ICGT 2014, Grenoble, France, June 30–July 4, 2014

Contraction Obstructions for Connected Graph Searching*

Micah J. Best† Arvind Gupta‡† Dimitrios M. Thilikos§¶ Dimitris Zoros§

Abstract

We consider the connected variant of the classic mixed search game where, in each search step, cleaned edges form a connected subgraph. We consider graph classes with bounded connected monotone mixed search number and we deal with the the question weather the obstruction set, with respect of the contraction partial ordering, for those classes is finite. In general, there is no guarantee that those sets are finite, as, in general, graphs are not well quasi ordered under the contraction partial ordering relation. In this paper we provide the obstruction set for $k = 2$. This set is finite, it consists of 174 graphs and completely characterizes the graphs with connected monotone mixed search number at most 2. Our proof reveals that the "sense of direction" of an optimal search searching is important for connected search which is in contrast to the unconnected original case.

1 Introduction

A *mixed searching game* is defined in terms of a graph representing a system of tunnels where an agile and omniscient fugitive with unbounded speed is hidden (alternatively, we can formulate the same problem considering that the tunnels are contaminated by some poisonous gas). The fugitive is occupying the edges of the graph and the searchers can be placed on its vertices. In the beginning of the game, the fugitive chooses some edge and there are no searchers at all on the graph. The objective of the searchers is to deploy a search strategy on the graph that can guarantee the capture of the fugitive. The fugitive is *captured* if at some point he resides on an edge e and one of the following capturing cases occurs.

A: *both endpoints of e are occupied by a searcher,*

B: *a searcher slides along e*, i.e., a searcher is moved from one endpoint of the edge to the other endpoint.

A *search strategy* on a graph G is a finite sequence $\mathcal{S}$ containing moves of the following types.

*The second author of this paper where funded by the Natural Science and Engineering Research Council of Canada, Mitacs Inc, and the University of British Columbia. The third and the fourth author of this paper where co-financed by the E.U. (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "Thales. Investing in knowledge society through the European Social Fund". Emails: `mjbest@cs.ubc.ca`, `arvind@mitacs.ca`, `sedthilk@thilikos.info`, `dzoros@math.uoa.gr` .

†Department of Computer Science, University of British Columbia, B.C., Canada.

‡Mathematics of Information Technology & Complex Systems (MITACS).

§Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece.

¶AlGCo project team, CNRS, LIRMM, France.

p(v): placing a new searcher on a vertex v,
r(v): deleting a searcher from a vertex v,
s(v, u): sliding a searcher on v along the edge $\{v, u\}$ and placing it on u.

We stress that the fugitive is *agile* and *omniscient*, i.e. he moves at any time in the most favorable, for him, position and is *invisible*, i.e. the searchers strategy is given "in advance" and does not depend on the moves of the fugitive during it.

Given a search $\mathcal{S}$, we denote by $E(\mathcal{S}, i)$ the set of edges that are clean after applying the first i steps of $\mathcal{S}$, where by "clean" we mean that the search strategy can guarantee that none of its edges will be occupied by the fugitive after the i-th step. More formally, we set $E(\mathcal{S}, 0) = \emptyset$ and in step $i > 0$ we define $E(\mathcal{S}, i)$ as follows: first consider the set Q_i containing all the edges in $E(\mathcal{S}, i-1)$ plus the edges of $E^{(i)}$ the set of edges that are cleaned after the i-th move because of the application of cases **A** or **B**. Notice that $E^{(i)}$ may be empty. In particular, it may be non-empty in case the i-move is a placement move, will always be empty in case the i-th move is a removal move and will surely be non-empty in case the i-th move is a sliding move. In the third case, the edge along which the sliding occurs is called *the sliding edge* of $E^{(i)}$. Then, the set $E(\mathcal{S}, i)$ is define as the set of all edges in Q_i minus those for which there is a path starting from them and finishing in an edge not in Q_i. This expresses the fact that the agile and omniscient fugitive could use any of these paths in order to occupy again some of the edges in Q_i. In case $E(\mathcal{S}, i) \subset Q_i$, we say that the i-th move is a *recontamination move*. Notice that in such a case we have that $E(\mathcal{S}, i-1) \not\subseteq E(\mathcal{S}, i)$.

The object of a mixed search is to clear all edges using a search. We call search $\mathcal{S}$ *complete* if at some step all edges of G are clean, i.e. $E(\mathcal{S}, i) = E(G)$ for some i.

Connected monotone mixed search number The mixed search number of a search is the maximum number of searchers on the graph during any move. A search without recontamination moves is called *monotone*. Mixed search number has been introduced in [2]. A search is *connected* if $E(\mathcal{S}, i)$ induces a connected subgraph of G for every step i. The mixed search number, $\mathbf{s}(G)$, of a graph G is the minimum mixed search number over all the possible complete searches on it (if G is an edgeless graph, then this number is 0). The monotone (resp. connected monotone) mixed search number, $\mathbf{ms}(G)$ (resp. $\mathbf{cms}(G)$), of G is the minimum mixed search number over all the possible complete monotone (connected monotone) searches of it (connected variants are defined only under the assumption that G is a connected graph). The concept of connectivity in graph searching was introduced for the first time in [1] and was motivated by application of graph searching where the "clean" territories should be maintained connected so to guarantee the safe communication between the searchers during the search.

Figure 1: The set $\mathcal{O}_1$ (left) and the set of rooted graphs $\mathcal{A}$ (right).

Obstructions Given a graph invariant $\mathbf{p}$, a partial ordering relation on graphs $\leq$, and an integer k we denote by $\mathbf{obs}_{\leq}(\mathcal{G}[\mathbf{p}, k])$ the set of all $\leq$-minimal graphs G where $\mathbf{p}(G) > k$ and we call it *the k-th $\leq$-obstruction set for* $\mathbf{p}$. We also say that $\mathbf{p}$ *is closed under* $\leq$ if for every two graphs H and G, $H \leq G$ implies that $\mathbf{p}(H) \leq \mathbf{p}(G)$. Clearly, if $\mathbf{p}$ is closed under $\leq$, then the k-th $\leq$-obstruction set for $\mathbf{p}$ provides a complete characterization for the class $\mathcal{G}_k = \{G \mid \mathbf{p}(G) \leq k\}$: a graph belongs in $\mathcal{G}_k$ iff none of the graphs in the k-th $\leq$-obstruction set for $\mathbf{p}$ is contained in G with respect to the relation $\leq$.

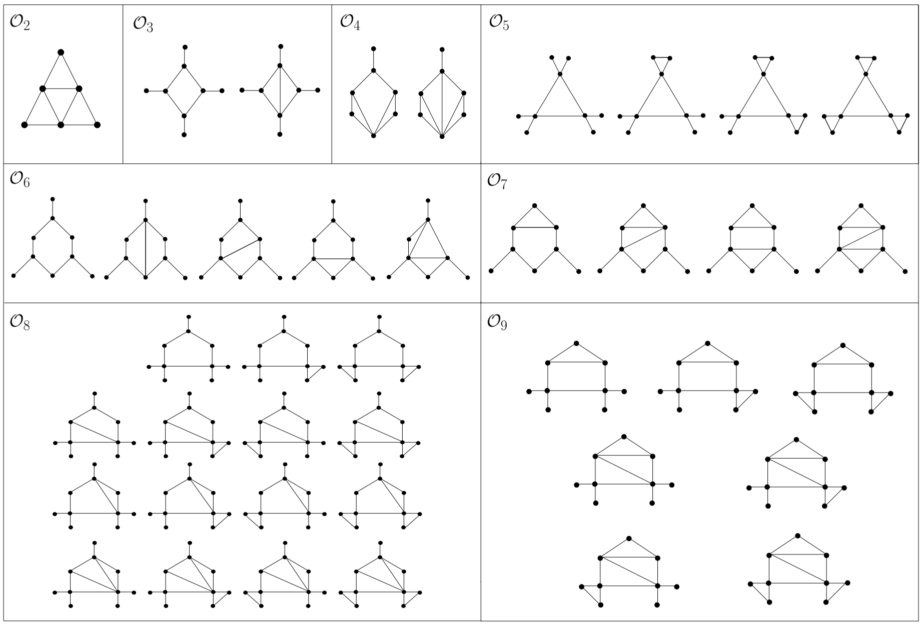

Figure 2: Some of the graphs in $\mathcal{D}_1$.

Our result In this paper we are interested in obstruction characterizations for the graphs with bounded connected monotone mixed search number. While It is known that $\mathbf{ms}$ is closed under taking of minors, this is not he case for $\mathbf{cms}$ where the connectivity requirement applies. From Robertson and Seymour theorem [3], then k-th $\leq$-obstruction set for $\mathbf{ms}$ is always finite. Moreover this set has been found for $k = 1$ (2 graphs) and $k = 2$ (36 graphs) in [4]. However, much less is known on obstruction characterizations of the connected monotone mixed search number. As we prove in this paper, $\mathbf{cms}$ is closed under contractions. Unfortunately, graphs are not well quasi ordered with respect to the contraction relation, therefore there is no guarantee that the k-th contraction obstruction set for $\mathbf{cms}$ is finite for all k. The finiteness of this set is straightforward if $k = 1$ as $\mathbf{obs}_{\preceq}(\mathcal{G}[\mathbf{cms}, 1]) = \{K_3, K_{1,3}\}$. In this paper we completely resolve the case where $k = 2$. We prove that $\mathbf{obs}_{\preceq}(\mathcal{G}[\mathbf{cms}, 2])$ is finite by providing all 174 graphs that it contains. The proof of our results is based on a series of lemmata that confine the structure of the graphs with connected monotone mixed search number at most 2. We should stress, that, in contrary to the case of $\mathbf{ms}$ the direction of searching is crucial for $\mathbf{cms}$. This makes the detection of the corresponding obstruction sets more elaborated as special obstructions are required in order to force a certain sense of direction in the search strategy. For this reason, our proof makes use of a more general variant of the mixed search strategy that forces the searchers to start and finish to specific sets of vertices. Obstructions for this more general type of searching are combined in order to form the required obstructions for **cms**.

2 Description of the obstruction set

Let $\mathcal{D}^1 = \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_{12}$ where $\mathcal{O}_1$ is depicted in the left part of Figure 1, $\mathcal{O}_2, \ldots, \mathcal{O}_9$ are depicted in Figure 2 and $\mathcal{O}_{10}$ and $\mathcal{O}_{11}$ and $\mathcal{O}_{12}$ are constructed as follows.

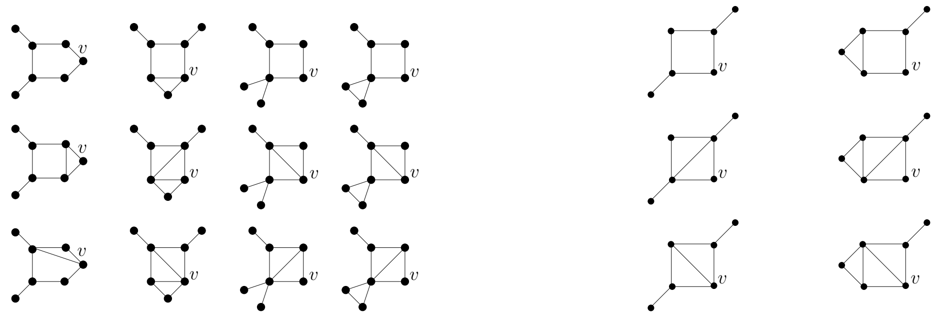

Figure 3: The sets of rooted graphs $\mathcal{B}$ (on the left) and $\mathcal{C}$ (on the right).

$\mathcal{O}_{10}$: contains every graph that can be constructed by taking three disjoint copies of some graphs in the right part of Figure 1 and then identify the vertices denoted by v in each of them to a single vertex. There are, in total, 35 graphs generated in this way.

$\mathcal{O}_{11}$: contains every graph that can be constructed by taking two disjoint copies of some graphs in the left part of Figure 3 and then identify the vertices denoted by v in each of them to a single vertex. There are, in total, 78 graphs generated in this way.

$\mathcal{O}_{12}$: contains every graph that can be constructed by taking two disjoint copies of some graphs in the right part of Figure 3. and then identify the vertices denoted by v in each of them to a single vertex. There are, in total, 21 graphs generated in this way.

Observe that, $\mathcal{D}^1$ contains 174 graphs. The proof of its correctness is lengthy and is omitted in this extended abstract.

References

[1] Lali Barrière, Paola Flocchini, Fedor V. Fomin, Pierre Fraigniaud, Nicolas Nisse, Nicola Santoro, and Dimitrios M. Thilikos. Connected graph searching. *Inf. Comput.*, 219:1–16, 2012.

[2] D. Bienstock and Paul D. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.

[3] Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004.

[4] Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Minimal forbidden minors for the family of graphs with proper-path-width at most two. *IEICE Trans. Fundamentals*, E78-A:1828–1839, 1995.