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

Archontia C. Giannopoulou [a], Paul Hunter [b], Dimitrios M. Thilikos [a,*]

[a] *Department of Mathematics, National and Kapodistrian University of Athens, Greece*
[b] *Department of Computer Science, University of Oxford, United Kingdom*

ARTICLE INFO

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 (directed) shelters in (di)graphs and prove a min–max theorem implying their equivalence to the cycle-rank/tree-depth parameter in (di)graphs. As (directed) 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.

© 2012 Elsevier B.V. All rights reserved.

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.

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, that is, whether their strategy provides to the fugitive access to already searched areas (non-monotone strategy)

or not (monotone strategy). Monotone search strategies lead to algorithmically useful decompositions, whereas non-monotone strategies are more robust under graph operations and hence reflect structural properties. Therefore, 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 connected 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—one of the possible generalizations 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 in [9], where it was shown to be a critical parameter for determining the star-height of regular languages. The success of tree-depth [10,14,12] rekindled interest in it as an important digraph parameter, especially from an algorithmic perspective.

It is well known that tree-depth can 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 another 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 directed shelters. The second one is motivated by the havens of [16]. Both the directed shelters and LIFO-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 the aforementioned simplified 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 directed 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.*