

LIFO-search: A min-max theorem and a searching game for cycle-rank and tree-depth*

Archontia C. Giannopoulou^{†‡} Paul Hunter^{§¶}
Dimitrios M. Thilikos[†]

Abstract

We introduce a variant of the classic node search game called LIFO-search where searchers are assigned different numbers. The additional rule is that a searcher can be removed only if no searchers of lower rank are in the graph at that moment. We show that all common variations of the game require the same number of searchers. We then introduce the notion of (strong) shelters in (di)graphs and prove a min-max theorem implying their equivalence to the cycle-rank/tree-depth parameter in (di)graphs. As (strong) shelters provide escape strategies for the fugitive, this implies that the LIFO-search game is monotone and that the LIFO-search parameter is equivalent to the one of cycle-rank/tree-depth in (di)graphs.

1 Introduction

Graph searching games are increasingly becoming a popular way to characterize, and even define, practical graph parameters. There are many advantages to a characterization by graph searching games: it provides a useful intuition which can assist in constructing more general or more specific parameters; it gives insights into relations with other, similarly characterized parameters; and it is particularly useful from an algorithmic perspective as many parameters associated with such games are both structurally robust and efficiently computable.

*Part of this research was presented during Dagstuhl Seminar 11071 (13.02.11 – 18.02.11) on *Theory and Applications of Graph Searching Problems* [11]. Preliminary versions of this paper appeared in EUROCOMB 2011 [13] and in FCT 2011 [15].

[†]Department of Mathematics, National and Kapodistrian University of Athens. Emails: {arcgian, sedthilk}@math.uoa.gr

[‡]Supported by a grant of the Special Account for Research Grants of the National and Kapodistrian University of Athens (project code: 70/4/10311).

[§]Department of Computer Science, University of Oxford. Email: paul.hunter@cs.ox.ac.uk

[¶]Supported by Model Checking Real-Time Systems project, EPSRC grant ref. BLRQEK

One of the most common graph searching games is the node-search game. In this game several searchers and one fugitive occupy vertices of the graph and make simultaneous moves. The (omniscient) fugitive moves along searcher-free paths of arbitrary length whereas the searchers' movements are not constrained by the topology of the graph. The goal of the game is to minimize the number of searchers required to capture the fugitive by cornering him in some part of the graph and placing a searcher on the same vertex. This game has been extensively studied [7] and several important graph parameters such as treewidth [28] and pathwidth [18] can be characterized by natural variants of this game. One variation frequently used, indeed the one which separates treewidth and pathwidth, is whether the location of the fugitive is known or unknown to the searchers. Another common variation is whether the searchers use a monotone or a non-monotone searching strategy. Monotone search strategies lead to algorithmically useful decompositions, whereas non-monotone strategies are more robust under graph operations and hence reflect structural properties, thus showing that monotone strategies require no more searchers than non-monotone strategies is an important and common question in the area. Whilst node-search games on undirected graphs tend to enjoy monotonicity [4, 28, 20], on digraphs the situation is much less clear [2, 1, 19].

Node-search games naturally extend to digraphs. However, in the translation another variation arises depending on how one views the constraints on the movement of the fugitive. One interpretation is that in the undirected case the fugitive moves along paths, so the natural translation would be to have the fugitive move along directed paths. Another view is that the fugitive moves to some other vertex in the same connected component, and here the natural translation would be to have the fugitive move within the same strongly connected component. Both interpretations have been studied in the literature, the former giving characterizations of parameters such as DAG-width [3, 26] and directed pathwidth [2] and the latter giving a characterization of directed treewidth [16].

We define a variant of the node-search game in which only the most recently placed searchers may be removed; that is, the searchers must move in a last-in-first-out (LIFO) manner and we show that the minimum number of searchers required to capture a fugitive on a (di)graph with a LIFO-search is independent of:

- Whether the fugitive is invisible or visible,
- Whether the searchers use a monotone or non-monotone search, and
- Whether the fugitive is restricted to moving in searcher-free strongly con-

nected components or along searcher-free directed paths.

This result is somewhat surprising: in the standard node-search game these options give rise to quite different parameters [2, 3, 19].

We show that on digraphs the LIFO-search game characterizes a pre-existing measure, cycle-rank – a generalization of tree-depth to digraphs (though as the definition of cycle-rank predates tree-depth by several decades, it is perhaps more correct to say that tree-depth is an analogue of cycle-rank on undirected graphs). The cycle-rank of a digraph is an important parameter relating digraph complexity to other areas such as regular language complexity and asymmetric matrix factorization. It was defined by Eggan [9], where it was shown to be a critical parameter for determining the star-height of regular languages, and interest in it as an important digraph parameter, especially from an algorithmic perspective, has recently been rekindled by the success of tree-depth [10, 14, 12].

It is well known that tree-depth can also be characterized by a node-search game where a visible fugitive plays against searchers that are only placed and never moved [12]. In that paper, Ganian et al. considered one extension of this game to digraphs. Here we consider the other natural extension, where the visible fugitive moves in strongly connected sets, and show that it also characterizes cycle-rank. From the above, we also obtain that the LIFO-search parameter is equivalent to the one of tree-depth.

Our final result uses these graph searching characterizations to define a dual parameter that characterizes structural obstructions for cycle-rank. We consider two kinds of obstructions. The first one is obtained from defining the notion of strong shelters. The second is motivated by the havens of [16]. Both the strong shelters and the havens define simplified strategies for the fugitive. The game characterization then implies that these structural features are necessarily present when the cycle-rank of a graph is large. By showing that such strategies are also sufficient for the fugitive, we obtain a rare instance of an exact min-max theorem relating digraph parameters. This also implies that the notion of shelters when transferred to simple graphs characterizes structural obstructions for tree-depth.

The results of this paper can be summarized with the following characterizations of cycle-rank and tree-depth respectively.

Theorem. *Let G be a digraph, and k a positive integer. The following are equivalent:*

- (i) G has cycle-rank $\leq k - 1$,
- (ii) On G , k searchers can capture a fugitive with a LIFO-search strategy,

- (iii) On G , k searchers can capture a visible fugitive restricted to moving in strongly connected sets with a searcher-stationary search strategy,
- (iv) G has no LIFO-haven of order $> k$, and
- (v) G has no strong shelter of thickness $> k$.

Theorem. *Let G be a non-empty graph and k be a positive integer. Then the following are equivalent.*

- (i) G has tree-depth at most k
- (ii) there is a monotone LIFO-search strategy in G of cost at most k that captures an invisible and agile fugitive.
- (iii) there is a LIFO-search strategy in G of cost at most k that captures an invisible and agile fugitive.
- (iv) every shelter in G has thickness at most k .
- (v) there is a monotone LIFO-search strategy in G using k searchers against a visible and agile fugitive.

The paper is organised as follows. In Section 2 we recall the definitions and notation that we use throughout the paper. In Section 3 we define the LIFO-search and searcher-stationary games and show that they characterize cycle-rank. In Section 4 we prove the min-max theorem for cycle-rank. In Section 5 we consider simple graphs and argue that our results imply the existence of a min-max theorem for LIFO-search and that the LIFO-search parameter is equivalent to the one of tree-depth, and in Section 6 we conclude with a discussion on further research and open problems.

2 Preliminaries

All (di)graphs in this paper are finite, simple, (directed) and without self-loops, although the results readily extend to multigraphs with self-loops. For simplicity, we also assume that all (di)graphs contain at least one vertex unless explicitly mentioned. We use standard notation and terminology, in particular $V(G)$ and $E(G)$ denote the sets of vertices and edges respectively of a (di)graph G and between (di)graphs, \subseteq denotes the subgraph relation. We will often interchange an induced subgraph with the set of vertices which defines it, in particular strongly connected sets of vertices are sets of vertices that induce a strongly connected subgraph, and we will often view strongly connected components as sets of vertices. Given a (di)graph G and a set of vertices $X \subseteq V(G)$,

we use $G \setminus X$ to denote the subgraph of G induced by $V(G) \setminus X$. An *initial component* of a digraph G is a strongly connected component C with no edges from $G \setminus C$ to C and $H \subseteq G$ is *successor-closed* if there are no edges in G from H to $G \setminus H$.

Given a finite set V , we use $\mathcal{P}(V)$ to denote its powerset, V^* to denote the set of finite words over V , and $V^{<k}$ to denote the set of words over V of length $< k$. We use ϵ to denote the empty word and \cdot or juxtaposition to denote concatenation. For $X, Y \in V^*$ we write $X \preceq Y$ if X is a prefix of Y , that is, if there exists a word $Z \in V^*$ such that $Y = X \cdot Z$. For $X = a_1 a_2 \cdots a_n \in V^*$, we use $|X|$ to denote the length of X , and $\{X\}$ to denote the set $\{a_1, a_2, \dots, a_n\}$. Given two sets A and B we use $A \Delta B$ to denote their symmetric difference, that is $A \Delta B = (A \cup B) \setminus (A \cap B)$. Given a set $\mathcal{S} \subseteq \mathcal{P}(V)$ of subsets of V , a \subseteq -*chain* is a subset $\{X_1, \dots, X_n\} \subseteq \mathcal{S}$ such that $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_n$. If there is no $Y \in \mathcal{S}$ such that $Y \subset X_1$, $X_i \subset Y \subset X_{i+1}$ for some i , or $X_n \subset Y$, then $\{X_1, \dots, X_n\}$ is a *maximal \subseteq -chain*.

Definition 1 (Cycle-rank). The *cycle-rank* of a digraph G , $\text{cr}(G)$, is defined as follows:

- If G is acyclic then $\text{cr}(G) = 0$.
- If G is strongly connected then $\text{cr}(G) = 1 + \min_{v \in V(G)} \text{cr}(G \setminus \{v\})$.
- Otherwise $\text{cr}(G) = \max_H \text{cr}(H)$ where the maximum is taken over all strongly connected components H of G .

We postpone the definition of tree-depth until Section 5.

3 Searching games for cycle-rank

We begin by formally defining the LIFO-search game, and its variants, for digraphs. Each variation of the LIFO-search game gives rise to a digraph parameter corresponding to the minimum number of searchers required to capture the fugitive under the given restrictions. The main result of this section is that for any digraph all these parameters are equal. Furthermore, we show they are all equal to one more than the cycle-rank of the digraph.

3.1 LIFO-search for digraphs

In summary, for the graph searching game in which we are interested the fugitive can run along searcher-free directed paths of any length, the searchers can move to any vertex in the graph, and the fugitive moves whilst the searchers are

relocating. The searcher's strategy may apply two types of moves in each step: either placement of a searcher on a vertex or removal of a searcher from a vertex with the restriction that only the most recently placed searchers may be removed. If a searcher is placed on the fugitive then he is captured and the searchers win, otherwise the fugitive wins. The goal is to determine the minimum number of searchers required to capture the fugitive. The variants we are primarily interested in are whether the fugitive is visible or invisible, and whether or not the fugitive must stay within the same strongly connected component when he is moving. As our fundamental definitions are dependent on these two options, we define four *game variants*: \mathbf{i} , \mathbf{isc} , \mathbf{v} , \mathbf{vsc} , corresponding to the visibility of the fugitive and whether he is constrained to moving within strongly connected components, that is, \mathbf{i} and \mathbf{v} correspond to an invisible and a visible fugitive respectively while \mathbf{isc} and \mathbf{vsc} correspond to an invisible and visible fugitive as above with the addition that the fugitives move inside the same strongly connected component. Then we parameterize our definitions by these variants.

Let us fix a digraph G . A position in a LIFO-search on G is a pair (X, R) where $X \in V(G)^*$ and R is a (possibly empty) induced subgraph of $G \setminus \{X\}$. Intuitively X represents the position and ordered placement of the searchers and R represents the part of G that the fugitive can reach (in the visible case) or the set of vertices where he might possibly be located (in the invisible case). We say a position (X, R) is an *\mathbf{i} -position* if R is successor-closed; an *\mathbf{isc} -position* if it is a union of strongly connected components of $G \setminus \{X\}$; a *\mathbf{v} -position* if R is successor-closed and has a unique initial component; and a *\mathbf{vsc} -position* if R is a strongly connected component of $G \setminus \{X\}$.

To reflect how the game transitions to a new position during a round of the game we say, for $\mathbf{gv} \in \{\mathbf{i}, \mathbf{isc}, \mathbf{v}, \mathbf{vsc}\}$, a \mathbf{gv} -position (X', R') is a *\mathbf{gv} -successor* of (X, R) if either $X \preceq X'$ or $X' \preceq X$, with $|\{X\} \Delta \{X'\}| = 1$, and

- (for $\mathbf{gv} \in \{\mathbf{i}, \mathbf{v}\}$) For every $v' \in V(R')$ there is a $v \in V(R)$ and a directed path in $G \setminus (\{X\} \cap \{X'\})$ from v to v' , or
- (for $\mathbf{gv} \in \{\mathbf{isc}, \mathbf{vsc}\}$) For every $v' \in V(R')$ there is a $v \in V(R)$ such that v and v' are contained in the same strongly connected component of $G \setminus (\{X\} \cap \{X'\})$.

Ideally we would like to assume games start from (ϵ, G) , however in the visible variants of the game this might not be a legitimate position. Thus, for $\mathbf{gv} \in \{\mathbf{v}, \mathbf{vsc}\}$, if (ϵ, G) is not a \mathbf{gv} -position we include it as a special case, and set as its \mathbf{gv} -successors all \mathbf{gv} -positions of the form (ϵ, R) . We observe that in all variants, the successor relation is monotone in the sense that if (X, R) and

(X, S) are positions with $S \subseteq R$ and (X', S') is a successor of (X, S) , then there is a successor (X', R') of (X, R) with $S' \subseteq R'$.

For $\mathbf{gv} \in \{\mathbf{i}, \mathbf{isc}, \mathbf{v}, \mathbf{vsc}\}$, a (**gv-LIFO**-)search in a digraph G from **gv**-position (X, R) is a (finite or infinite) sequence of **gv**-positions $(X, R) = (X_0, R_0), (X_1, R_1), \dots$ where for all $i \geq 0$, (X_{i+1}, R_{i+1}) is a **gv**-successor of (X_i, R_i) . A LIFO-search is *complete* if either $R_n = \emptyset$ for some n , or it is infinite. We observe that if $R_n = \emptyset$, then $R_{n'} = \emptyset$ for all $n' \geq n$.

We say that a complete LIFO-search is *winning for the searchers* if $R_n = \emptyset$ for some n , otherwise it is winning for the fugitive. A complete LIFO-search from (ϵ, G) is *monotone* if $R_{i+1} \subseteq R_i$ for all i , that is, the fugitive cannot reside in already searched areas of the digraph, it is *searcher-stationary* if $X_i \preceq X_{i+1}$ for all i where $R_i \neq \emptyset$; and it *uses at most k searchers* if $|X_i| \leq k$ for all i .

Whilst a complete LIFO-search from (ϵ, G) describes a single run of the game, we are more interested in the cases where one of the players (particularly the searchers) can always force a win, no matter what the other player chooses to do. For this, we introduce the notion of a strategy. For $\mathbf{gv} \in \{\mathbf{i}, \mathbf{isc}, \mathbf{v}, \mathbf{vsc}\}$, a (*searcher*) **gv-strategy** is a (partial¹) function σ from the set of all **gv**-positions to $V(G)^*$ such that for all (X, R) , $\sigma(X, R)$ is the first component of a **gv**-successor of (X, R) ; so with the possible exception of $(X, R) = (\epsilon, G)$, either $\sigma(X, R) \preceq X$ or $X \preceq \sigma(X, R)$. A **gv-LIFO**-search $(X_0, R_0), (X_1, R_1), \dots$ is *consistent* with a **gv**-strategy σ if $X_{i+1} = \sigma(X_i, R_i)$ for all $i \geq 0$. A strategy σ is winning from (X, R) if all complete LIFO-searches from (X, R) consistent with σ are winning for the searchers. Likewise, a strategy is monotone (searcher-stationary, uses at most k searchers) if all consistent complete LIFO-searches from (ϵ, G) are monotone (searcher-stationary, use at most k searchers respectively). We say k searchers can capture a fugitive on G in the **gv**-game with a (monotone) LIFO-search strategy if there is a (monotone) **gv**-strategy that uses at most k searchers and is winning for the searchers from (ϵ, G) .

For $\mathbf{gv} \in \{\mathbf{i}, \mathbf{isc}, \mathbf{v}, \mathbf{vsc}\}$, we define the (monotone) **gv-LIFO**-search number of G , $\text{LIFO}^{\mathbf{gv}}(G)$ ($\text{LIFO}^{\text{mgv}}(G)$), as the minimum k for which there is a (monotone) winning **gv**-strategy that uses at most k searchers. We also define the visible, strongly connected, searcher-stationary search number of G , $\text{SS}^{\mathbf{vsc}}(G)$ as the minimum k for which there is a searcher-stationary winning **vsc**-strategy that uses at most k searchers.

In Section 4 we will also consider fugitive **gv**-strategies: a partial function ρ from $V(G)^* \times \mathcal{P}(G) \times V(G)^*$ to induced subgraphs of G , defined for (X, R, X') if (X, R) is a **gv**-position and X' is the first component of a **gv**-successor of

¹A strategy need only be defined for all positions (X, R) that can be reached from (ϵ, G) in a LIFO-search consistent with the strategy. However, as this definition is somewhat circular, we assume strategies are total.

(X, R) . A LIFO-search $(X_0, R_0), (X_1, R_1), \dots$ is *consistent* with a fugitive \mathbf{gv} -strategy ρ if $R_{i+1} = \rho(X_i, R_i, X_{i+1})$ for all $i \geq 0$, and a fugitive strategy is winning if all consistent complete LIFO-searches are winning for the fugitive. In this section, a strategy will always refer to a searcher strategy.

3.2 Relating the digraph searching parameters

We observe that in all game variants, a strategy that is winning from (X, R) can be used to define a strategy that is winning from (X, R') for any $R' \subseteq R$: the searchers can play as if the fugitive is located in the larger space; and from the monotonicity of the successor relation, the assumption that the actual set of locations of the fugitive is a subset of the assumed set of locations remains invariant. One consequence is that a winning strategy on G defines a winning strategy on any subgraph of G , so the search numbers we have defined are monotone with respect to the subgraph relation.

Proposition 1. *Let G be a digraph and G' a subgraph of G . Then:*

- $SS^{vsc}(G') \leq SS^{vsc}(G)$, and
- $LIFO^{\mathbf{gv}}(G') \leq LIFO^{\mathbf{gv}}(G)$ for $\mathbf{gv} \in \{i, isc, v, vsc, mi, misc, mv, mvsc\}$.

Another consequence is that a winning strategy in the invisible fugitive variant defines a winning strategy when the fugitive is visible; and a winning strategy when the fugitive is not constrained to moving within strongly connected components defines a winning strategy when he is. This corresponds to our intuition of the fugitive being more (or less) restricted. Also, in all game variants, a monotone winning strategy is clearly a winning strategy, and because a searcher-stationary LIFO-search is monotone, a winning searcher-stationary strategy is a monotone winning strategy. These observations yield several inequalities between the search numbers defined above. For example $LIFO^{vsc}(G) \leq LIFO^{mi}(G)$ as any winning monotone i -strategy is also a winning vsc -strategy. The full set of these relationships is shown in a Hasse diagram in Figure 1, with the larger measures towards the top.

The main result of this section is that all these digraph parameters are equal to one more than cycle-rank.

Theorem 1. *For any digraph G :*

$$\begin{aligned} 1 + cr(G) &= LIFO^{mi}(G) = LIFO^i(G) = LIFO^{misc}(G) = LIFO^{isc}(G) \\ &= LIFO^{mv}(G) = LIFO^v(G) = LIFO^{mvsc}(G) = LIFO^{vsc}(G) \\ &= SS^{vsc}(G). \end{aligned}$$

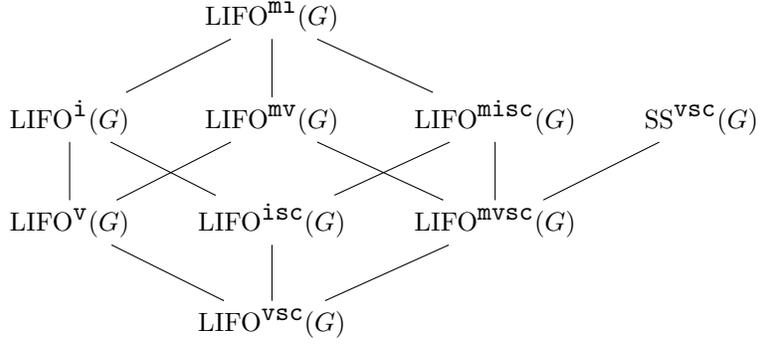


Figure 1: Trivial relations between digraph searching parameters

Proof. From the above observations, to prove Theorem 1 it is sufficient to prove the following three inequalities:

- (1) $\text{LIFO}^{\text{vsc}}(G) \geq \text{SS}^{\text{vsc}}(G)$,
- (2) $\text{SS}^{\text{vsc}}(G) \geq 1 + \text{cr}(G)$, and
- (3) $1 + \text{cr}(G) \geq \text{LIFO}^{\text{mi}}(G)$.

□

These are established with the following series of lemmas.

Lemma 1. *For any digraph G , $\text{LIFO}^{\text{vsc}}(G) \geq \text{SS}^{\text{vsc}}(G)$.*

Proof. We show that if a **vsc**-strategy is not searcher-stationary then it is not a winning strategy from (ϵ, G) . The result then follows as this implies every winning **vsc**-strategy is searcher-stationary. Let σ be a **vsc**-strategy, and suppose $(X_0, R_0), (X_1, R_1), \dots$ is a complete **vsc**-LIFO-search from $(X_0, R_0) = (\epsilon, G)$ consistent with σ which is not searcher-stationary. Let j be the least index such that $X_j \succeq X_{j+1}$ and $R_j \neq \emptyset$. As $X_0 = \epsilon$, there exists $i < j$ such that $X_i = X_{j+1}$. By the minimality of j , and the assumption that we only place or remove one searcher in each round, $i = j - 1$. As $X_{j-1} \preceq X_j$, $R_j \subseteq R_{j-1}$, and as $X_{j+1} \preceq X_j$, $R_j \subseteq R_{j+1}$. As $R_j \neq \emptyset$, it follows that R_{j-1} and R_{j+1} are the same strongly connected component of $G \setminus \{X_{j-1}\}$. Thus (X_{j-1}, R_{j-1}) is a **vsc**-successor of (X_j, R_j) . As $\sigma(X_j, R_j) = X_{j+1} = X_{j-1}$, it follows that $(X_0, R_0), (X_1, R_1), \dots, (X_{j-1}, R_{j-1}), (X_j, R_j), (X_{j-1}, R_{j-1}), (X_j, R_j), \dots$ is an infinite, and hence complete, **vsc**-LIFO-search (from (ϵ, G)) consistent with σ . As $R_i \neq \emptyset$ for all $i \geq 0$, the LIFO-search is not winning for the searchers. Thus σ is not a winning strategy. □

Lemma 2. For any digraph G , $SS^{vsc}(G) \geq 1 + cr(G)$.

Proof. We prove this by induction on $|V(G)|$.

If $|V(G)| = 1$, then $SS^{vsc}(G) = 1 = 1 + cr(G)$.

Now suppose $SS^{vsc}(G') \geq 1 + cr(G')$ for all digraphs G' with $|V(G')| < |V(G)|$. We first consider the case when G is not strongly connected. From Proposition 1, $SS^{vsc}(G) \geq \max_H SS^{vsc}(H)$ where the maximum is taken over all strongly connected components H of G . As G is not strongly connected, $|V(H)| < |V(G)|$ for all strongly connected components H of G . Therefore, by the induction hypothesis

$$\begin{aligned} SS^{vsc}(G) &\geq \max_H SS^{vsc}(H) \\ &\geq \max_H (1 + cr(H)) \\ &= 1 + cr(G). \end{aligned}$$

Now suppose G is strongly connected. Let σ be a winning searcher-stationary vsc -strategy which uses $SS^{vsc}(G)$ searchers. As (ϵ, G) is a legitimate vsc -position, if (X, R) is a vsc -successor of (ϵ, G) then $|X| = 1$. Thus $|\sigma(\epsilon, G)| = 1$. Let $\sigma(\epsilon, G) = v_0$. As σ is a searcher-stationary strategy which uses the minimal number of searchers, it follows that $SS^{vsc}(G \setminus \{v_0\}) = SS^{vsc}(G) - 1$. Thus, by the induction hypothesis,

$$\begin{aligned} SS^{vsc}(G) &= SS^{vsc}(G \setminus \{v_0\}) + 1 \\ &\geq (1 + cr(G \setminus \{v_0\})) + 1 \\ &\geq (1 + \min_{v \in V(G)} cr(G \setminus \{v\})) + 1 \\ &= 1 + cr(G). \end{aligned}$$

□

Lemma 3. For any digraph G , $1 + cr(G) \geq LIFO^{mi}(G)$.

Proof. We also prove this by induction on $|V(G)|$.

If $|V(G)| = 1$, then $1 + cr(G) = 1 = LIFO^{mi}(G)$.

Now suppose $1 + cr(G') \geq LIFO^{mi}(G')$ for all digraphs G' with $|V(G')| < |V(G)|$. First we consider the case when G is not strongly connected. As $|V(H)| < |V(G)|$ for each strongly connected component H , by the inductive hypothesis, there is a monotone i -strategy, σ_H , which captures a fugitive using at most $1 + cr(H)$ searchers. From the definition of cycle-rank, for each strongly connected component H of G , $cr(G) \geq cr(H)$, thus σ_H uses at most $1 + cr(G)$ searchers. We define a monotone i -strategy which captures a fugitive on G with at most $1 + cr(G)$ searchers as follows. Intuitively, we search the strongly

connected components of G in topological order using the monotone strategies σ_H . More precisely, let H_1, H_2, \dots, H_n be an ordering of the strongly connected components of G such that if there is an edge from H_i to H_j then $i < j$. We define σ as follows.

- $\sigma(\epsilon, G) = \sigma_{H_1}(\epsilon, H_1)$,
- For $1 \leq i$, if $\{X\} \subseteq H_i$ and $R = R' \cup \bigcup_{j=i+1}^n H_j$ where $\emptyset \neq R' \subseteq H_i$, $\sigma(X, R) = \sigma_{H_i}(X, R')$,
- For $1 \leq i < n$, if $\emptyset \neq \{X\} \subseteq H_i$ and $R = \bigcup_{j=i+1}^n H_j$ then $\sigma(X, R) = X'$ where X' is the maximal proper prefix of X .

From the definition of i -successors and the ordering of the strongly connected components if $(X_0, R_0), \dots, (X_n, R_n)$ is an i -LIFO-search on G where $\{X_n\} \subseteq H_i$ and $\bigcup_{j>i} H_j \subseteq R_{n-1} \subseteq \bigcup_{j \geq i} H_j$, then $\bigcup_{j>i} H_j \subseteq R_n \subseteq \bigcup_{j \geq i} H_j$. It follows (by induction on the length of a LIFO-search) that every LIFO-search from (ϵ, G) consistent with σ can be divided into a sequence of LIFO-searches $\lambda_1, \lambda_2, \dots, \lambda_n$, where λ_i can be viewed as a LIFO-search consistent with σ_{H_i} with $\bigcup_{j>i} H_j$ added to the second component of every position. Thus if each σ_{H_i} is monotone, winning and uses at most $1 + \text{cr}(G)$ searchers, then σ is also monotone, winning and uses at most $1 + \text{cr}(G)$ searchers.

Now suppose G is strongly connected. Let v_0 be the vertex which minimizes $f(v) = \text{cr}(G \setminus \{v\})$. Let $G' = G \setminus \{v_0\}$, so $\text{cr}(G) = 1 + \text{cr}(G')$. By the induction hypothesis, there exists a winning monotone i -strategy σ' which uses at most $1 + \text{cr}(G')$ searchers to capture a fugitive on G' . We define an i -strategy σ on G which uses at most $2 + \text{cr}(G') = 1 + \text{cr}(G)$ searchers as follows. Initially, place (and keep) a searcher on v_0 , then play the strategy σ' on $G \setminus \{v_0\}$. More precisely, $\sigma(\epsilon, G) = v_0$ and $\sigma(v_0 X, R) = v_0 \cdot \sigma'(X, R)$. Clearly any LIFO-search consistent with σ can be viewed as a LIFO-search consistent with σ' prepended with the position (ϵ, G) and where the first component of every position is prepended with v_0 . Thus if σ' is monotone, then σ is monotone, and if σ' is winning then σ is winning. Thus σ is a monotone winning i -strategy which uses at most $1 + \text{cr}(G)$ searchers. \square

3.3 Relation with other graph parameters

With a characterization of cycle-rank in terms of several graph searching games we can compare it with other digraph measures defined by similar games. In particular, the directed pathwidth of a digraph, $\text{dpw}(G)$, which can be characterized by an invisible-fugitive graph searching game [2], and the DAG-depth, $\text{dd}(G)$ which can be characterized by a visible-fugitive, searcher-stationary searching

game [12]. Whilst the relationships we present here are known [14, 12], using the game characterizations we obtain a more simple and more intuitive proof.

Corollary 1. *For any digraph G , $d_{pw}(G) \leq cr(G) \leq dd(G) - 1$.*

4 Obstructions for cycle-rank

In this section we consider the dual parameter arising from considering the graph searching games from the fugitive's perspective. We show that it can be characterized by two types of structural features, akin to the havens and brambles used to dually characterize treewidth [28]. To do so we first define the notion of the strong shelter of a digraph, a structural obstruction which we show to be dual to cycle-rank.

Definition 2. A *strong shelter* of a digraph G is a collection \mathcal{S} of non-empty strongly connected sets of vertices such that for any $S \in \mathcal{S}$

$$\bigcap \{S' : S' \in M_{\mathcal{S}}(S)\} = \emptyset,$$

where $M_{\mathcal{S}}(S)$ is the \subseteq -maximal elements of $\{S' \in \mathcal{S} : S' \subset S\}$. The *thickness* of a shelter \mathcal{S} is the minimal length of a maximal \subseteq -chain.

The second structural obstruction we consider is motivated by the definition of a haven in [16], a structural feature dual to directed treewidth.

Definition 3. A *LIFO-haven of order k* of a digraph G is a function h from $V(G)^{<k}$ to induced subgraphs of G such that:

- (H1) $h(X)$ is a non-empty strongly connected component of $G \setminus \{X\}$, and
- (H2) If $X \preceq Y$ and $|Y| < k$ then $h(Y) \subseteq h(X)$.

Whilst Adler [1] has shown that the havens of [16] do not give an exact min-max characterization of directed treewidth and Safari [27] has shown that directed versions of havens and brambles give rise to distinct parameters, we show that LIFO-havens and strong shelters both give a tight min-max characterization of cycle-rank.

Theorem 2 (Min-max theorem for cycle-rank). *Let G be a digraph and k a positive integer. The following are equivalent:*

- (i) G has cycle-rank less than k ,
- (ii) G has no LIFO-haven of order greater than k , and

(iii) G has no strong shelter of thickness greater than k .

Proof. (i) \Rightarrow (ii). Assume that it is not the case that G has no LIFO-haven of order greater than k , that is, G has a LIFO-haven h of order at least $k + 1$. We show that the fugitive has a winning strategy against k searchers, so by Theorem 1, $\text{cr}(G) \geq k$. Define a **vsc**-strategy ρ for the fugitive (against k searchers) by defining $\rho(X, R, X') = h(X')$ for all suitable triples (X, R, X') . From (H1), $(X', \rho(X, R, X'))$ is a valid **vsc**-position. Furthermore, (H2) implies that if (X, R) is a **vsc**-position such that $R = h(X)$, then $(X', \rho(X, R, X'))$ is a **vsc**-successor of (X, R) , so ρ is a **vsc**-strategy (defined for all LIFO-searches that use at most k searchers). Also, if $(X_0, R_0), (X_1, R_1) \dots$ is a complete LIFO-search consistent with ρ then $R_i = h(X_i)$ for all $i > 0$. As $h(X) \neq \emptyset$ when $|X| \leq k$, it follows that all consistent complete LIFO-searches that use at most k searchers are winning for the fugitive. Thus ρ is a winning strategy for the fugitive, so $\text{LIFO}^{\text{vsc}}(G) > k$. By Theorem 1, $\text{cr}(G) \geq k$.

(ii) \Rightarrow (iii). We show that a strong shelter \mathcal{S} of thickness at least k can be used to define a haven of order at least k . For each $X \in V(G)^{<k}$ we define $S_X \in \mathcal{S}$ inductively as follows. For $X = \epsilon$, let S_ϵ be any \subseteq -maximal element of \mathcal{S} . Note that $\{S \in \mathcal{S} : S \subset S_\epsilon\}$ is a strong shelter of thickness at least $k - 1$. Now suppose $X = X'v$, $S_{X'}$ is defined, $S_{X'} \cap \{X'\} = \emptyset$, and $\mathcal{S}_{X'} = \{S \in \mathcal{S} : S \subset S_{X'}\}$ is a strong shelter of thickness at least $k - 1 - |X'|$. From the definition of a strong shelter, there exists a \subseteq -maximal element of $\mathcal{S}_{X'}$ that does not contain v , as otherwise $v \in S$ for all $S \in M_{\mathcal{S}}(S_{X'})$. Let S_X be that element. As $S_{X'} \cap \{X'\} = \emptyset$ and $v \notin S_X$, it follows that $S_X \cap \{X\} = \emptyset$. Further, $\{S \in \mathcal{S} : S \subset S_X\}$ is a strong shelter of thickness at least $(k - 1 - |X'|) - 1 = k - 1 - |X|$, satisfying the assumptions necessary for the next stage of the induction. Now for all $X \in V(G)^{<k}$, S_X is a non-empty strongly connected set such that $S_X \cap \{X\} = \emptyset$. Thus there is a unique strongly connected component of $G \setminus \{X\}$ that contains S_X . Defining $h(X)$ to be that component we see that h satisfies (H1). For (H2), from the definition of S_X , if $X \preceq Y$ and $|Y| < k$, then $S_X \supseteq S_Y$, so $h(X) \supseteq h(Y)$. Therefore h is a haven of order at least k .

(iii) \Rightarrow (i). Again, we prove the contrapositive. Suppose $\text{cr}(G) \geq k$. Let G' be a strongly connected component of G which has cycle-rank $\geq k$. We prove by induction on k that G' , and hence G , has a strong shelter of thickness at least $k + 1$. Every digraph with $|V(G)| \geq 1$ has a strong shelter of thickness 1: take $\mathcal{S} = \{\{v\}\}$ for some $v \in V(G)$. Thus for $k = 0$, the result is trivial. Now suppose for $k' < k$ every digraph of cycle-rank $\geq k'$ contains a strong shelter of thickness at least $k' + 1$. For $v \in V(G')$, let $G'_v = G' \setminus \{v\}$. From the definition of cycle-rank, $\text{cr}(G'_v) \geq k - 1$ for all $v \in V(G')$. Thus, by the induction hypothesis, G'_v contains a strong shelter, \mathcal{S}_v , of thickness at least $(k - 1) + 1$. As $v \notin \mathcal{S}$ for

all $S \in \mathcal{S}_v$, it follows that $\mathcal{S} = \{G'\} \cup \bigcup_{v \in V(G')} \mathcal{S}_v$ is a strong shelter of G . As \mathcal{S}_v has thickness at least k for all $v \in V(G')$, \mathcal{S} has thickness at least $k + 1$. \square

Combining Theorems 1 and 2 we obtain the following.

Theorem 3. *Let G be a digraph, and k a positive integer. The following are equivalent:*

- (i) G has cycle-rank at most $k - 1$,
- (ii) On G , k searchers can capture a fugitive with a LIFO-search strategy,
- (iii) On G , k searchers can capture a visible fugitive restricted to moving in strongly connected sets with a searcher-stationary search strategy,
- (iv) G has no LIFO-haven of order greater than k , and
- (v) G has no strong shelter of thickness greater than k .

5 LIFO-search in simple graphs

In this section we consider the consequences of our results to simple graphs. In order to do so, we first give the definition of tree-depth and then restrict the notions of strong shelters on simple graphs.

Definition 4 (Tree-depth). The *tree-depth* of a graph G , $\text{td}(G)$, is defined as follows:

- If $|V(G)| = 1$ then $\text{td}(G) = 1$.
- If $|V(G)| > 1$ and G is connected then $\text{td}(G) = 1 + \min_{v \in V(G)} \text{td}(G \setminus \{v\})$.
- Otherwise $\text{td}(G) = \max_H \text{td}(H)$ where the maximum is taken over all connected components H of G .

Tree-depth is also known as the vertex ranking problem [5], the ordered colouring problem [17], or the the minimum-height of an elimination tree of a graph [6, 8, 23] and has received much attention, mostly because of the theory of graph classes of bounded expansion, developed by Nešetřil and Ossona de Mendez in [23, 21, 24, 25, 22].

Given a graph G we define the digraph G_d where $V(G_d) = V(G)$ and $E(G_d) = \{(x, y) \mid \{x, y\} \in E(G)\}$, that is, G_d is obtained from G after replacing every edge $\{x, y\} \in E(G)$ with the arcs (x, y) and (y, x) . From Definitions 1 and 4 we get that.

Observation 1. For every graph G , $td(G) = cr(G_d) + 1$.

We now give the definition of shelters when restricted to simple graphs.

Definition 5 (Shelter). A *shelter* of G is a collection \mathcal{S} of connected sets in G such that for every set $S \in \mathcal{S}$ its children have no vertex in common, in other words,

$$\bigcap \{S' \mid S' \in M_{\mathcal{S}}(S)\} = \emptyset,$$

where $M_{\mathcal{S}}(S)$ is the \subseteq -maximal elements of $\{S' \in \mathcal{S} : S' \subset S\}$. The *thickness* of a shelter \mathcal{S} is the minimal length of a maximal \subseteq -chain.

Observation 1 ensures that we may restate Theorem 3 for simple graphs in the following way.

Theorem 4. Let G be a non-empty graph and k be a positive integer. Then the following are equivalent.

- (i) G has tree-depth at most k
- (ii) there is a monotone LIFO-search strategy in G of cost at most k that captures an invisible and agile fugitive.
- (iii) there is a LIFO-search strategy in G of cost at most k that captures an invisible and agile fugitive.
- (iv) every shelter in G has thickness at most k .
- (v) there is a monotone LIFO-search strategy in G using k searchers against a visible and agile fugitive.

6 Conclusions and further work

To conclude, this multiple characterization of cycle-rank gives a new perspective on the measure which can be useful for further investigation. For example, whilst it is known that computing the cycle-rank is NP-complete [14], the characterization in terms of a graph searching game with a visible fugitive automatically implies that for any fixed k , deciding if a digraph has cycle-rank k is decidable in polynomial time. From a parameterized complexity perspective, techniques based on separators have shown measures such as directed treewidth are fixed-parameter tractable. Whether the visible, strongly connected game characterizations of cycle-rank can improve the known complexity from XP to FPT is part of ongoing research.

References

- [1] Isolde Adler. Directed tree-width examples. *Journal of Combinatorial Theory, Series B*, 97(5):718–725, 2007.
- [2] János Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.
- [3] Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. DAG-width and parity games. In *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science*, pages 524–536, 2006.
- [4] Daniel Bienstock and Paul D. Seymour. Monotonicity in graph searching. *Journal of Algorithms*, 12:239–245, 1991.
- [5] Hans L. Bodlaender, Jitender S. Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. Rankings of graphs. *SIAM J. Discrete Math.*, 11(1):168–181 (electronic), 1998.
- [6] P. de la Torre, R. Greenlaw, and A. A. Schäffer. Optimal edge ranking of trees in polynomial time. *Algorithmica*, 13(6):592–618, 1995.
- [7] Nick D. Dendris, Lefteris M. Kirousis, and Dimitrios M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoretical Computer Science*, 172(1-2):233–254, 1997.
- [8] J. S. Deogun, T. Kloks, D. Kratsch, and H. Müller. On vertex ranking for permutation and other graphs. In *STACS 94 (Caen, 1994)*, volume 775 of *Lecture Notes in Comput. Sci.*, pages 747–758. Springer, Berlin, 1994.
- [9] L. C. Eggan. Transition graphs and the star-height of regular events. *Michigan Mathematical Journal*, 10(4):385–397, 1963.
- [10] Stanley Eisenstat and Joseph Liu. The theory of elimination trees for sparse unsymmetric matrices. *SIAM Journal of Matrix Analysis & Applications*, 26(3):686–705, 2005.
- [11] Fedor V. Fomin, Pierre Fraigniaud, Stephan Kreutzer, and Dimitrios M. Thilikos. Theory and Applications of Graph Searching Problems (GRASTA 2011) (Dagstuhl Seminar 11071). *Dagstuhl Reports*, 1(2):30–46, 2011.
- [12] Robert Ganian, Petr Hliněný, Joachim Kneis, Alexander Langer, Jan Obdržálek, and Peter Rossmanith. On digraph width measures in parameterized algorithmics. In *4th International Workshop on Parameterized and Exact Computation (IWPEC 2009)*, pages 185–197, 2009.

- [13] Archontia C. Giannopoulou and Dimitrios M. Thilikos. A min-max theorem for lifo-search. *Electronic Notes in Discrete Mathematics*, 2011.
- [14] Hermann Gruber. Digraph Complexity Measures and Applications in Formal Language Theory. In *4th Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2008)*, pages 60–67, 2008.
- [15] Paul Hunter. Lifo-search on digraphs: A searching game for cycle-rank. In *FCT*, 2011.
- [16] Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory (Series B)*, 82(1):138–154, 2001.
- [17] Meir Katchalski, William McCuaig, and Suzanne Seager. Ordered colourings. *Discrete Math.*, 142(1-3):141–154, 1995.
- [18] Lefteris Kirousis and Christos Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(3):205–218, 1986.
- [19] Stephan Kreutzer and Sebastian Ordyniak. Digraph decompositions and monotonicity in digraph searching. In *34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008)*, pages 336–347, 2008.
- [20] Andrea S. LaPaugh. Recontamination does not help to search a graph. *Journal of the ACM*, 40(2):224–245, 1993.
- [21] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion II. Algorithmic aspects. *Eur. J. Comb.*, 29(3):777–791, 2008.
- [22] Jaroslav Nešetřil and Patrice Ossona de Mendez. Linear time low tree-width partitions and algorithmic consequences. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 391–400. ACM, New York, 2006.
- [23] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European J. Combin.*, 27(6):1022–1041, 2006.
- [24] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion. I. Decompositions. *European J. Combin.*, 29(3):760–776, 2008.

- [25] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion. II. Algorithmic aspects. *European J. Combin.*, 29(3):777–791, 2008.
- [26] Jan Obdržálek. DAG-width: Connectivity measure for directed graphs. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 814–821, 2006.
- [27] Mohammad Ali Safari. D-width: A more natural measure for directed tree width. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, pages 745–756, 2005.
- [28] Paul D. Seymour and Robin Thomas. Graph searching, and a min-max theorem for tree-width. *Journal of Combinatorial Theory (Series B)*, 58:22–33, 1993.