reachable for the robber in the previous move.*

Proof. Assume that σ prescribes to move to a component C although no cop was placed in the reachability region of the robber. Change the strategy such that the robber never moves in such positions. Obviously, some cop eventually must be placed in the region, otherwise the robber wins. After this, the robber can still move to the same component of the current position as from C . \square

Proposition 4.26. *If k cops have a winning strategy in the strongly monotone shy robber game on G , then $2k$ cops have a winning strategy in the weakly monotone game on G .*

Proof. We say “shy game” for the strongly monotone shy robber game and “weak game” for the weakly monotone game. We translate the moves of the robber from the weak game to the shy game and the moves of the cops vice versa. Let σ be a pure winning strategy for the cops in the shy game. We describe the new *shy-similar* strategy $\text{shy-sim}(\sigma)$ for the weakly monotone game.

Consider a robber move $(M', C, C', R) \mapsto (M', C', R')$ in the weak game. If $R' \subseteq R$, then we translate the move as $(M', C, C', R) \mapsto (M', C', R')$ and take the next move according to σ , so that nothing changes with respect to σ . Otherwise, i.e. if R' is not a subset of R , we consider the latest move $(M_i, C_i, R_i) \mapsto (M_{i+1}, C_i, C_{i+1}, R_i)$ of the cops, such that $R_i \supseteq R'$. (Since it is the latest such move, we know that $R_{i+1} \not\supseteq R'$.) As σ is strongly monotone, M_i is blocked from R' by $C' \setminus R$. Furthermore, $M' \setminus C_{i+1}$, i.e. the set of vertices where chasers have been placed after position $(M_{i+1}, C_i, C_{i+1}, R_i)$, is not reachable from R' either. Let us place all the guards for the position where R' appears in the continuation of the play from $(M_{i+1}, C_i, C_{i+1}, R_i)$ towards R' . Due to Lemma 4.25 we can assume that the robber remains on R' during this time. After this move, we remove the other guards (here weak non-monotonicity can occur) and place the chasers as in the position for R' . This is the only place where (weak) non-monotonicity occurs. We have the same position that would occur if the robber would have moved to R' in the shy-monotone game, and we continue to play σ from there. \square

Definition 4.27. A winning strategy σ' is *shy-similar* if there is a winning strategy σ for the cops in the strongly monotone shy robber game such that $\sigma' = \text{shy-sim}(\sigma)$ where $\text{shy-sim}(\sigma)$ is the strategy that is constructed from σ as shown in Proposition 4.26.

Corollary 4.28. *If k cops win the weakly monotone cops and robber game on G , then $6k$ cops win the weakly monotone game on G .*

Proof. If $cn_G(\text{wmDAG}) \leq k$, then $cn_G(\text{wmshyDAG}) \leq k$ because the cops can use the same winning strategy. By Corollary 4.15, $cn_G(\text{shyDAG}) \leq 3k$ and by Proposition 4.26, $cn_G(\text{wmDAG}) \leq 6k$. \square

4.5 Strongly monotone strategies: two attempts

In this section, we use the decomposition defined above to construct a strongly monotone winning strategy for a bounded number of cops. Our construction is a combination of two approaches: leaving tied cops and freezing the context. Leaving cops is a transformation of a strategy σ by not removing tied cops, i.e. those who must be removed according to σ , but whose removal would immediately lead to strong non-monotonicity. Freezing the context changes a given strategy by marking the current robber component R and playing further only in R , i.e. omitting any changes outside the component, until the robber leaves R or is captured: the cops outside of R are “frozen”. In particular, no cops are placed outside the robber component. Obviously, both transformations produce strongly monotone strategies, but may use more cops than σ . In the following we define both approaches formally and show that, first, both taken independently lead to an unbounded number of additional cops they introduce, but, second, they can be combined into one transformation that uses only a quadratic number of additional cops.

4.5.1 Leaving tied cops is not enough

To make precise which cops are tied, we define the *front* of a subset X of vertices of a graph G with respect to R . Let $X, R \subseteq G$, $X \cap R = \emptyset$. Then the front $\text{front}_G(R, X)$ is the inclusion minimal subset of X that blocks $R \rightarrow X$ in G . If $R = \{v\}$, we also write $\text{front}_G(v, C)$. Let us prove that this set is unique. Indeed, assume that two distinct minimal subsets $X_0 \subseteq X$ and $X_1 \subseteq X$ block $R \rightarrow X$. Then w.l.o.g. there is a vertex $v \in X_0 \setminus X_1$. As X_0 is minimal, there is a $X_0 \setminus \{v\}$ -free path from R to v . As X_1 blocks $R \rightarrow X$, this path goes through a vertex $w \in X_1 \setminus X_0$. However the prefix of the path from R to w is X_0 -free, which contradicts that X_0 blocks $R \rightarrow X$.

The *leaving-cops* strategy σ_{lc} is as σ , but it leaves the cops from $\text{front}_G(v, C)$ on their vertices. Here v is the robber vertex and C is the placement of the cops. More formally, we define σ_{lc} as a memory strategy. The memory stores the cop placement we would have playing according to σ . So a memory state is a set $P \subseteq V$. Initially, $P = \emptyset$. When the robber moves, P does not change. In a position (C, v) with a memory state P , the new strategy prescribes to move as if the position was (P, v) , but removing only those cops that are not reachable from the robber vertex. In other words,

$$\sigma_{lc}(C, v) = \text{front}_G(v, C \cup \sigma(P, v)).$$

Obviously, if σ is a strongly monotone winning strategy for k cops, then σ_{lc} is strongly monotone winning strategy for k cops. If σ is a weakly monotone winning strategy, then so is σ_{lc} , but σ_{lc} may use more cops.

It is not a priori clear whether there is a class of graphs and a strategy σ such that σ uses a bounded and σ_{lc} an unbounded number of cops on graphs from that class. However, we show in this subsection that σ_{lc} can be arbitrarily bad compared to σ . The idea is to iterate the argument from [KO08], with the (rough) correspondence between the graph G_2 in Figure 10 and their graph D_p (Figure 1 in [KO08]) as follows. The component C_1 in G_2 corresponds to C_0 in D_p , the component R_1 corresponds to C_2 in D_p , A_1 corresponds to C_1^1 , and finally C_2 in G_2 to C_1^2 in D_p . Disregarding the sizes, the only D_p -edges missing in G_2 are between C_1^1 and C_0 , which corresponds to connecting A_1 and C_1 in G_2 . While adding an edge from C_1 to A_1 is possible in G_2 , it is essential that no $A_1 \rightarrow C_1$ edge is present. But these edges, corresponding to $C_1^1 \rightarrow C_0$ edges in D_p , are not important in D_p .

Lemma 4.29. *Let $m \geq 1$. There is a class of graphs $(G_n)_{n>0}$ and winning strategies σ^n for $4m$ cops such that σ_{lc}^n uses $4m^2(n-1)$ cops.*

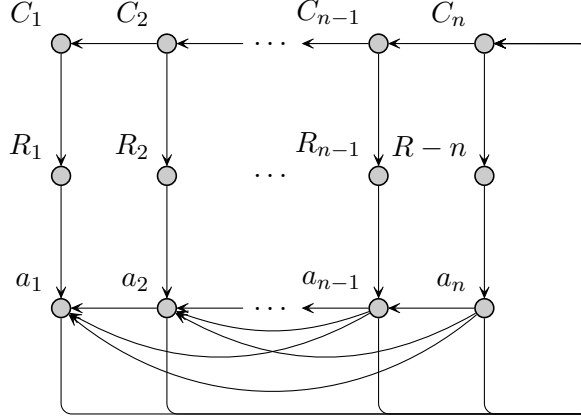


Figure 10: Strategy σ_{lc} uses an unbounded number of additional cops.

Proof. Consider the graph G_n in Figure 10. It consists of n cliques C_1, \dots, C_n of size $2m$, n cliques R_1, \dots, R_n of size $3m$, and n independent sets A_1, \dots, A_n of size m (we could also take cliques instead of independent sets). Each clique C_i is connected to the clique C_{i-1} , for $i = 2, \dots, n$, i.e. there are edges from every vertex of C_i to every vertex of C_{i-1} . Furthermore, each C_i is connected to R_i , each R_i to A_i , and each A_i to C_n for $i = 1, \dots, n$. Finally, each A_i is connected to A_j for $i = 2, \dots, n$ and $j < i$.

The strategy σ^n is as follows. At the beginning, $2m$ cops occupy C_1 . We can assume that the robber goes to R_1 because all other components are reachable from R_1 . Then m cops occupy A_1 . If the robber remains in A_1 , the cops from C_1 go to R_1 and capture the robber. The other possibility for the robber is to switch to the component that contains C_2 . Now the cops from A_1 are taken away from the graph (inducing weak non-monotonicity). The robber can only remain in his component. Then $2m$ new cops occupy C_2 , the robber goes to R_2 , m cops from C_1 occupy A_2 , the robber switches to the component containing C_3 , the cops are taken from A_2 and the rest of C_1 and placed on C_3 and so on. In the last step, the cops occupy C_n , the robber is in R_n and m cops occupy A_n . The robber switches to some A_i , but the cops from A_n expel him from any A_i and the robber is captured in A_1 . Note that in every move during the described game, A_n and thus all A_i are reachable from the robber component. Hence, the robber sticking to the same strategy as above (always switching to the new C_i), the strategy σ_{lc}^n prescribes to leave cops on all A_i . \square

4.5.2 Freezing the context is not enough

Given a strategy σ , the *context freezing* strategy σ^{**} is obtained from σ as follows. We define two memory variables: P stores the placement of cops as if we played according to σ (analogously to the case of σ_{lc}) and $\mathcal{R} = (R_1, \dots, R_n)$ is a stack of memorized robber components with $R_{i+1} \subset R_i$, for all i . Initially, $P = \emptyset$ and $\mathcal{R} = ()$ is the empty stack. A robber move $(C, C', v) \rightarrow (C', w)$ does not change P and $\mathcal{R} = (R_1, \dots, R_i)$ is updated by deleting all R_j with $w \notin R_j$.

For the cop move, let (C, v) be a position in a play consistent with σ^{**} played so far and let P and $\mathcal{R} = (R_1, \dots, R_i)$ be the current memory state. The variable P is updated to $\sigma(P, v)$. We define σ^{**} by

$$\sigma^{**}(C, v) = (C \setminus \text{cmpt}_G(v, C)) \cup (\sigma(P, v) \cap \text{cmpt}_G(v, C)),$$

i.e. “the context” $(G - \text{cmpt}_G(v, C))$ is not changed and we place cops as prescribed by σ , but only

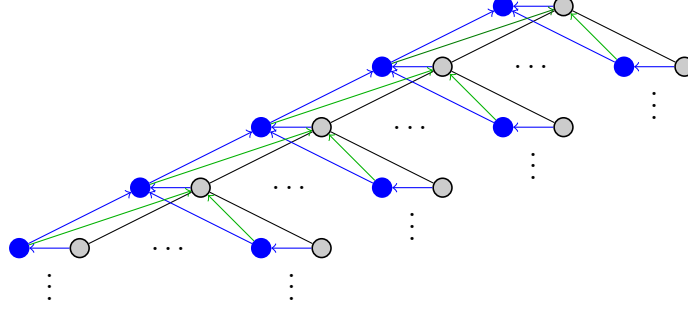


Figure 11: $\text{DAG-w}(G_n) = 4$ but the robber wins against n cops if they never guard.

within the robber component. If $(\sigma(P, v) \cap R_i) \setminus \text{cmpt}_G(v, C) = \emptyset$ (all cops are placed outside the robber component), the stack \mathcal{R} remains unchanged. Otherwise we push $\text{cmpt}_G(v, C)$ on the stack, thus freezing the new context.

The next lemma states that changing an arbitrary weakly monotone winning strategy σ to σ^* may introduce an unbounded number of additional cops: it is essential that the cops are placed also in the context. The counter examples are double trees, shown in Figure 11.

Lemma 4.30 ([PR11]). *There exist graphs G_n , for all $n \in \mathbb{N}$, such that $\text{DAG-w}(G_n) \leq 4$ but every winning strategy of the cops which is restricted to place cops only inside the robber component uses at least $n + 1$ cops.*

Sketch. Let, for $i \in \{0, 1\}$ and $0 < m, n \in \mathbb{N}$, $A(i, m, n) = (\{1, \dots, n\} \times \{i\})^{\leq m}$ be the set of all sequences of length at most m over the alphabet $\{1, \dots, n\}$ labeled with i (the labeling is used only to distinguish vertices). Let, for a $v = (v_0, i), \dots, (v_l, i) \in A(i, m, n)$, v' be the word $(v_0, 1 - i), \dots, (v_l, 1 - i) \in A(1 - i, m, n)$. Consider the following class of directed graphs $G_n = (V_n, E_n)$ for $0 < n \in \mathbb{N}$ (see Figure 11). Hereby $V_n = T_n^0 \cup T_n^1$ where $T_n^0 = A(0, m + 1, n)$ and $T_n^1 = A(1, m + 1, n)$. The edges are defined by $E_n = E_n^0 \cup E_n^1 \cup E_n'$. Hereby $E_n^0 = \{(v, vj), (vj, v) : v \in A(0, n, n), j \in A(0, 1, n)\}$, $E_n^1 = \{(vj, v) : v \in A(1, n, n), j \in A(1, 1, n)\}$, and $E_n' = \{(v, v') : v \in A(0, m + 1, n)\} \cup \{(vj, v') : v \in A(1, n, n), j \in A(1, 1, n)\}$.

The first statement of the theorem is easy to see. For the second one, note that it makes no sense for the cops to leave out holes, i.e. to place cops on subtrees of (T_n^0, E_n^0) rooted at a vertex $v \in T_n^0$, but not on v . Indeed, due to the high branching degree, the robber can switch between subtrees of v going into those having no cop in them until v is occupied by a cop. Clearly, in this position, there is no need to have cops in subtrees other than the one with robbers in it. So we can assume that the cops play top-down, i.e. they never leave out holes. Then the robber strategy is just to stay in the left-most branch. Note that after a vertex $v \in T_n^0$ is occupied by a cop the vertex $v' \in T_n^1$ is not in the SCC of the robber any more. It is easy to see that more and more cops become bounded, i.e. for every cop on a vertex v , there is a cop free path from the robber vertex to v . \square

4.5.3 Combining leaving cops and freezing

In this section we define a translation of a winning strategy in the weakly monotone game to a winning strategy in the strongly monotone game. Our solution is a combination of the approaches discussed in the previous sections. First we describe the strategy informally.

Recall that $\text{front}_G(R, X)$ is the inclusion minimal subset of X that blocks $R \rightarrow X$ in G . The combined strategy σ_{lc}^* is obtained from σ as follows. Recall that robber components are defined

with respect to the set M of vertices that have been occupied by chasers. The cops start to play sticking to σ until the robber changes his component. If σ prescribes to remove a cop from some vertex v such that weak non-monotonicity occurs, the cop is not removed (and neither any other tied cops, but let us concentrate on v for now). The cops play further according to σ as if the tied cops were removed until the robber chooses a component R such that $v \notin R$. If that never happens and the robber always chooses the component containing v , then a new cop, say occupying w , can be tied only if the cop on v becomes not tied (and we can continue with w in place of v). Indeed, if σ is strongly monotone against a shy robber, a cop can become tied only if the (now not shy) robber changes his component; if the robber component R contains v and the robber leaves R towards some R' , then v is not reachable from R' by Lemma 4.21.

When we have $v \notin R$, the context of R is frozen. In that position we have at most k cops in the context of R including v . Now the cops play according to σ restricted to R until the robber leaves it, or he is captured. Hereby, if a cop becomes unreachable from the robber vertex, he is removed from the graph and can be reused later. If the robber leaves R and enters another component R' , the placement of the cops outside R is still the same as when the robber chose R (and not R'), so he will be captured in or expelled from R' and from any other such component in the same way as for R , and the cops win.

It remains to see that the new strategy uses at most k^2 cops. Our argument is that any tied cop becomes untied before k^2 other cops become tied. When a cop on v is tied, we freeze at most k cops and continue to play only within R . Up to the position when according to σ the last k th cop enters R , we have enough cops by *induction on the number of cops used by σ* : we need at most $(k-1)^2 = k^2 - (2k-1)$ cops. If the robber is already captured in R , we are done. If he leaves R before the k th cop enters R according to σ , we argue for his next component as for R . For the (most interesting) case that the k th cop comes to R , we are going to show that the cop tied at the beginning, i.e. that on v , is now untied. If the cop on v was still tied, he would be in particular reachable from the robber vertex, so the robber can leave R . We will see that then he can also reach some vertices that induce strong non-monotonicity, which contradicts the fact that σ is weakly monotone.

The combined strategy Let σ be a shy-similar strategy. We define the combined strategy σ_{lc}^* by induction on the maximal number of cops that appear in a play consistent with σ . The combined strategy uses the same memory as σ^* : it keeps track of a play consistent with σ by memorizing a cop placement P and stores the history of freezing in a stack \mathcal{R} . If p is a position, then $P(p)$ and $\mathcal{R}(p)$ are the values of P and \mathcal{R} , respectively, in position p . At the beginning of a play, $P(\emptyset) = \emptyset$ and $\mathcal{R}(\emptyset) = ()$. A robber move $(C, C', v) \rightarrow (C', w)$ does not change P and $\mathcal{R} = (R_1 \dots, R_i)$ is updated by deleting all R_j with $w \notin R_j$.

For the cop move, let (C, v) be a position in a play consistent with σ_{lc}^* played so far and let $\mathcal{R}(C, v) = (R_1, \dots, R_i)$. The variable P is updated to $\sigma(P, v)$. In the cop move, there are two differences to σ^* . The more substantial one is that now the cops are placed also outside the robber component, but still not outside R_i . The other one is that we can change the cop placement in the context removing some cops if this does not directly lead to non-monotonicity. Note that removing those cops could also be performed for σ^* . We did not do it to keep the description of σ^* simpler, but this would not make σ^* work. Formally we define

$$\sigma_{\text{lc}}^*(C, v) := \text{front}_G(v, C) \cup (\sigma(P(C, v), v) \cap R_i).$$

Note that within R_i , even σ prescribes only strongly monotone moves and removes unreachable

cops, so we could represent σ_{lc}^{**} in a way more similar to σ^{**} :

$$\sigma_{\text{lc}}^{**}(C, v) = (\hat{C} \setminus R_i) \cup (\sigma(P(C, v), v) \cap R_i)$$

where $\hat{C} = C \cap \text{front}_G(v, C)$.

We update the stack \mathcal{R} by pushing $\text{cmpt}_G(v, C)$ on \mathcal{R} if $(\sigma(P(C, v), v) \cap R_i) \setminus \text{cmpt}_G(v, C) \neq \emptyset$ and let \mathcal{R} unchanged otherwise.

As $\text{front}_G(v, C) \subseteq \sigma_{\text{lc}}^{**}(C, v)$, it is immediately clear that σ_{lc}^{**} is strongly monotone. It also guarantees a capture of the robber because σ^{**} does it and $\sigma^{**}(C, v) \subseteq \sigma_{\text{lc}}^{**}(C, v)$, so σ_{lc}^{**} prescribes to place a cop into the robber component again and again. We have to prove that σ_{lc}^{**} uses at most k^2 cops.

Lemma 4.31. *If σ is a shy-similar winning strategy for k cops, then σ_{lc}^{**} uses at most k^2 cops.*

Proof. We show that, for every shy-similar winning strategy σ for k cops in the weakly monotone game and every graph G , every tied cop in a play consistent with σ_{lc}^{**} on G becomes not reachable from the robber vertex before k^2 new cops are tied. Clearly, this implies the statement of the lemma.

The proof is done by induction on k . Without loss of generality, assume that G is strongly connected (otherwise repeat the argument for every strongly connected component). Consider a fixed play π consistent with σ_{lc}^{**} . If $k = 1$, then there are no tied cops and the statement is trivial. Let $k > 1$. Let $(C_0, C_1, v_0) \rightarrow (C_1, v_1)$ be a move in π such that $\mathcal{R}_0 = \mathcal{R}(C_0, v_0) = (R_1, \dots, R_i)$ and $\mathcal{R}_1 = \mathcal{R}(C_1, v_1) = (R_1, \dots, R_j)$ with $j \leq i$. Assume that the move results in a new tied cop on $v \in C_1$.

Either v is in the robber component until the end of π , or not. We show that in the first case there are no further tied cops, so we are done. As σ is shy-similar, tied cops appear only when robber changes his component, compare the proof of Proposition 4.26. If another cop on w becomes tied, then the robber changes his component, say from R_1 to R_2 . But then either $v \notin R_1$, or $v \notin R_2$, so there is a position in that the robber is not in the same component as v .

For the other case, let $(C_2, C_3, v_2) \rightarrow (C_3, v_3)$ be the first move such that $v \notin \text{cmpt}(v_3, C_3)$. Let $\mathcal{R}(C_3, v_3) = (R_0, \dots, R_m)$. Without loss of generality we can assume that the rest of π is played in R_m . Otherwise, until the robber leaves R_m and goes to some component R' , cops are placed only in R_m , which is not reachable from R' without introducing strong non-monotonicity (by Lemma 4.21), so we would repeat our arguments for R_m for R' .

Until σ prescribes to place k cops in R_m , i.e. while $|\sigma(P, v) \cap R_m| < k$, by the induction hypothesis for σ and R_m , we have at most $(k-1)^2$ tied cops. Note that R_m is strongly connected, so we do not violate our assumption that the graph on that we play is strongly connected. Consider the first move $(C_4, v_4) \rightarrow (C_4, C_5, v_4)$ with $|P(C_5, v_4) \cap R_m| = k$. We want to show that then v is not reachable from v_4 , which means that the cop on v is not tied any more.

As G is strongly connected, there is a path P from v to v_4 in G . Recall that $v \notin R_m$ by the case distinction, but by our assumption that the remaining of the play takes place in R_4 , we have $v_4 \in R_m$. If v were reachable from R_m in $G - P(C_5, v_4)$, by Lemma 4.21, cops outside of R_m block $\{v_4\} \rightarrow R_m$. However, in position $(P(C_5, v_4), v_4)$ all cops are in R_4 , a contradiction. \square

We can count the number of additional cops more accurately. For the first tied cop we need to freeze at most k new cops, for the next tied cop at most $k-1$ cops and so on, so in total, we can come up with $k^2/2 + k/2$ cops. Finally, we obtain the desired result.

Theorem 4.32. *If k cops have a winning strategy in the weakly monotone shy robber game, then $18k^2 + 3k$ cops have a winning strategy in the strongly monotone game.*

Proof. Assume that k cops have a winning strategy σ in the weakly monotone shy robber game. Then by Corollary 4.15 $\text{mb}(\sigma)$ is a winning strategy for $3k$ cops in the strongly monotone shy robber game. By Proposition 4.26 one needs $2 \cdot 3k = 6k$ cops to win weakly monotonically. Finally one needs $((6k)^2 + 6k)/2 = 18k^2 + 3k$ cops to win strongly monotonically. \square

5 Comparing Width Measures with Respect to Generality

This section is devoted to the question, given two measures a and b , whether the class of graphs with bounded values of a is a subclass of the class of graphs with bounded values of b .

5.1 Comparing DAG-width and Kelly-width

Kelly-width is a complexity measure for directed graphs introduced by Hunter and Kreutzer in [HK08]. Kelly-width is similar to DAG-width and can be defined by a decomposition, by a graph searching game and by an elimination order, similar to tree width.

An *elimination order* \triangleleft for a graph $G = (V, E)$ is a linear order on V . For a vertex v define $V_{\triangleright v} := \{u \in V : v \triangleleft u\}$. The *support* of a vertex v with respect to \triangleleft is

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

The *width* of an elimination order \triangleleft is $\max_{v \in V} |\text{supp}_{\triangleleft}(v)|$. The *Kelly-width* $\text{Kelly-w}(G)$ of G is one plus the minimum width of an elimination order of G .

Hunter and Kreutzer conjecture in [HK08, Conjecture 30] that DAG-width and Kelly-width bound each other by a constant factor. More generally, the question is whether there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that, for every graph G , we have

- (1) $\text{DAG-w}(G) \leq f \cdot \text{Kelly-w}(G)$ and
- (2) $\text{Kelly-w}(G) \leq f \cdot \text{DAG-w}(G)$.

In [HK08] it is shown that if $\text{Kelly-w}(G) = k$, then $2k - 1$ cops have a (possibly non-monotone) winning strategy σ in the DAG-width game. We demonstrate that σ is, in fact, weakly monotone, thus answering the first question affirmatively.

Theorem 5.1. *If $\text{Kelly-w}(G) = k + 1$, then $\text{wm-DAG-w}(G) \leq 2k + 1$.*

Proof. Our proof follows the proof of Theorem 20 from [HK08], which shows that an elimination order of width k induces a (possibly non-monotone) strategy for $2k + 1$ cops in the DAG-width game. What we prove additionally is just that the constructed strategy is weakly monotone.

Let \triangleleft be an elimination order for G of width k . We define a weakly monotone winning strategy σ for $2k + 1$ cops in the weakly monotone game on G .

Any play consistent with σ can be partitioned into two kinds of rounds: the blocking rounds and the chasing rounds. A blocking round consists of a blocking cop move and an answer of the robber. A chasing round may contain a longer sequence of moves. The cops are divided into two teams: a team of $k + 1$ blockers and a team of k chasers. While a play proceeds, a cop may change his team.

During the play, after every blocking round, the following invariant will hold. Let the blockers occupy the set of vertices B , let the chasers occupy the set C and let the robber be on v .

1. $|B| \leq k + 1, C = \emptyset$.
2. If u is the \triangleleft -least vertex from B , then $v \triangleleft u$ and B blocks $\{v\} \rightarrow V_{\triangleright u} \setminus B$.

In the first move, $k + 1$ blockers occupy the \triangleleft -maximal vertices of G and the robber chooses some vertex. It is trivial that the invariant holds.

Consider a position after some blocking round has been just finished. Let v be the robber vertex, B the set of vertices occupied by blockers and C the set of vertices occupied by chasers such that the invariant holds. Let $u = \min_{\triangleleft}(B)$ and let \mathcal{R} be the set of components of $G - V_{\triangleright u}$ where $V_{\triangleright v} = V_{\triangleright} \cup \{u\}$. Let \sqsubset be the linear order on \mathcal{R} defined by $R \sqsubset R'$ if and only if $\max_{\triangleleft}(R) \triangleleft \max_{\triangleleft}(R')$ and let \sqsubseteq be its reflexive closure. Let R be the component in \mathcal{R} with $v \in R$ and let $w = \max_{\triangleleft}(R)$. The chasing round proceeds as follows. The cops announce to place (at most k) chasers on $\text{supp}_{\triangleleft}(w)$. The robber chooses a vertex v' in a component R' . If $R' \sqsubseteq R$, then in the next position the chasers on $\text{supp}_{\triangleleft}(w)$ block every path from v' to $V_{\triangleright u'}$ where $u' = \min_{\triangleleft}(\text{supp}_{\triangleleft}(w))$. (Indeed, assume that there is a path P from v' to some u'' with $u' \triangleleft u''$ such that $P \cap \text{supp}_{\triangleleft}(w) = \emptyset$. Let (a, b) be the first edge with $u' \triangleleft b$. Then there is a path from w to b' via v' , thus $b \in \text{supp}_{\triangleleft}(w)$, a contradiction.) This completes the chasing round.

If $R \sqsubset R'$, then the chasers are removed from $\text{supp}_{\triangleleft}(w)$ and placed on $\text{supp}_{\triangleleft}(w')$ where $w' = \max_{\triangleleft}(R')$. As the robber can change to a \sqsubset -greater component only until he reaches B , this process is finite and at some point the robber is blocked by the chasers, i.e. we have the previous case. Note that by the definition of $\text{supp}_{\triangleleft}$, for all $w \in V$ we have $w \triangleleft \min_{\triangleleft}(\text{supp}_{\triangleleft}(w))$. Hence, as w is chosen to be the \triangleleft -maximal in the robber component, the chasers are always placed outside of the robber component. Thus placing and removing them in a chasing round never induces strong non-monotonicity.

When the chasing round is over, the next blocking round begins. Let v be the robber vertex, R the robber component of $G - V_{\triangleleft u}$ (where $V_{\triangleleft u} = \{v \in V : v \triangleleft u\}$ and u is still the \triangleleft -minimal element in B), and w the \triangleleft -maximal element of R . In the blocking round, the chasers from the previous chasing round become blockers (let B' be the set of vertices they occupy) and a blocker from B is placed on w . Other old blockers from B become chasers and are removed from the graph. After the robber makes his move (say, he goes to v'), the blocking round is finished.

We have to check that the invariant still holds and that no strong non-monotonicity occurred during the last blocking round. The first invariant property holds because we used at most k cops as chasers and the new blockers are the old chasers plus the cop on w . Furthermore, the chasers have been removed from the graph. The second property (that B' blocks $\{v'\} \rightarrow V_{\triangleright u'} \setminus B$ where $u' = \min_{\triangleleft}(B')$) holds by the construction and implies that removing cops from B was (even strongly) monotone. Finally, the space available for the robber shrinks after every blocking round because the cops occupy w , so the robber is finally captured. \square

Corollary 5.2. *If $\text{Kelly-w}(G) = k$, then $\text{DAG-w}(G) = \mathcal{O}(k^2)$.*

5.2 Separating D-width from DAG-width, Kelly-width and directed tree width

Safari suggests in [Saf05] D-width as another structural complexity measure. Recall that for a directed graph G , we denote its undirected underlying graph by \bar{G} . A D -decomposition of a graph G is a pair $(T, (X_t)_{t \in V(T)})$ where T is a directed tree with edges oriented away from the root. Furthermore for all $t \in V(T)$, $X_t \subseteq V(G)$ and

- (1) $\bigcup_{t \in V(T)} X_t = V(G)$, and

- (2) for all strongly connected sets $S \subseteq V(G)$ the underlying undirected subgraph of $T[\{t \in V(T) : X_t \cap S \neq \emptyset\}]$ is a connected subtree of \bar{T} .

The *width* of $(T, (X_t))$ is $\max_{t \in V(T)} |X_t|$.² The *D-width* $D-w(G)$ of G is the minimum width of a D-decomposition of G .

The following definition of D-width may be more useful in algorithmic applications and suits our goals better. For a graph G , if $X, Y \subseteq V(G)$ and X is a union of strongly connected components of $G - Y$, we say that X is *Y-normal*. DS-width is a complexity measure that differs from D-width at most by the factor of two. Let G be a graph. A *DS-decomposition* of G is pair $(T, (X_t)_{t \in V(T)})$ where T is a directed tree with edges oriented away from the root r and X_t are sets of vertices of G such that the following holds. Let $X_{\geq t} = \bigcup_{q \geq t} X_q$ for all $t \in V(T)$. Then

- (1) $\bigcup_{t \in V(T)} X_t = V(G)$,
- (2) for all $v \in V(G)$ the set $\{t \in V(T) : v \in X_t\}$ is connected in \bar{T} ,
- (3) for all edges $(s, t) \in E(T)$, $X_{\geq t} \setminus X_s$ is $(X_s \cap X_t)$ -normal.

The width of $(T, (X_t))$ is $\max_{t \in V(T)} |X_t|$. The *DS-width* of G , $DS-w(G)$ is the minimum of the widths of all DS-decompositions of G .

Lemma 5.3 (See [Gru08]). *For all graphs G , $D-w(G) \leq DS-w(G) \leq 2D-w(G)$.*

Proof. Let $(T, (X_t)_{t \in V(T)})$ be a D-decomposition of width k . We obtain a DS-decomposition $(T', (X'_t)_{t' \in V(T')})$ of width $2k$ from $(T, (X_t)_{t \in V(T)})$ as follows. Replace every edge $(s, t) \in E(T)$ by a new node st and edges (s, st) and (st, t) and let X_{st} be $X_s \cup X_t$. Then $(T', (X'_t)_{t' \in V(T')})$ is a DS-decomposition of G . Indeed, assume that for some edge $(s, st) \in E(T')$ there is a path P from some $w \in X_{\geq st} \setminus X_s = X_{\geq t} \setminus X_s$ to some $v \in V(G) \setminus (X_{\geq st} \setminus X_s)$ and back to w . Assume for a contradiction that P avoids $X_s \cap X_{st} = X_s$. Then P is a strongly connected subgraph of G , so by the second property of D-decomposition, the set $\{t \in V(T) : P \cap X_t \neq \emptyset\}$ is connected in \bar{T} . Thus $X_s \cap P \neq \emptyset$, but we assumed that this is not true. Hence $X_{\geq st} \setminus X_s$ is $(X_s \cap X_{\geq st})$ -normal. With the same argument one can see that for all edges of the form (st, t) , the set $X_{\geq t} \setminus X_{st}$ is $(X_t \cap X_{\geq st})$ -normal.

Properties (1) and (2) follow trivially from the properties of the D-decomposition. Furthermore, it is clear that the width of $(T', (X'_t)_{t' \in V(T')})$ is at most $2k$.

Now assume that we have a DS-decomposition $(T, (X_t)_{t \in V(T)})$ of width k . We show that it is also a D-decomposition. Let S be a strongly connected set of G and assume that $\{t \in V(T) : X_t \cap S \neq \emptyset\}$ is not connected in \bar{T} . Let q, s and t be some nodes of T such that $X_q \cap S \neq \emptyset$, $X_t \cap S \neq \emptyset$, $X_s = \emptyset$ and s is on the path P_{tqs} from t to q in \bar{T} . Choose s, q and t such that P_{tqs} has minimal length. Either $q < s$ or $q < t$, say $q < t$, then $(q, t) \in E(T)$ (the case $q < s$ is analogous). As S is strongly connected, there is a path P from $X_{\geq t} \setminus X_q$ to $X_{\geq s} \setminus X_q$ and back within S . As $S \cap X_q = \emptyset$, P avoids X_q . Note that $(X_{\geq s} \setminus X_q) \cap (X_{\geq t} \setminus X_q) = \emptyset$, so P leaves $X_{\geq t} \setminus X_q$ and returns there without visiting X_q thus violating the normality condition of the DS-decomposition for the edge (q, t) . \square

We separate D-width from directed tree width, DAG-width, Kelly-width, and from the cop- and robber-monotone component game. First we show in Theorem 5.4 that if D-width is bounded, then a bounded number of cops suffices to capture the robber in a cop- and robber-monotone way in the component game. It follows that then directed tree width is bounded as well (this is already known from [Saf05]). It is known that there are classes of graphs where directed tree width is bounded but

²In [Saf05] the width is $\max_{t \in V(T)} |X_t| - 1$.

neither Kelly-width, nor DAG-width are: undirected binary trees with additional edges forming the upward transitive closure [BDH⁺12]. The D-width of those graphs is also bounded. We show that there is a class \mathcal{G} of graphs where three (four) cops win in the cop- and robber-monotone component (resp. reachability) game, but whose D-width is unbounded (Theorem 5.6). Hence, directed tree width and DAG-width are bounded on \mathcal{G} , but D-width is unbounded. We also show that Kelly-width is bounded on \mathcal{G} . Finally, we use Theorem 3.2 to separate D-width from directed tree width in another way in Theorem 5.7.

Theorem 5.4. *For all graphs G , if there is a DS-decomposition of G of width k , then $cn_G(\text{cmdtw}) \leq k$.*

Proof. Let $(T, (X_t)_{t \in V(T)})$ be a DS-decomposition of width k . The cops have the following winning strategy. In the first move they occupy X_r where r is the root of T . In general they keep the invariant true that if the current position is (C, R) , then $C = X_s$ for some $s \in V(T)$ and $R \subseteq X_{\geq t} \setminus X_s$ for some t with $(s, t) \in E(T)$. Then the next move of the cops is to (C, X_t, R) and after the next robber move the invariant holds. Note that $X_t \setminus X_s \subseteq R$. Note also that R is a strongly connected component of $G - (X_s \cap X_t)$, so the play is robber-monotone. Furthermore, by property (2), it is also cop-monotone. When the cops reach a leaf, the robber is captured. Clearly, exactly k cops are used. \square

The opposite direction fails because the cops may be forced to occupy the same vertex when the robber goes to different components. Assume that we reached a position (C, C', R) and the robber can choose R'_1 or R'_2 . In both cases after playing some time the cops must occupy a vertex v : in the first case because $v \in R'_1$ and in the second case because they have to block the robber in R'_2 . The decomposition corresponding that strategy has v in different successors of the bag that corresponds to position (C, R) , but not in the bag of (C, R) itself. However this violates the connectivity condition of a DS-decomposition. Theorem 5.6 shows that the described situation is unavoidable.

In the proof of Theorem 5.6 we use a technical notion of a game which at least partially corresponds to D-width. The *D-width game* on a graph G is another type of graph searching games that does not match our framework. At the beginning the cops group components of G into equivalence classes and the robber chooses one class and goes there. At this moment the cops do not see the robber and only know his class. From now on every cop can be placed only within that class. Then the cops make a move as in all games described before and group the emerging components within the current class into new classes and so on.

Formally the cop positions are (C, \mathcal{R}) where $C \subseteq V(G)$ and $\mathcal{R} = \{R_1, \dots, R_m\}$ is a set of components of $G - C$. Hereby every R_i is a component the cops consider to be a possible robber component. The cops can move to a position $(C, C', \sim, \mathcal{R})$ where $C' \subseteq \bigcup_{1 \leq i \leq m} R_i$ and \sim is an equivalence relation on components of $G - C'$. From $(C, C', \sim, \mathcal{R})$ the robber can move to a position (C', \mathcal{R}') where $\mathcal{R}' = \{R'_1, \dots, R'_s\}$ is the set of components R'_j (for $j \in \{1, \dots, s\}$) of $G - C'$ such that $R'_j \subseteq R_i$ for some R_i and all R'_j are \sim -equivalent. In other words the robber chooses an equivalence class of components, a *group*. If there is a path from some R'_j outside of R'_j and then back to R'_j in $G - (C \cap C')$, then the robber wins (by the non-robber-monotonicity). He also wins all infinite and all non-cop-monotone plays. The cops win if they capture the robber (i.e. he has no legal move).

Lemma 5.5. *If $\text{DS-w}(G) = k$, then $2k$ cops have a winning strategy in the D-width game on G .*

Proof. Let $(T, (X_t)_{t \in V(T)})$ be a DS-decomposition where the root of T is r . Note that for all $(s, t) \in E(T)$ the set $\bigcup_{q \geq t} X_q \setminus X_s$ is a union of components of $G - X_s$. The cops occupy r in the first move and for components R_1 and R_2 of $G - X_r$ they define $R_1 \sim R_2$ if and only if R_1 and

R_2 are both contained in $\bigcup_{q \geq t} X_q \setminus X_r$ for some $(r, t) \in E(T)$. The robber chooses a \sim -class, i.e. essentially an edge (r, t) . Note that the robber is blocked in $\bigcup_{q \geq t} X_t \setminus X_r$ by $X_r \cap X_t$. Then the cops occupy $X_r \cup X_t$ and define the equivalence relation in the same way as before whereby now t plays the role of r . Finally the cops capture the robber at the latest in some X_s for a leaf $s \in V(T)$. The cop strategy is cop-monotone by the monotonicity of the DS-decomposition. \square

Theorem 5.6. *There is a class of graphs G_n such that 3 cops have a cop- and robber-monotone winning strategy in the directed tree width and DAG-width games on each G_n and $\text{Kelly-w}(G_n) = 4$, but $\text{D-w}(G_n) \geq n$.*

Proof. Informally, the graph G_n consists of two vertex disjoint parts (below we give a formal definition). One is a copy of T_{n+1}^n with root $r(T_{n+1}^n)$ and additional edges forming the downward transitive closure. The other part has (another copy of) $V(T_{n+1}^n)$ as its vertex set and edges $\{(va, v) : v \in \{0, \dots, n\}^n, a \in \{0, \dots, n\}\}$. We denote the second part as T' . The parts are connected by edges going from a vertex v in T_{n+1}^n to its copy v' in T' and from v' to the parent of v . Now we transform the resulting graph by applying the following operation once on each vertex of T_{n+1}^n (except the root) in a top-down manner. The current vertex $v \in V(T_{n+1}^n)$ (except the root) is replaced by n copies v_1, \dots, v_n . Let w be the parent of v . Then the edge $\{v, w\}$ is replaced by edges $\{v_i, w\}$ and the edge (v, v') by edges (v_i, v') . Every v_n is the root of a copy of the subtree of T_{n+1}^n rooted at v . Hereby the edges going to and from T' are also copied.

Formally $V(G_n)$ is the disjoint union of $V(T_{n+1}^{n^2})$ and $V(T_{n+1}^n)$ with edges E_T, E_{tr}, E_{up} and E_{cross} where $E_T = \{\{v, va\} : v \in \{0, \dots, n^2 - 1\}^{\leq n}, a \in \{0, \dots, n^2 - 1\}\}$ are edges forming $T_{n+1}^{n^2}$, $E_{tr} = \{(v, vw) : v, w, vw \in \{0, \dots, n^2 - 1\}^{\leq n}\}$ is the downward transitive closure on $T_{n+1}^{n^2}$, $E_{up} = \{(va, v) : v \in \{0, \dots, n\}^{\leq n}, a \in \{0, \dots, n\}\}$ forms T_{n+1}^n and E_{cross} connects $T_{n+1}^{n^2}$ and T_{n+1}^n as follows. For a word $w = w_1 \dots w_m \in \{0, \dots, n^2 - 1\}^{\leq n}$ with $|w| > 0$ let w' be the word w'_1, \dots, w'_m with $w'_i = \lfloor w_i/n \rfloor$. Then there are edges $(w, w') \in V(T_{n+1}^{n^2} \times T_{n+1}^n)$ and $(w', w_1 \dots w_{m-1})$ (i.e. if $w = w_1$, then the edge is (w', ε)).

The winning strategy for the cops in the cop-monotone directed tree width game is, roughly, to traverse $T_{n+1}^{n^2}$ and T_{n+1}^n in parallel downwards. As long the robber is in $T_{n+1}^{n^2}$, the cops descend from the root of $T_{n+1}^{n^2}$ to some leaf w which is in the current robber component and in T_{n+1}^n also from the root to the leaf w' . If the robber changes to T_{n+1}^n , he finds himself in a component that consists of one vertex and is captured in the next move. It is easy to see that three cops suffice to win.

In the DAG-width game, four cops follow the robber, in T_{n+1}^n in the same way as in $T_{n+1}^{n^2}$: in parallel downwards. For the Kelly-width, consider the elimination order where every vertex of T_{n+1}^n is smaller than every vertex of $T_{n+1}^{n^2}$ and vertices within T_{n+1}^n and within $T_{n+1}^{n^2}$ are ordered in the obvious way (following the depth-first search). Then the support of every vertex is at most three, so the Kelly-width is four.

We show by induction on n that the robber has a winning strategy in the D-width game against n cops. At the beginning he chooses the group containing the component with $r(T_{n+1}^{n^2})$ and continues to do so as long as the root is not occupied by the cops in a move $(C, \mathcal{R}) \rightarrow (C, C', \sim, \mathcal{R})$. For vertex $v \in V(T_{n+1}^{n^2})$ the subtree of $T_{n+1}^{n^2}$ rooted at v is denoted T_v and similarly we write $T_{v'}$ for the subtree of $T' = T_{n+1}^n$ rooted at v' . Let r_1, \dots, r_{n^2} be the direct successors of $r(T_{n+1}^{n^2})$. Due to the robber-monotonicity the cops have visited vertices in at most $n - 1$ sets $T_{r_i} \cup T_{r'_i}$. Hence there are at least n vertices r_i , say r_1, \dots, r_n , with the same successor r' in T_{n+1}^n such that all T_{r_i} for $1 \leq i \leq n$ and $T_{r'}$ are cop free. Every T_{r_i} and $T_{r'}$ are current components and the equivalence relation \sim declared by the cops defines groups of them.

There are two cases. In the first case there are at least two groups. One of them contains some T_{r_i} , but not $T_{r'}$. The robber chooses this group and plays from now on only on T_{r_i} . His strategy is to remain in the group containing a vertex v that is possibly high in the tree. Due to the high branching degree when the cops occupy v , there is a direct successor w of v such that T_w is cop free. The robber chooses the group containing w and plays further in the same way. During this play the cops occupying the vertices v_1, v_2, \dots on the path from $r(T_{n+1}^{n^2})$ to v cannot be removed because there is a path from w via $T_{r'}$ to all v_i and then back to w . This path is cop free because the cops are not allowed to occupy $T_{r'}$, which is not in the current group. Hence if the cops leave some v_i , the robber wins by non-robber-monotonicity. When the robber reaches a leaf, all n cops stay on the path from $r(T_{n+1}^{n^2})$ to that leaf and cannot be removed, so the robber wins.

In the other case there is only one group. The robber may be in each component C containing some T_{r_i} . Each such component has an isomorphic copy G_{n-1} as a subgraph. (Note that C is not a copy of G_{n-1} because its branching degree is still n^2 and not $(n-1)^2$ as in G_{n-1} .) Thus by the induction hypothesis the robber wins against $n-1$ cops, so if the cops should win, they need the cop from $r(T_{n+1}^{n^2})$ in C . But then the robber can reach $r(T_{n+1}^{n^2})$ from another T_{r_j} , which causes the non-robber-monotonicity. \square

It was conjectured that directed tree width and D-width are the same ([Saf05, Page 750]³). We show, however, that the gap between them is not bounded by any function (which is clear from Theorem 5.6, but we give yet another proof.)

Theorem 5.7. *There is a class of graphs with bounded directed tree width and unbounded D-width.*

Proof. Consider the class of graphs from Theorem 3.2. The directed tree width of the graphs from that class is bounded. If the D-width were bounded, DS-width would be bounded as well (Lemma 5.3), and by Lemma 5.4 a bounded number of cops could capture the robber on each graph in a cop-monotone way, but this is not the case. \square

5.3 Oriented tree width

Definition 5.8. Let G be an undirected graph. A *tree decomposition* of G is a tuple $(T, (X_t)_{t \in V(T)})$ where T is an undirected tree and

- $\bigcup_{t \in V(T)} X_t = V(G)$,
- for all $\{v, w\} \in E(G)$ there is some $t \in V(T)$ with $\{v, w\} \subseteq X_t$,
- for all $v \in V(G)$ the set $\{t \in V(T) : v \in X_t\}$ induces a (connected) subtree of T .

Definition 5.9. Let G be a directed graph. An *oriented tree decomposition* is a pair $(T, (X_t)_{t \in V(T)})$ where T is an orientation of an undirected tree and for each $t \in V(T)$, $X_t \subseteq V(G)$ such that $E(G)$ can be partitioned into two, possibly empty, sets $E(G) = E^t \cup E^{shc}$ (the tree edges and the shortcut edges) and the following conditions hold.

- (1) $(\bar{T}, (X_t)_{t \in V(T)})$ is a tree decomposition of $(V(G), E^t)$.
- (2) If $(u, v) \in E^{shc}$, $u \in X_s$, and $v \in X_t$, then there is a path from s to t in T .

³Safari actually conjectures that D-width equals directed tree width which would imply cop-monotonicity.

We say that an edge $e \in E(G)$ is *covered by the tree* if e is contained in some bag X_t . Otherwise we say that e is a *shortcut edge*. The *width* of an oriented tree decomposition $(T, (X_t)_{t \in T})$ is $\max_{t \in T} |X_t|$.

The *oriented tree width* $\text{otw}(G)$ of a graph G is the minimum width over all oriented tree decompositions of G .

The following lemma states that all undirected edges are covered by the tree. The proof is easy and we omit it.

Lemma 5.10. *Let G be a graph. Let $(T, (X_t)_{t \in V(T)})$ be an oriented tree decomposition of G and let $E(G) = E^t \cup E^{shc}$ be a corresponding partition of $E(G)$. If $(v, u) \in E(G)$ and $(u, v) \in E(G)$, then $(u, v) \in E^t$ or $(v, u) \in E^t$.*

In the next lemma we give a normal form for oriented tree decompositions.

Lemma 5.11. *Let G be a graph, let $(T, (X_t)_{t \in V(T)})$ be an oriented tree decomposition of G and let $E^t \cup E^{shc}$ be a corresponding partition of $E(G)$. Then there is an oriented tree decomposition of G of the same width such that for all $(s, t) \in V(T)$, $X_s \not\subseteq X_t$.*

Proof. We construct a new decomposition by successively eliminating bags whose neighbours are their supersets. This suffices as by monotonicity if $X_s \subseteq X_t$, then $X_s \subseteq X_q$ for all $q \in V(T)$ on the path between s and t in \bar{T} . Let $(s, t) \in E(T)$ be nodes with $X_s \not\subseteq X_t$. The new decomposition is $(T', (X_r)_{r \in T'})$ where $V(T') = (V(T) \setminus \{s\})$ and $E(T') = (E(T) \setminus \{(r, s), (s, r) : r \in T\}) \cup \{(t, t') : (s, t') \in E(T)\} \cup \{(t', t) : (t', s) \in E(T)\}$. It is straightforward to check that all conditions of an oriented tree decomposition hold. Furthermore the width of the decomposition did not change. \square

Our next goal is to compare oriented tree width with D-width.

Theorem 5.12. *For all graphs G , $\text{D-w}(G) \leq \text{otw}(G)$.*

Proof. Let $(T, (X_t)_{t \in V(T)})$ be an oriented tree decomposition of G . We argue that it is also a D-decomposition (of the same width). First, $\bigcup_{t \in V(T)} X_t = V(G)$, as $(\bar{T}, (X_t)_{t \in V(\bar{T})})$ is a tree decomposition of \bar{G} . Let S be a strongly connected set of vertices of G and assume that there are s, q and t in $V(T)$ with $X_s \cap S \neq \emptyset$, $X_q \cap S = \emptyset$, and $X_t \cap S \neq \emptyset$ such that q is on the path from s to t in \bar{T} . We choose s, q , and t such that the path from s to t has minimum length. As S is strongly connected, there is a path P from some vertex in $X_s \cap S$ to $X_t \cap S$ in G . Let (v, w) be the first edge of P with $v \in (P \cap X_s) \setminus X_t$ and $w \in (P \cap X_t) \setminus X_s$. (If (v, w) does not exist, then there is a vertex $u \in X_s \cap P \cap X_t$ and thus $u \in X_q$, as q is between s and t in \bar{T} , but $P \subseteq S$, so $u \in X_q \cap S$, a contradiction.) By the choice of s, q and t , the edge (v, w) is not covered by the tree (otherwise there would be an edge of the tree decomposition connecting s and t , but there is another path from s via q to t). It follows that (v, w) is a shortcut edge and hence all edges of the tree decomposition on the path from s to t are oriented from s to t . By a symmetric argument we can show that all edges are oriented from t to s , a contradiction. \square

By ?? it follows that directed tree width is bounded in oriented tree width as well.

Corollary 5.13. *For every class \mathcal{G} of graphs, if oriented tree width is bounded on \mathcal{G} , then directed tree width is bounded on \mathcal{G} .*

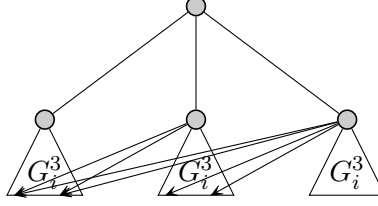


Figure 12: The graph G_{i+1}^3 .

Recall that undirected binary trees with the additional upward transitive closure from [BDH⁺12] separate directed tree width and D-width from DAG-width and Kelly-width. It is easy to see that the oriented tree width of such a graph G is also small. The oriented decomposition tree has the same shape as G and all tree edges are oriented upwards, so $\text{otw}(G) = 2$. Thus on some graphs oriented tree width is bounded, but DAG-width and Kelly-width are not. The next theorem shows that the opposite is also true: on some graph classes with bounded directed tree width, DAG-width and Kelly-width, oriented tree width is not bounded by any function, so the measures are incomparable in this sense.

Theorem 5.14. *There is a family of graphs G_n with $cn_{G_n}(\text{dtw}) = \text{DAG-w}(G_n) = \text{Kelly-w}(G_n) = \text{D-w}(G_n) = 2$ such that for each $k > 2$ there is some n with $\text{otw}(G_n) > k$.*

Proof. The graph G_n is constructed inductively. Let G_1^m be a single vertex. Then G_{i+1}^m has a new root $r(G_{i+1}^m)$ with m successors v_1, \dots, v_m . Each such successor v_i is the root $r(G_i^m)$ of a copy of G_i^m and has outgoing edges to all leaves of the copies G_i^m rooted at v_j with $j < i$. The construction of the graph G_{i+1}^3 from G_i^3 is shown in ???. Finally, $G_n = G_n^m$. Formally, $V(G_n) = [n]^{\leq n}$ is the set of words of length at most n over the alphabet $[n]$, $E(G_n) = E_t \cup E_r$ where $E_t = \{(w, wa), (wa, w) : w \in [n]^{\leq n}, a \in [n]\}$ are edges forming the tree and $E_r = \{(wa, wbv) : a, b, \in [n], b < a, |v| = n - |w| + 1\}$.

In the DAG-width game two cops play from the top to the bottom of R along the path chosen by the robber. If the robber changes to a smaller subtree, the last placed cop follows him to that subtree on its root. Of course, one cop is unable to win. So $\text{DAG-w}(G) = 2$ and for the same reason also $cn_G(\text{dtw}) = 2$. The elimination order for the Kelly-width is the depth-first search with choosing the right-most successor first. The D-decomposition is the usual tree decomposition of a tree (every bag contains two neighboured vertices and one bag contains only the root): edges from E_r do not destroy the conditions of a D-decomposition.

Now assume towards a contradiction that there is an oriented tree decomposition $(T, (X_t)_{t \in V(T)})$ of G_n of width k where $n = k + 3$. Let $E(G_n) = E^t \cup E^{shc}$ be the corresponding partition of edges of G_n . We first analyse the tree decomposition of $T_n := \overline{(V(G_n), E^t)}$. By Lemma 5.10, $E_t \subseteq E^t$.

For a connected subgraph G' of an undirected graph G with tree decomposition $(T, (X_t)_{t \in V(T)})$, the restriction of $(T, (X_t)_{t \in V(T)})$ to G is $(T', (X'_t)_{t \in V(T')})$ where T' is the subgraph of T induced by bags t with $X_t \cap V(G') \neq \emptyset$ and $X'_t = X_t \cap V(G)$. Note that the restriction is a tree decomposition of G' of width at most the width of $(T, (X_t)_{t \in V(T)})$.

Claim 2. *There is a subtree T'_n of T_n of (the same) depth n and such that*

- every node of T'_n which is not a leaf has two children, and
- the restriction $(T', (X'_n))$ of $(T, (X_t))$ to T'_n is a natural decomposition, i.e. up to isomorphism:
 - $T' = T'_n$,

- $X'_r = \{\varepsilon\}$ (the root bag contains only the root of T'_n), and
- $X'_t = \{s, t\}$ where s is the predecessor of t if t is not the root bag.

Proof (of Claim 2). We use the characterisation of tree width by the tree width game. It is played on an undirected graph as the DAG-width game. It is well known that a tree decomposition of width k induces a winning strategy for $k+1$ cops. In the first move they occupy the root bag. The robber chooses a subtree of the decomposition tree and the cops occupy the root of that subtree in the next move. Continuing in that way they finally capture the robber in a leaf bag.

Let σ be the strategy for $k+1$ cops on T_n . For a vertex $v \in T_n$ let T_v be the subtree of T_n rooted at v . Consider a position of a play consistent with σ where the robber is in T_v and T_v is cop free. Then the following lemma holds.

Claim 3. *There are two children w_1 and w_2 of v such that in any play from the current position that is consistent with σ , for $i = 1, 2$, w_i is the first vertex of T_{w_i} occupied by a cop.*

Indeed, v has more children than there are cops. If a cop is placed in a subtree rooted at a child of v , then there will be at least one cop in that subtree until v is occupied (otherwise the robber-monotonicity is violated).

Now we define T'_n in a top-down manner. The root of T'_n is the root of T_n . Assume that a subtree of T_n up to some level is constructed. Let v be a current leaf and let $w_1, w_2 \in T_n$ be the children of v whose existence is guaranteed by Claim 3. Then v has two children in T'_n : w_1 and w_2 . Then $(T'_n, (X'_t)_{t \in V(T'_n)})$ is a natural decomposition. This proves the claim. \square

Without loss of generality let $T'_n = (\{0, 1\}^n, \{\{v, va\} : v \in \{0, 1\}^n - 1, a \in \{0, 1\}\})$. Consider the edges $e_1 = (01, 0^n)$ and $e_2 = (1, 01^{n-1})$ in E_r . By the construction of $(T', (X'_t)_{t \in V(T')})$, the edge e_1 is not covered by the tree. Thus the orientation of T' allows the path from 1 to 01^{n-1} . In particular all edges on the path from ε to 01^{n-1} are oriented towards 01^{n-1} , i.e. the edge $\{0, 01\}$ is oriented as $(0, 01)$. The edge e_2 is not covered by the tree either, so the orientation allows the path from 01 to 0^n . In particular the edge $\{0, 01\}$ is oriented as $(01, 0)$, a contradiction. \square

Theorem 5.15. *For all graphs G we have $\text{DS-w}(G) \leq \text{D-w}(G) \leq \text{otw}(G)$.*

Proof. Let $(T, (X_t)_{t \in V(T)})$ be an oriented tree decomposition of G of width k and let $E(G) = E^t \cup E^{shc}$ be a corresponding partition of $E(G)$. Let $\{s, t\} \in G(\bar{T})$ be an edge of \bar{T} . Let T^t be the maximal subtree of \bar{T} containing t , but not s and T^s the maximal subtree containing s , but not t . Let $X^t = (\bigcup_{q \in V(T^t)} X_q) \setminus X_s$ and $X^s = (\bigcup_{q \in V(T^s)} X_q) \setminus X_t$. Let $(v, w) \in E(G)$. If $v \in X^t$ and $w \in X^s$, then (v, w) is a shortcut edge and $(t, s) \in E(T)$ (and thus $(s, t) \notin E(T)$).

Let r be an arbitrary node of T that we declare to be the root. Let T_r be the orientation of T such that all edges are oriented away from r . We claim that $(T_r, (X_t)_{t \in V(T_r)})$ fulfils all requirements of a DS-decomposition except, possibly (4). Note that $(T, (X_t)_{t \in V(T)})$ and $(T_r, (X_t)_{t \in V(T_r)})$ have the same width. Requirements (1) and (2) hold because $(\bar{T}, (X_t))$ is a tree decomposition. For (3) assume for some $(s, t) \in E(T_r)$ that there is a path P that starts in X^t , leaves it and then returns to X^t such that $P \cap X_s \cap X_t = \emptyset$. Then there is an edge $(v, w) \in E(G)$ that goes from X^t to X^s , so $(t, s) \in E(T)$ (and $(s, t) \notin E(T)$), and there is an edge that goes from X^s to X^t , so $(s, t) \in E(T)$, a contradiction. \square

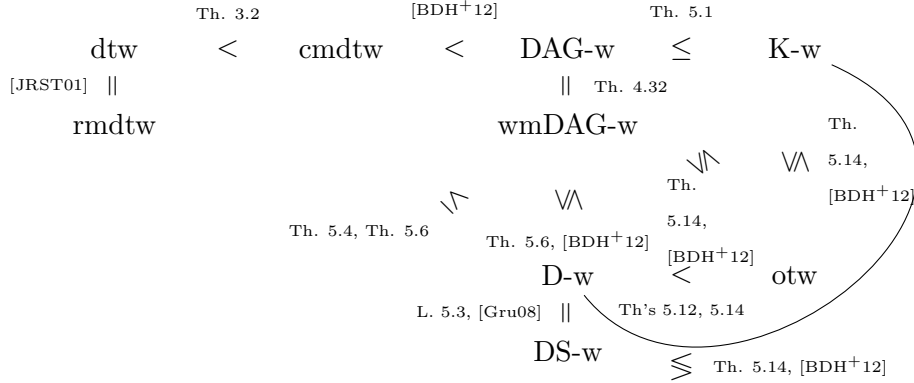


Figure 13: The relations between different measures.

6 Conclusion

6.1 The Relations between Widths and Cop Numbers

The relations between directed tree width, $cn_G(\text{rmdtw})$ (the robber monotone cop number in the component game), $cn_G(\text{cmdtw})$, DAG-width, weakly monotone DAG-width, Kelly-width, D-width, and oriented tree width are presented in ???. All relations are considered in terms of boundedness. If a and b are two measures from the above list, $a \leq b$ means that there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for all graphs G , $a(G) \leq f(b(G))$. We write $a < b$ if $a \leq b$ and there is a class \mathcal{G} of graphs and a number $t \in \mathbb{N}$ such that for all $G \in \mathcal{G}$, $a(G) \leq t$ and for all $s \in \mathbb{N}$ there is a graph $G \in \mathcal{G}$ with $b(G) > s$, i.e. a is bounded on \mathcal{G} and b is not. We write $a = b$ if there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for all graphs G , $a(G) \leq f(b(G))$ and $b(G) \leq f(a(G))$. Finally $a \lesssim b$ means that there is a class of graphs on which a is bounded and b is not and vice versa: there is a class of graphs on which b is bounded and a is not.

6.2 Future Work

We believe that the quadratic blowup in the number cops when we change from a weakly monotone winning strategy for the cops to a strongly monotone strategy can be reduced to a linear one. The largest ratio between the numbers of needed cops in the weakly monotone and the strongly monotone cases we are aware of is $4/3$ from the examples by Kreutzer and Ordyniak. However, the induction on the number of cops needed by the weakly monotone strategy seems to enforce the use of quadratically many cops. It would be interesting to achieve a linear upper bound or to find better lower bound than $4/3$.

Another topic for the future work is the gap between the non-monotone and weakly monotone case. This question seems to be the most interesting in this area. It would be also important to determine whether the inequality $\text{DAG-w} \leq \text{Kelly-w}$ should be strict or an equality.

It is not known whether DAG-width can be decided in NP as Kelly-width. Neither is known whether Kelly-width can be computed in time $n^{O(k)}$ where n is the size of the given graph and k is its Kelly-width.

References

- [Adl07] Isolde Adler. Directed tree-width examples. *J. Comb. Theory, Ser. B*, 97(5):718–725, 2007.
- [Bar06] János Barát. Directed Path-width and Monotonicity in Digraph Searching. *Graphs and Comb.*, 22(2), 2006.
- [BBG13] Guillaume Bagan, Angela Bonifati, and Benoit Groz. A trichotomy for regular simple path queries on graphs. In *Proc. 32nd Symposium on Principles of Database Systems*, pages 261–272, New York, NY, USA, 2013.
- [BDH⁺12] Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdržálek. The DAG-width of directed graphs. *J. Comb. Theory*, 102(4):900–923, 2012.
- [BG05] Dietmar Berwanger and Erich Grädel. Entanglement—A Measure for the Complexity of Directed Graphs with Applications to Logic and Games. In *LPAR '04*, volume 3452 of *LNCS*. Springer, 2005.
- [Die12] Reinhard Diestel. *Graph Theory, 4th Edition*. Springer, 2012.
- [FT08] Fedor V. Fomin and Dimitrios M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.
- [Gru08] Hermann Gruber. On the D-width of directed graphs. Manuskript, February 2008.
- [HK08] Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3), 2008.
- [JRST01] Thor Johnson, Neil Robertson, Paul Seymour, and Robin Thomas. Directed Tree-Width. *J. Comb. Theory, Ser. B*, 82(1), 2001.
- [KO08] Stephan Kreutzer and Sebastian Ordyniak. Digraph decompositions and monotonicity in digraph searching. In *Graph-Theoretic Concepts in Computer Science: 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*. Springer, 2008.
- [KO11] Stephan Kreutzer and Sebastian Ordyniak. Digraph decompositions and monotonicity in digraph searching. *Theor. Comput. Sci.*, 412(35):4688–4703, 2011.
- [Kre11] Stephan Kreutzer. Graph searching games. In Krzysztof R. Apt and Erich Grädel, editors, *Lectures in Game Theory for Computer Scientists*, chapter 7, pages 213–263. CUP, 2011.
- [PR10] Bernd Puchala and Roman Rabinovich. Parity games with partial information played on graphs of bounded complexity. In *MFCFS '10*, volume 6281 of *LNCS*. Springer, 2010.
- [PR11] Bernd Puchala and Roman Rabinovich. Graph searching, parity games and imperfect information. arXiv:1110.5575v1 [cs.GT], 2011.
- [Ree99] Bruce Reed. Introducing directed tree-width. *Electronic Notes in Discrete Mathematics*, 3:222 – 229, 1999.

- [Saf05] Mohammad Ali Safari. D-width: A more natural measure for directed tree width. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *MFCS*, volume 3618 of *Lecture Notes in Computer Science*, pages 745–756. Springer, 2005.
- [Tam10] Hisao Tamaki. A directed path-decomposition approach to exactly identifying attractors of boolean networks. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, pages 844–849, Oct 2010.