Graph Searching Games and Width Measures for Directed Graphs

Saeed Akhoondian Amiri[1], Łukasz Kaiser[2], Stephan Kreutzer[1], Roman Rabinovich*[1], and Sebastian Siebertz[1]

1 Logic and Semantic, Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{saeed.akhoondianamiri,stephan.kreutzer,roman.rabinovich,
sebastian.siebertz}@tu-berlin.de
2 LIAFA, CNRS & Université Paris Diderot†, lukaszkaiser@google.com

Abstract

In cops and robber games a number of cops tries to capture a robber in a graph. A variant of these games on undirected graphs characterises tree width by the least number of cops needed to win. We consider cops and robber games on digraphs and width measures (such as DAG-width, directed tree width or D-width) corresponding to them. All of them generalise tree width and the game characterising it.

For the DAG-width game we prove that the problem to decide the minimal number of cops required to capture the robber (which is the same as deciding DAG-width), is PSPACE-complete, in contrast to most other similar games. We also show that the cop-monotonicity cost for directed tree width games cannot be bounded by any function. As a consequence, D-width is not bounded in directed tree width, refuting a conjecture by Safari.

A large number of directed width measures generalising tree width has been proposed in the literature. However, only very little was known about the relation between them, in particular about whether classes of digraphs of bounded width in one measure have bounded width in another. In this paper we establish an almost complete order among the most prominent width measures with respect to mutual boundedness.

1998 ACM Subject Classification G.2.m Graph Theory

Keywords and phrases cops and robber games, directed graphs, DAG-width

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.34

1 Introduction

Graph searching games, also known as *cops and robber* or *pursuit-evasion* games, are an important type of games on graphs and digraphs studied intensively in the literature. While there are many different forms of graph searching games, the basic idea is always that a number of *searchers* tries to find or catch a *fugitive* hiding in the vertices or edges of a graph or digraph. See Section 2 for details of the games used in this paper and see [15] for an introduction and [12] for a comprehensive survey of graph searching games.

Graph searching games have originally been introduced to model the search of rescuers trying to find a miner lost in a mine after some accident. Any graph searching game defines a natural graph invariant assigning to every graph the minimal number of cops needed to

* Partially supported by ESF, http://www.esf.org.
† Currently at Google Inc.

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 34–47
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

guarantee capture of the robber. It has subsequently emerged that these graph invariants are closely related to width measures such as directed or undirected *tree* or *path width* studied in graph structure theory (see e.g. [8, 14, 2]).

In particular, a type of strategies for the cops called *monotone* strategies often corresponds exactly to decompositions such as *tree* or *path decompositions* and hence concepts such as *tree width* etc. can equivalently be defined by a particular variant of graph searching games. In this paper we will therefore treat strategies for the cops and corresponding graph or digraph decompositions equivalently, emphasising either of the two views whenever it seems more appropriate.

Graph searching games can be defined on undirected or directed graphs. On undirected graphs, the *visible robber game* defines exactly tree width, the *invisible robber game* defines path width and a variation of the invisible game called the *inert robber game* again defines tree width. These games can be generalised to digraphs in two different ways. In this way we naturally obtain games on digraphs corresponding to directed width measures such as *directed tree width* [14], *DAG-width* [4], *Kelly-width* [13] or *directed path width* [2]. We will therefore refer to these games as the *directed tree width games*, *DAG-width games* etc.

Structural width measures such as tree and path width have found important applications in algorithms and complexity theory. In view of the correspondence between graph searching games and such structural decomposition based width measures, a natural question arising is the problem of determining for a given graph or digraph the minimal number of cops that guarantees to capture the robber in a particular game variant. For most game variants including tree or path width games (both directed and undirected) one can show that this problem is in NP: decompositions, i.e. monotone winning strategies for the cops (see below), are of size polynomial in the input graph and one can therefore simply guess such a strategy and verify the correctness of the guess. In this way for most game variants relevant in this context it was shown that they can be decided in NP and they are usually NP-complete. Only the complexity of DAG-width games was left as an open problem as there the corresponding decompositions are DAG-like and hence not obviously seen to be polynomial.

Surprisingly, in this paper we show that deciding the DAG-width of a digraph is not only not in NP (under standard complexity theoretical assumptions), it is in fact PSPACE-complete and therefore exhibits the worst case complexity of such games. This result is quite unexpected and especially surprising as such a high complexity was to date only exhibited by a form of graph searching games called *domination games* (see [10, 9, 16]). In these games, each cop not only occupies his current vertex but a whole neighbourhood of fixed radius, which essentially allows to simulate set quantification making the problem PSPACE-complete.

The DAG-width game, however, is a straight forward translation of the NP-complete game for tree width to digraphs and to the best of our knowledge this is the only graph searching game with the usual capturing condition that exhibits such a complexity.

As a consequence of the proof technique used to prove this result we also show that there are classes of graphs for which any DAG decomposition of optimal width must contain a super polynomial number of bags. (If NP $\neq$ PSPACE, this would follow from the previous result, but we show this unconditionally.) Furthermore, we obtain that there cannot be a polynomial time approximation algorithm for DAG-width with only an additive error.

As explained above, the cop number of graph searching game variants is very closely related, and often equivalent, to standard width measures for graphs and digraphs. In the literature on digraph width measures a significant number of width measures have been proposed as directed analogue of undirected tree width. Among these are directed tree width [14], DAG-width [4], Kelly-width [13] and D-width [23]. Furthermore, there are some game variants such as cop-monotone directed tree width games and non-monotone DAG-width games, for which no corresponding width measure has been defined. The obvious question is how these different measures compare to each other, i.e. whether a class of digraphs of bounded width in one measure has bounded width in another. For some pairs of digraph width measures the relation has been determined, but to date there is no clear picture. In particular, the relation between DAG-width, Kelly-width, D-width and cop-monotone directed tree width games is not known. As the second main result of this paper we establish a nearly complete order among these width measures. The most difficult part hereby is to show that any class of digraphs of bounded Kelly-width also has bounded DAG-width.

A crucial concept in graph searching games is *monotonicity.* A strategy for the cops is *robber-monotone* if vertices unavailable for the robber at a position of a play never become available later on and it is *cop-monotone* if the cops never go back to a vertex they have left before. Monotone strategies are particularly well behaved and in fact, cop-monotone strategies are very similar to decompositions such as tree or path decompositions. A highly desirable property of a particular variant of graph searching games therefore is that the number of cops needed to catch the robber on a graph or digraph G with a monotone strategy is the same (or at least bounded in) the number of cops needed with any strategy. The number of extra cops needed for monotone strategies is called the *cop-* or *robber-monotonicity cost* of the game.

This *monotonicity* problem has driven the field of graph searching games from the very beginning, see e.g. [18, 5, 24, 6, 2, 11, 25, 14, 1, 27, 26, 12, 19, 7]. For undirected graphs, the monotonicity problem is by now well understood and most natural graph searching variants are indeed monotone. For directed graphs, the situation is very different. The games corresponding to directed path width are cop- and robber-monotone [2]. The games for directed tree width are not robber- and not cop-monotone, but a robber-monotone strategy requires at most three times the number of cops [14]. However, many important problems regarding monotonicity on directed graphs are still wide open.

Among the most important open problems in this respect are the questions whether the cop-monotonicity cost for the game corresponding to directed tree width can be bounded by any function and whether the robber-monotonicity cost for the games corresponding to DAG-width or Kelly-width can be bounded. In [23], it has been conjectured that the cop-monotonicity cost for directed tree width games is bounded, but the problem was left open to date. In this paper we refute this conjecture by showing that there is a class of graphs such that on every digraph in this class, 4 cops have a robber-monotone winning strategy in the directed tree width game, but the number of cops needed for cop-monotone winning strategies is unbounded.

As a technical tool to show that DAG-width is bounded in Kelly-width we introduce another notion of monotonicity for DAG-width games that we call *weak monotonicity.* The core of the argument is to show that any weakly monotone strategy in the DAG-width game can be translated into a fully monotone strategy with only a quadratic increase in the number of cops. We can then show that strategies in the games corresponding to Kelly-width can be translated into weakly monotone strategies in the DAG-width game and hence into monotone strategies.

While this relation between Kelly and DAG-width is the most explicit application of this concept of weak monotonicity, we believe that weak monotonicity will have many more applications. In particular, as explained above, the outstanding open problem in the area of digraph searching games is the monotonicity for DAG-width (and Kelly-width) games. These games have been shown to be non-monotone in [17]. More precisely, in [17] it was shown that there are classes of digraphs on which the cops need 4/3 times as many cops for a monotone strategy than for an unrestricted strategy. However, all attempts to use the techniques developed in [17] to show that the monotonicity costs cannot be bounded by any constant, or any function at all, have failed. Our result on weak monotonicity proves that these attempts are doomed to fail as the non-monotone strategy used by cops in the examples in [17] is in fact weakly monotone. We therefore believe that weak monotonicity will prove to be a valuable step towards a solution of the monotonicity problem of DAG-width games. And indeed this was the original motivation for introducing weak monotonicity in [21]. It is worth mentioning that the weakly monotone DAG-width games have a corresponding decomposition. These weak DAG decompositions approximate DAG decompositions and always have size polynomial in the size of the graph.

Our contributions. The main results of this paper are the following.

- We show that deciding the DAG-width of a graph, or equivalently deciding the number of cops needed to win the corresponding monotone graph searching game, is PSPACE-complete.
- We show that there are graphs for which no DAG decomposition of polynomial size exist whose width is at most an additive constant away from the optimal width.
- We refute a conjecture by Safari [23] by showing that the cop-monotonicity costs for the graph searching games corresponding to directed tree width are unbounded. As a consequence, we obtain that D-width is not bounded by any function in the directed tree width. In fact, D-width is not even bounded by any function in the number of cops needed in the cop-monotone directed tree width game. Furthermore, we also show that D-width cannot even be bounded by any function in the DAG-width and in the Kelly-width.
- We show that DAG-width can be bounded by a quadratic function in the Kelly-width. Together with the previous results, we obtain an almost complete classification of the directed width measures proposed in the literature.

2 Preliminaries

We assume familiarity with basic concepts of graph theory and refer to [8] for background. 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 digraphs with a symmetric edge relation. We write $\bar{G}$ for the underlying undirected graph of G. If G is a graph, then $V(G)$ is its set of vertices and $E(G)$ is 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 set $V' \subseteq V(G)$ is denoted $\text{Reach}_G(V')$. If $V' = \{v\}$, we also write $\text{Reach}_G(v)$. A *strongly connected component* of a digraph G is a maximal subgraph C of G which is strongly connected, i.e. between any pair $u, v \in V(C)$ there are directed paths from u to v and from v to u. All components of digraphs considered in this paper will be strong and hence we simply speak of *components*.

2.1 Graph Searching Games

A graph searching game (also known as cops and robber game and pursuit-evasion 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. when the cops move from their current position C to C' and the robber is on v, the robber has exactly the same choice of 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 induces a component of $G - C$.

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

$$\mathrm{Moves}_c(G) := \{\big((C, R), (C, C', R)\big) : (C, R) \in \mathrm{Pos}_c, (C, C', R) \in \mathrm{Pos}_r\}\,.$$

The difference is in the definition of the set of possible robber moves.

Reachability game

In the *reachability game*, we define $\mathrm{Moves}(G)$ as $\mathrm{ReachMoves}(G)$, where

$$\begin{aligned}\mathrm{ReachMoves}(G) &:= \mathrm{Moves}_c(G) \cup \{\big((C, C', R), (C', R')\big) : (C, C', R) \in \mathrm{Pos}_r,\\ &(C', R') \in \mathrm{Pos}_c \text{ and } R' \text{ is a component of } G - C' \text{ such that } R' \subseteq \mathrm{Reach}_{G-(C\cap C')}(R)\}\,.\end{aligned}$$

In other words, 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 $\mathrm{Moves}(G)$ as $\mathrm{CompMoves}(G)$, where

$$\begin{aligned}\mathrm{CompMoves}(G) :=&\mathrm{Moves}_c(G) \cup \{\big((C, C', R), (C', R')\big) : (C, C', R) \in \mathrm{Pos}_r,\\ &(C', R') \in \mathrm{Pos}_c \text{ and } R' \text{ is a component of } G - C' \text{ such that } R\\ &\text{and } R' \text{ are subsets of the same component of } G - (C \cap C')\}\,.\end{aligned}$$

That means, 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 is a set of finite plays. The cops win all plays (C_0, R_0), (C_0, C_1, R_0), $(C_1, R_1), \ldots$ in Mon where $R_i = \emptyset$ for some i (and the play stops here) and the robber wins all other plays. Usually Mon describes cop- or robber-monotonicity: $\mathrm{Mon} \in \{\mathrm{cm}(G) \cup \mathrm{rm}(G)\}$. A play (C_0, R_0), (C_0, C_1, R_0), $(C_1, R_1), \ldots$ is

- in $\mathrm{cm}(G)$, called *cop-monotone,* if for all $i, j, k \geq 0$ with $i < j < k$ we have $C_i \cap C_k \subseteq C_j$,

- in rm(G), called *robber-monotone*, if $R_{i+1} \subseteq R_i$ for all i.

Cop-monotonicity means that the cops never reoccupy vertices. Robber-monotonicity means that once the robber cannot reach a vertex, he will never be able to reach it in the future. 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.** Let Fin be the set of all finite plays. For every digraph G and for

$$X \in \{\mathrm{dtw}, \mathrm{cmdtw}, \mathrm{rmdtw}, \mathrm{nmDAG}, \mathrm{cmDAG}, \mathrm{DAG}\}$$

let $cn_G(X)$ be the minimal number of cops that have a winning strategy in the game $\mathbb{G}_G(X)$ where

- $\mathbb{G}(\mathrm{dtw}, G) := (\mathrm{Pos}(G), \mathrm{CompMoves}(G), \mathrm{Mon} = \mathrm{Fin})$,
- $\mathbb{G}(\mathrm{cmdtw}, G) := (\mathrm{Pos}(G), \mathrm{CompMoves}(G), \mathrm{Mon} = \mathrm{cm}(G))$,
- $\mathbb{G}(\mathrm{rmdtw}, G) := (\mathrm{Pos}(G), \mathrm{CompMoves}(G), \mathrm{Mon} = \mathrm{rm}(G))$,
- $\mathbb{G}(\mathrm{nmDAG}, G) := (\mathrm{Pos}(G), \mathrm{ReachMoves}(G), \mathrm{Mon} = \mathrm{Fin})$,
- $\mathbb{G}(\mathrm{cmDAG}, G) := (\mathrm{Pos}(G), \mathrm{ReachMoves}(G), \mathrm{Mon} = \mathrm{cm}(G))$,
- $\mathbb{G}(\mathrm{DAG}, G) := (\mathrm{Pos}(G), \mathrm{ReachMoves}(G), \mathrm{Mon} = \mathrm{rm}(G))$.

It follows immediately from the definitions that, for all digraphs G,

$$\begin{gathered} cn_G(\mathrm{dtw}) \leq cn_G(\mathrm{cmdtw}), cn_G(\mathrm{rmdtw}) \text{ and} \\ cn_G(\mathrm{cmdtw}), cn_G(\mathrm{rmdtw}) \leq cn_G(\mathrm{nmDAG}) \leq cn_G(\mathrm{DAG}), cn_G(\mathrm{cmDAG})\,. \end{gathered} \tag{1}$$

The number $cn_G(\mathrm{cmdtw}) - cn_G(\mathrm{dtw})$ 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.

2.2 Decompositions and Widths

Most of the games described in Definition 2.1 can be characterised by widths of decompositions of the graphs. In the following let G be an arbitrary graph. For $v, w \in V(G)$ we write $v \leq w$ if $w \in \mathit{Reach}_G(v)$ and $v < w$ if, additionally, $v \neq w$.

Directed tree width [22, 14] was the first generalisation of tree width to digraphs. For $X, Y \subseteq V(G)$ we say that X is Y-normal if X is a union of components of $G - Y$. An *arboreal decomposition* of G is a triple (R, X, W) where R is a directed tree with edges oriented away from the root and $X = \{X_e : e \in E(R)\}$ and $W = \{W_r : r \in V(R)\}$ are collections of sets of vertices of G such that

(i) W is a partition of $V(G)$ into nonempty sets and
(ii) if $e = (t, s) \in E(R)$, then $W_{\geq e}$ is X_e-normal where $W_{\geq e} = \bigcup\{W_r : r \in V(R), r \geq s\}$.

The *width* of (R, X, W) is $\max_{r \in V(R)} |W_r \cup \bigcup_{e \sim r} X_e| - 1$ where $e \sim r$ means that r is incident with e. The *directed tree width* of G is the least width of an arboreal decomposition of G.

DAG-width was defined in [3] and simultaneously in [20]. A *DAG decomposition* of G is a tuple (D, B) where D is a DAG and $B = \{B_d : d \in V(D)\}$ is a set of bags, i.e. subsets of $V(G)$, such that

1. $\bigcup_{d \in V(D)} B_d = V(G)$,
2. for all $a, b, c \in D$, if $a < b < c$, then $B_a \cap B_c \subseteq B_b$,
3. for every root $r \in V(D)$, $\mathrm{Reach}_G(B_{\geq r}) = B_{\geq r}$ where $B_{\geq r} = \bigcup_{r \leq d} B_d$,
4. for each $(a, b) \in E(D)$, $\mathrm{Reach}_{G-(B_a \cap B_b)}(B_{\geq b} \setminus B_a) = B_{\geq b} \setminus B_a$.

The width of (D, B) is $\max_{d \in V(D)} |B_d|$ and its size is $|V(D)|$. The *DAG-width* DAG-w(G) of G is the minimal width of a DAG decomposition of G.

Kelly-width is a complexity measure for digraphs introduced in [13]. Similarly to tree width, Kelly-width can be defined by a decomposition, by a graph searching game or by an elimination order. We choose the latter definition. An *elimination order* $\lhd$ for a graph $G = (V, E)$ is a linear order on V. For a vertex v define $V_{\rhd v} := \{u \in V : v \lhd u\}$. The *support* of a vertex v with respect to $\lhd$ is

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

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

In [23], Safari suggests D-width as another structural complexity measure. Let G be a graph. A *D-decomposition* of G is a pair $(T, (X_t)_{t \in V(T)})$ where T is an undirected tree and $X_t \subseteq V(G)$ for all $t \in V(T)$ is a set of bags such that for all $v \in V(G)$ the set $\{t \in V(T) : v \in X_t\}$ is non-empty and connected in T and for every edge $(s, t) \in E(T)$ and every strongly connected component C of $G - (X_s \cap X_t)$, either $V(C) \subseteq \bigcup_{r \in V(T_s)} X_r$ or $V(C) \subseteq \bigcup_{r \in V(T_t)} X_r$, where T_s, T_t are the two connected components of $T - \{(s, t), (t, s)\}$. The width of $(T, (X_t))$ is $\max_{t \in V(T)} |X_t|$. The *D-width* of G, D-w(G), is the minimum of the widths of all D-decompositions of G. [1]

2.3 Known Relations between Cop Numbers and Widths

We are interested in the question which cop numbers and widths are bounded in terms of which (other) cop numbers and widths. For instance, for DAG-width and Kelly-width we want to know whether there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all graphs G we have DAG-w$(G) \leq f($Kelly-w$(G))$. Besides bounds from the inequalities in (1) the following relations are known.

▶ **Theorem 2.2** ([14, 4, 23] [2]). *Let G be a graph.*
1. $\mathrm{dtw}(G)$, $cn_G(\mathrm{dtw})$ *and* $cn_G(\mathrm{rmdtw})$ *are within factor* 3 *of each other.*
2. $cn_G(\mathrm{DAG}) = cn_G(\mathrm{cmDAG}) = \text{DAG-w}(G)$.
3. $cn_G(\mathrm{dtw}) \leq 2cn_G(\mathrm{cmdtw}) + 1 < \text{D-w}(G)$.

This allows us to call $\mathbb{G}(\mathrm{dtw}, G)$ the directed tree width game and $\mathbb{G}(\mathrm{DAG}, G)$ the *DAG-width game.*

Berwanger et al. present a class of graphs certifying that DAG-width is not bounded in directed tree width. The same holds if we substitute directed tree width by D-width and/or DAG-width by Kelly-width (using the same class of graphs).

▶ **Theorem 2.3** ([4]). *There is a class of graphs G_n such that $cn_{G_n}(\mathrm{dtw}) = cn_G(\mathrm{cmdtw}) = 2$ and $cn_G(\mathrm{rmdtw}) = \text{D-w}(G_n) = 1$, but* DAG-w$(G)$ *and* Kelly-w(G) *are not bounded.*

[1] In [23] the width is $\max_{t \in V(T)} |X_t| - 1$.
[2] The second inequality in (3) is not proven in the cited works, but easy to prove, see Appendix.

3 The Complexity of DAG-width and the DAG-width Game

For all widths W considered in our work except DAG-width it is easy to see that the problem, given a graph G and a natural number k, whether $W(G) \leq k$, is in NP. The reason is that the size of the corresponding decompositions is polynomial in the size of G. Because on graphs with a symmetric edge relation all widths considered here are equal to tree width and tree width is NP-hard, the width measures here are NP-complete. We show that the situation with DAG-width is different, however. It turns out that DAGW, the problem whether DAG-w$(G) \leq k$, is PSPACE-complete and, moreover, for some graphs, there are no decompositions of polynomial size even if we allow a constant additive error in the width.

▶ **Theorem 3.1.** DAGW *is* PSPACE-*complete.*

Proof sketch. The easier part is to show that DAGW is in PSPACE. Due to the robber-monotonicity, the length of every play in $\mathbb{G}(\mathrm{DAG}, G)$ is linear in $|G|$. Hence the winner of the game can be determined in alternating PTIME (by simply simulating the game) and thus in PSPACE.

For the hardness, we reduce QBF, which is PSPACE-complete, to DAGW. A *quantified boolean formula* φ is of the form $\varphi = Q_1 X_1 \dots Q_r X_r \psi(X_1, \dots, X_r)$ where Q_i is either $\forall$ or $\exists$ and ψ is a propositional formula in CNF with variables from $\mathcal{X} = \{X_1, \dots, X_r\}$. A formula $\exists X \psi(X)$ is true if there is a value $\beta(X) \in \{0,1\}$ for X such that ψ is true. A formula $\forall X \psi(X)$ is true if for both values $\beta(X) \in \{0,1\}$ for X, ψ is true.

It is very well known that deciding QBF, the problem whether a given quantified formula is true, is PSPACE-complete. The idea of our reduction is to simulate the choice of a truth value for a variable by a quantifier in the game $\mathbb{G}(\mathrm{DAG}, S_\varphi)$, where S_φ is some graph constructed from φ. The choices are stored as vertices occupied by cops using the monotonicity of the game. These cops only reflect the history of the play and do not change the flow of the remaining play.

Let $\varphi = Q_1 X_1\ Q_2 X_2 \dots Q_r X_r\ \psi(X_1, \dots, X_r)$ be a quantified boolean formula. The graph S_φ is constructed inductively level by level, each of which corresponds to some X_i.

If φ has no variables, then if φ is true, S_φ is a single vertex, and if φ is false, S_φ is a 2-clique. Then one cop wins if, and only if, φ is true. Otherwise we start the construction of S_φ with a gadget F_ψ. It has a vertex v and for every clause $C = L_1 \vee L_2 \vee \dots L_{r(C)}$ an $r(C)$-clique K^C with vertices v_1^C, v_2^C, …, $v_{r(C)}^C$. The edges go from v to every vertex of K^C and back, i.e. we have edges (v, v_i^C) and (v_i^C, v) for all clauses C and all $i \in \{1, \dots, r(C)\}$.

For $j = r, r-1, \dots, 1$ we construct graphs S_φ^j such that $S_\varphi^1 = S_\varphi$. For convenience, let $S_\varphi^{r+1} = F_\psi$.

Assume that S_φ^{j+1} has already been constructed. Then S_φ^j is the following graph. There are two cases. If $Q_j = \exists$, then the vertex set is

$$V(S_\varphi^j) = V_\exists(j) = V(S_\varphi^{j+1}) \cup A(j) \cup B(j) \cup C_0(j) \cup C_1(j) \cup M(j) \cup D(j) \cup \{c_0(j), c_1(j)\}$$

where $|A(j)| = |B(j)| = |D(j)| = 2, |C_i(1)| = |M(1)| = 4, |C_i(k+1)| = |M(k+1)| = |M(k)| + 3$ for all $k \in \{2, \dots, j-1\}$ and $i \in \{0,1\}$. Furthermore, $B(j) = \{b_0(j), b_1(j)\}$. We set $N(j) = M(j) \cup D(j)$.

The set of edges is

$$E(S^j_\varphi) = E(S^{j+1}_\varphi) \cup \binom{N(j)}{2} \cup \bigcup_{i=0}^{1} \binom{C_i(j)}{2} \cup \binom{A(j)}{2}$$
$$\cup \bigcup_{i=0}^{1} \Big(\big(N(j) \times \{c_i(j)\}\big) \cup \big(\{c_i(j)\} \times C_i(j)\big) \cup \big(C_i(j) \times D(j)\big) \cup \big(C_i(j) \times \{b_i(j)\}\big) \Big)$$
$$\cup \big(B(j) \times A(j)\big) \cup \big(A(j) \times B(j)\big) \cup \big(A(j) \times M(j)\big)$$
$$\cup \big(N(j) \times V(S^{j+1}_\varphi)\big) \cup \big(A(j) \times V(S^{j+1}_\varphi)\big) \cup \big(V(S^{j+1}_\varphi) \times A(j)\big) \cup E(j)\,.$$

Hereby, for a set X, the notation $\binom{X}{2}$ means $\{(a, b) \in X^2 : a \neq b\}$ and $E(j)$ is the set of edges connecting F_ψ to the new level defined as follows. Let K^C be a clique in F_ψ corresponding to the clause $C = L_1 \vee \ldots \vee L_r$. If $X_j = L_i$, then $(v^C_i, b_1(j)) \in E(j)$. If $\neg X_j = L_i$, then $(v^C_i, b_0(j)) \in E(j)$. Otherwise (i.e. if X_j does not appear in C) $\{(v^C_i, b_0(j)), (v^C_i, b_1(j))\} \subseteq E(j)$.

In the second case $Q_j = \forall$. Then $V(S_\varphi(j)) = V_\forall(j) = V_\exists(j) \setminus \{c_0(j), c_1(j)\}$ and the edges are as in an existential level (including edges connecting the level and F_ψ), but edges containing $c_i(j)$ are replaced by edges $\bigcup_{i=0}^{1} N(j) \times C_i(j)$. In other words, the paths that lead from $N(j)$ to $C_i(j)$ through $c_i(j)$ are replaced by direct edges.

One can show that $r + 1$ cops win on S_φ if, and only if, the formula φ is true. The main ingredient of the proof is that in the cops and robber game on S_φ, the cops can expel the robber from a level ℓ only in one way up to irrelevant changes. Hereby, exactly $\ell - 3$ cops remain free for use in the next level $\ell - 3$. They occupy $N(\ell)$, the robber goes to one of the $C_i(\ell)$ and a cop is placed to $b_i(\ell)$. If now the robber remains in $C_i(\ell)$, he is captured there by the cops from $M(\ell)$, so he goes to $A(\ell)$ or to the next level. In any case the cops from $D(\ell)$ move to $A(\ell)$ and the robber is in the next level $\ell - 3$. Note that the cops occupy $A(\ell)$ (blocking the robber in lower levels) and one vertex from $B(\ell)$. This one vertex encodes the choice for the value of the variable from φ that corresponds to that level. In universal levels it is the robber who makes the choice and in the existential these are the cops.

If the level is universal, the robber determines which vertex from $B(\ell)$ will be occupied by deciding in which $C_i(\ell)$ he goes after the cops occupy $N(\ell)$. In the existential level, the cops can determine in which $C_i(\ell)$ the robber must go. If they want $b_{1-i}(\ell)$ to be occupied when the robber leaves level ℓ, they place a cop on $c_i(\ell)$ before occupying $N(\ell)$. Then the cops expel the robber from $C_i(\ell)$ if he is there and occupy $N(\ell)$. The robber goes to $C_{1-i}(\ell)$ (all paths to $C_i(\ell)$ are blocked) or directly to $A(\ell) \cup B(\ell) \cup S^{\ell-3}_\varphi$. In any case the cop from $c_i(\ell)$ moves to $b_{1-i}(\ell)$. If the robber was in $C_{1-i}(\ell)$ and remains there, he is captured by the cops from $M(\ell)$ as before, so after $b_{1-i}(\ell)$ is occupied, the robber is in $A(\ell)$ and after the cops from $D(\ell)$ occupy $A(\ell)$, he is in the next level.

When the robber leaves the last level and proceeds to F_ψ, one cop remains free and goes to v. The robber chooses a clique K^C corresponding to the clause C in ψ. At this point, the value for X_j from C is $\alpha(X_j) = i$ if and only if a cop occupies $b_i(j)$. Furthermore, the construction of edges between F_ψ and the levels guarantees that $\alpha \models C$ if and only if the cop from $B(j)$ can be reused without violating robber-monotonicity. Finally, the cops capture the robber in K^C if and only if they have one free cop. Summing up, the cops win if and only if φ is true. ◀

We can change the construction of S_φ to obtain graphs that have no polynomial size DAG decomposition of width that differs from the optimal one in at most a fixed additive constant. We replace F_ψ in S_φ by a single vertex, make every level universal and adjust the sizes of $A(\ell)$, $B(\ell)$ and $D(\ell)$ by setting $|A(\ell)| = |B(\ell)| = |D(\ell)| = \lfloor \frac{\ell}{\log \ell} \rfloor$. Then a careful calculation of used cops proves the following theorem.

▶ **Theorem 3.2.** *There is no polynomial size approximation of an optimal DAG decomposition of $G_n(s,t)$ with an additive constant error.*

4 Comparing Width Measures with Respect to Generality

By Theorem 2.2, directed tree width and the robber-monotone variant of the corresponding game are bounded in each other. One would expect that the same holds for the cop-monotone variant. This was implicitly assumed by Safari in [23] who conjectured that D-width and directed tree width are the same. Note that by Theorem 2.2, $s \cdot cn_G(\mathrm{dtw}) + 1 \leq \text{D-w}(G)$, so if $\mathrm{dtw}(G) = \text{D-w}(G)$, then the cop-monotonicity cost for directed tree width is zero. We show, however, that it is not only positive, but, moreover, cannot be bounded by any function.

▶ **Theorem 4.1.** *There is a class $\{G_n : n > 2\}$ of graphs such that for all $n > 2$, $cn_G(\mathrm{dtw}) = cn_{G_n}(\mathrm{rmdtw}) \leq 4$ and $cn_{G_n}(\mathrm{cmdtw}) \geq n$.*

Proof. Let $n > 2$. We inductively define a sequence of graphs G_n^m and sets of marked vertices $M(G_n^m) \subseteq V(G_n^m)$ for $m \in \{1, \ldots, n+1\}$. 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. Let T_ℓ^d denote the complete undirected tree of branching degree d and depth ℓ (the depth is the maximum number of vertices on the path from the root to a leaf). 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 $s \in \{1, \ldots, (n+1)^{n+2}\}$. The graph G_n^{m+1} is the disjoint union of T_{n+2}^{n+1} and $n \cdot (n+1)^{n+2}$ copies $H_j^{m+1}(v_s)$ of G_n^m where $j \in \{1, \ldots, n\}$ and $s \in \{1, \ldots, (n+1)^{n+2}\}$ plus some additional edges which we describe next. We denote the subgraph of G_n^{m+1} induced by T_{n+2}^{n+1} by $T(G_n^{m+1})$ and the root of $H_j^{m+1}(v_s)$ by $r(H_j^{m+1}(v_s))$ for all m, j and s.

For every leaf $v \in \{v_s : 1 \leq s \leq (n+1)^{n+2}\}$ of $T(G_n^{m+1})$ there is an undirected edge from v to the root of $H_i^{m+1}(v)$. Let $x_i^{m+1}(v)$ be the ith 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 $M(G_n^{m+1}) := V(T(G_n^{m+1}))$.

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 cannot leave $H_j^i(v)$ as he cannot re-enter $H_j^i(v)$.

The cops chase the robber from the root of G_n downwards. In $T(G_n)$, two cops suffice for that. Consider a position where the cops just expelled the robber from $T(G_n)$. The robber is in some $H_j^n(v)$ and the cops occupy v and its predecessor w. Now the cop from w goes to $x_j^n(v)$ (here non-cop-monotonicity occurs) and a third cop occupies $r(H_j^n(v))$. The cop on $r(H_j^n(v))$ together with the cop on $x_j^n(v)$ block all paths from $T(G_n)$ to the robber in $H_j^n(v) - r(H_j^n(v))$, so the cops on v and w are not needed any more. These two cops chase the robber down the tree further, while the other cops remains on $r(H_j^n(v))$ and $x_j^n(v)$.

In general, assume for some $i < n$, j and v (j and v are new), the robber is blocked in $H_j^i(v)$ by cops on v, on $r(H_{j'}^{i+1}(v'))$ and on $x_j^{i+1}(v)$. Hereby j' and v' are such that $r(H_{j'}^{i+1}(v'))$ is on the path from $r(H_j^i(v))$ to the root of G_n. Now the cop from the predecessor of v goes to $x_j^{i+1}(v)$ (again non-monotonicity occurs). Then the cop from $r(H_{j'}^{i+1}(v'))$ goes to $r(H_j^i(v))$.

These two cops block all paths from $G_n - H^i_j(v)$ to $H^i_j(v) - r(H^i_j(v))$. Hence the other two cops can chase the robber down the tree further. Finally the robber is captured in some leaf of G_n.

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 $T(G_n)$. 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 that the number of cops, there is a vertex $x^n_{i_n}(w_n)$ that has been left by a cop. When a cop occupies w_n, the robber goes to $G^n_{i_n}(w_n)$. Now on $G^n_{i_n}(w_n)$ (which is isomorphic to G^{n-1}_n) the robber plays in the same way as on G_n and so on recursively for each m on $G^m_{i_m}(w_m)$. 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 $T(G^j_{i_j}(w_j))$. Thus there is a cop free path from v to w_j, then to $x^j_{i_j}(w_j)$ within $T(G^j_{i_j}(w_j))$ and then via $x^{j-1}_{i_{j-1}}(w_{j-1})$, $x^{j-2}_{i_{j-2}}(w_{j-2})$, …, $x^2_{i_2}(w_2)$ back to v. Note that all those x-vertices are not occupied by cops by construction of the robber strategy. Thus the robber can return to w_j and play from w_j as before. In this way the robber will never be captured. ◀

▶ **Corollary 4.2.** *D-width is not bounded in directed tree width.*

There is yet another reason why Corollary 4.2 holds. In the definition of D-width we have the condition that the set of bags containing a vertex v is connected in the decomposition tree. This implies cop-monotonicity in the directed tree width game. Moreover, this forbids the existence of two distinct plays such that the cops are placed on v in both plays, but not in their common prefix. However, one can construct graphs where this restriction leads to an unbounded blow up of the number of needed cops. As the DAG-width and the Kelly-width of those graphs are bounded, we obtain the following theorem[3].

▶ **Theorem 4.3.** D-w(G) *is bounded neither in* cn_G(cmdtw)*, nor in* DAG-w(G)*, nor in* Kelly-w(G)*. More precisely, 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* Kelly-w$(G_n) = 4$*, but* D-w$(G_n) \geq n$*.*

4.1 Kelly-width is Bounded in DAG-width

We show that DAG-width is bounded in Kelly-width by a quadratic function.

▶ **Theorem 4.4.** *If* Kelly-w$(G) = k+1$*, then* DAG-w$(G) \leq 72k^2 + 42k + 18$.

In order to prove this we introduce a weaker notion of robber-monotonicity for the DAG-width games. Then we show that with a quadratic number of additional cops one can turn a winning cop strategy for the game with weak monotonicity into a winning strategy for the game with strong (i.e. usual) monotonicity. By a construction from [13], if Kelly-w$(G) = k$,

[3] See the appendix for a proof.

$$\begin{array}{ccccccc} \mathrm{dtw} & < & \mathrm{cmdtw} & < & \text{DAG-w} & \leq & \text{K-w} \\ \parallel & & & & & & \\ \mathrm{rmdtw} & & & \rotatebox{-45}{$<$} & \rotatebox{-90}{$\leq$} & \rotatebox{-135}{$\lessgtr$} & \\ & & & & \text{D-w} & & \end{array}$$

Figure 1 The boundedness relation between different measures. "=" means mutually bounded, "<" means bounded only in one direction, "≤" at least in one direction, "≶" not bounded in any direction.

then $2k - 1$ cops have a (possibly non-monotone) winning strategy in the DAG-width game. We observe that this strategy is, in fact, weakly monotone and thus can be converted into a strongly monotone one.

Weak monotonicity relaxes the winning condition for the cops, so that they win more plays. Formally, for a digraph G we define the set $\mathrm{wm}(G)$ as the set of all finite plays $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \ldots$ 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 $(C_0, R_0), (C_0, C_1, R_0), (C_1, R_1), \ldots$ 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 $\mathrm{wm}(G)$ is the set of all weakly monotone plays on G. The *weakly monotone game* is the game defined by $\mathbb{G}(\mathrm{wmDAGW}, G) = (\mathrm{Pos}(G), \mathrm{ReachMoves}(G), \mathrm{Mon} = \mathrm{wm}(G))$.

▶ **Lemma 4.5.** $cn_G(\mathrm{wmDAG}) \leq 18 \cdot cn_G(\mathrm{DAG})^2 + 3 \cdot cn_G(\mathrm{DAG})$.

As, clearly, $cn_G(\mathrm{DAG}) \leq cn_G(\mathrm{wmDAG})$, we obtain that weakening the monotonicity in the DAG-width game does not change the boundedness of $cn_G(\mathrm{DAG})$.

We remark that it is possible to define a decomposition corresponding to the weakly monotone game. Unlike DAG decompositions a *weak* DAG decomposition is always of polynomial size in the size of G. Hence we have an NP-algorithm that computes a succinct representation of a DAG decomposition whose width is at most quadratically worse than the optimum.

▶ **Lemma 4.6.** *If* $\text{Kelly-w}(G) = k + 1$*, then* $cn_G(\mathrm{wmDAG}) \leq 2k + 1$*.*

The other direction, i.e. whether DAG-width is bounded in Kelly-width, is the last open question in our scheme.

We obtain the picture shown in Figure 1. The only blank spot is the strictness of the inequality $\text{DAG-w}(G) \leq \text{Kelly-w}(G)$, i.e. whether Kelly-width is a function of DAG-width. It was conjectured that Kelly-width and DAG-width differ by at most a constant factor [13, Conjecture 30]. However, methods we used to show a weaker version of one direction of the conjecture do not seem to apply for the other direction.

References

1 I. Adler. Directed tree-width examples. *J. Comb. Theory, Ser. B*, 97(5):718–725, 2007.
2 J. Barát. Directed Path-width and Monotonicity in Digraph Searching. *Graphs and Comb.*, 22(2), 2006.

3 D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-width and parity games. In *STACS '06*, volume 3884 of *LNCS*. Springer, 2006.

4 D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, and J. Obdržálek. The DAG-width of directed graphs. *J. Comb. Theory*, 102(4):900–923, 2012.

5 D. Bienstock and P. Seymour. Monotonicity in graph searching. *Journal of Algorithms*, 12(2):239–245, 1991.

6 N. Dendris, L. Kirousis, and D. Thilikos. Fugitive search games on graphs and related parameters. *Theoretical Computer Science*, 172(1–2):233 – 254, 1997.

7 D. Dereniowski. From pathwidth to connected pathwidth. In *28th Symposium on Theoretical Aspects of Computer Science (STACS)*, 2011.

8 R. Diestel. *Graph Theory, 4th Edition*. Springer, 2012.

9 F. Fomin, P. Golovach, and D. Thilikos. Approximation algorithms for domination search. In Klaus Jansen and Roberto Solis-Oba, editors, *Approximation and Online Algorithms (WAOA)*, volume 6534 of *Lecture Notes in Computer Science*, pages 130–141. Springer Berlin / Heidelberg, 2011.

10 F. Fomin, D. Kratsch, and H. Müller. On the domination search number. *Discrete Applied Mathematics*, 127(3):565–580, 2003.

11 F. Fomin and D. Thilikos. On the monotonicity of games generated by symmetric submodular functions. *Discrete Applied Mathematics*, 131(2):323 – 335, 2003.

12 F. Fomin and D. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.

13 P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3), 2008.

14 T. Johnson, N. Robertson, P. Seymour, and R. Thomas. Directed Tree-Width. *J. Comb. Theory, Ser. B*, 82(1), 2001.

15 S. 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.

16 S. Kreutzer and S. Ordyniak. Distance-d-domination games. In *34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, 2009.

17 S. Kreutzer and S. Ordyniak. Digraph decompositions and monotonicity in digraph searching. *Theor. Comput. Sci.*, 412(35):4688–4703, 2011.

18 A. S. LaPaugh. Recontamination does not help to search a graph. *Journal of the ACM*, 40:224–245, 1993.

19 F. Mazoit and N. Nisse. Monotonicity of non-deterministic graph searching. *Theor. Comput. Sci.*, 399(3):169–178, 2008.

20 J. Obdržálek. *Algorithmic analysis of parity games.* PhD thesis, School of Informatics, University of Edinburgh, 2006.

21 R. Rabinovich. *Graph Complexity Measures and Monotonicity.* PhD thesis, RWTH Aachen University, 2013.

22 B. Reed. Introducing directed tree-width. *Electronic Notes in Discrete Mathematics*, 3:222 – 229, 1999.

23 M. A. 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.

24 P. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory Ser. B*, 58(1), 1993.

25 Y. Stamatiou and D. Thilikos. Monotonicity and inert fugitive search games. In *6th Twente Workshop on Graphs and Combinatorial Optimization CTW 1999.* University of Twente, Enschede, 1999.

26 B. Yang and Y. Cao. Monotonicity of strong searching on digraphs. *J. Comb. Optim.*, 14(4):411–425, 2007.

27 B. Yang and Y. Cao. On the monotonicity of weak searching. In *COCOON*, pages 52–61, 2008.