Synchronous black hole search in directed graphs☆

Adrian Kosowski [a,b,*], Alfredo Navarra [c], Cristina M. Pinotti [c]

[a] INRIA Bordeaux Sud-Ouest, LaBRI, 351 cours de la Libération, 33405 Talence, France
[b] Department of Algorithms and System Modeling, Gdańsk University of Technology, Narutowicza 11/12, 80952 Gdańsk, Poland
[c] Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Perugia, Italy

ARTICLE INFO

ABSTRACT

The paper considers a team of robots which has to explore a graph G, where some nodes can be harmful. Robots are initially located at the so-called *home base* node. The dangerous nodes are the so-called *black hole* nodes, and once a robot enters in one of them, it is destroyed. The goal is to find a strategy in order to explore G in such a way that minimum number of robots is wasted. The exploration ends if there is at least one surviving robot which knows all the edges leading to the black holes. As many variations of the problem have been considered so far, the solution and its measure heavily depend on the initial knowledge and the capabilities of the robots. In this paper, we assume that G is a directed graph, the robots are associated with unique identifiers, they know the number of nodes n of G (or at least an upper bound on n), and they know the number of edges Δ leading to the black holes. Each node is associated with a whiteboard where robots can read and write information in a mutual exclusive way.

A recently posed question [J. Czyzowicz, S. Dobrev, R. Kralovic, S. Miklik, D. Pardubska, Black hole search in directed graphs, in: Proc. of 16th International Colloquium on Structural Information and Communication Complexity, SIROCCO, LNCS, vol. 5869, 2009, pp. 182–194.] is whether some number of robots, expressed as a function of parameter Δ only, is sufficient to detect black holes in directed graphs of arbitrarily large order n. We give a positive answer to this question for the synchronous case, i.e., when the robots share a common clock, showing that $O(\Delta \cdot 2^\Delta)$ robots are sufficient to solve the problem. This bound is nearly tight, since it is known that at least 2^Δ robots are required for some instances. Quite surprisingly, we also show that unlike in the case of undirected graphs, for the directed version of the problem, synchronization can sometimes make a difference: for $\Delta = 2$, in the synchronous case 4 robots are always sufficient, whereas in the asynchronous case at least 5 robots are sometimes required.

1. Introduction

The subject of exploring an unknown graph by means of mobile entities has been widely considered in recent years. The increasing interest to the problem comes from the variety of applications that it meets. In robotics, it might be very useful to let a robot or a team of robots explore dangerous or impervious zones. In networking, software agents might automatically discover nodes of a network and perform updates and/or refuse their connections. In this paper, we are interested in the

exploration of a graph with faulty nodes, i.e., nodes that destroy any entering entity. Such nodes are called *black holes*, and the exploration of a graph in such kind of networks is usually referred as *black hole search*. In what follows, we refer to the mobile entities as robots. According to the assumed initial settings of the network, and the knowledge and the capabilities of the robots, many results have been provided. Pure exploration strategies, without dealing with black holes, have been widely addressed, see for instance [9,12] and references therein. In this case, the requirement is usually to perform the exploration as fast as possible. When black holes are considered, along with the time (or equivalently the number of edge traversals) required for a full exploration, the main goal resides in minimizing the number of robots that may fall into some black hole. We will say that a team of robots *solves the black hole search problem* if, at the end of the exploration, all the edges which do not lead into a black hole are marked as safe edges, and at least one robot from the team survives.

1.1. Related work

Undirected graphs. The literature dealing with black hole search problems mainly refer to undirected graphs. In this context, a further distinction is made between *synchronous* and *asynchronous* systems.

Concerning asynchronous systems, the most general results for undirected graphs can be found in [6] where the black hole search problem was considered in the presence of a single black hole without any limitation on the network topology. The authors showed that $\Delta + 1$ robots are sufficient to complete the exploration and require $\Theta(n^2)$ steps, with Δ being the degree of the black hole, and n the size of the network.

In [7], the attention was devoted to rings. The authors showed that 2 robots are enough in order to detect one black hole and they provided an algorithm which requires $O(n \log n)$ moves, where n is the dimension of the ring. This approach was generalized to arbitrary graphs in [6], where it is shown that with complete topological knowledge of the input graph, only two robots suffice and require $\Theta(n \log n)$ steps to locate the black hole.

The variant in which the robot entering the node cannot learn the link by which it has arrived was studied in [10]. For this scenario, a bound of $\frac{\Delta^2 + \Delta}{2} + 1$ on the number of required robots was established and shown to be tight for some instances of the problem.

Other restrictions have also been considered, like the knowledge of a map in [8] where the authors provided a strategy for locating one black hole by means of 2 robots in $O(n + d \log d)$ moves, where n is the size of the network and d its diameter. In [5], one black hole is optimally located by 2 robots in $O(n)$ moves in the case when the input graph has size n and its topology is among hypercubes, cube-connected cycles, star graphs, wrapped butterflies, chordal rings, multidimensional meshes or tori of restricted diameter. Time optimal algorithms for black hole search on ring networks have been recently provided in [1].

Concerning synchronous systems, in [3], the exploration subject was restricted to black holes in tree topologies. The authors showed that 2 robots are enough to perform the exploration when only one black hole exists. They provided a $\frac{5}{3}$-approximation algorithm with respect to the required number of edge traversals. In [11], the authors assume that the map is known and then show that the problem of locating one black hole by means of 2 synchronous robots is not polynomial time approximable within any constant factor less than $\frac{389}{388}$, and provide a 6-approximation algorithm. A slightly different problem was considered in [2] where the network admits many black holes and if a robot gets destroyed into one, then such a black hole disappears but not the underlying node.

Directed graphs. The additional property of having directed edges can only increase the difficulty of the problem as a robot which has traversed an edge, cannot in general come back directly by reversing its move. The first results concerning directed graphs have been recently published in [4]. The authors considered a setting where robots are associated with unique identifiers (IDs), they know the number of nodes n of the input directed graph $G = (V, A)$, and they know the number Δ of edges belonging to G leading to the black holes. If $BH \subset V$ is the set of black holes in G, the induced subgraph $G[V \setminus BH]$ is assumed to be strongly connected. Each node $v \in V$ is associated with a so-called *whiteboard* which is an available bounded memory (with size polynomial in n) accessible in a mutually exclusive way by the robots located at node v. The obtained results show a general case lower bound on the number of robots required in order to accomplish the black hole search.

Theorem 1.1 (*[4]*)**.** *In both the asynchronous and synchronous models, 2^Δ robots are sometimes required to solve the directed black hole search problem, for all $\Delta \geq 1$.*

Such a worst-case example is obtained, e.g., by considering a directed cycle on $n - 1$ vertices (including the home base), and connecting Δ of these nodes by outgoing links to a single black hole vertex. Using a combination-lock argument, by induction it was shown [4] that in this graph any team of less than 2^Δ robots may sometimes perish in the black hole, without any of the robots traversing the last link of the cycle which leads back to the home base.

In [4] it is also shown that $2\Delta + 1$ robots are enough for solving the problem on planar graphs with the planar embedding known by the robots, and 2Δ robots are needed in the worst case.

1.2. Our results

In this paper, we extend the results on directed graphs by also separately considering the synchronous and the asynchronous cases, i.e. when the robots share or not a common clock. Under the same settings of [4], for the synchronous

case we provide a general strategy which requires $O(\Delta \cdot 2^{\Delta})$ robots. This answers the main question posed in [4], i.e. whether some number of robots, expressed as a function of parameter Δ only, is sufficient to detect black holes in directed graphs of arbitrarily large order n. We also provide a strategy for $\Delta = 1$ and $\Delta = 2$ which requires 2 and 4 robots, respectively. For the asynchronous case, we show that 2 robots are still sufficient when $\Delta = 1$, but for $\Delta = 2$ at least 5 robots are sometimes required.

The paper is organized as follows. In Section 2 we recall the most important assumptions regarding the directed black hole search problem, and introduce some further notation. In Section 3 we state the main positive results of the paper for synchronous robots, proving that $O(\Delta \cdot 2^{\Delta})$ robots suffice to explore any graph, and 4 robots suffice when $\Delta = 2$. The results for asynchronous robots for $\Delta = 1$ and $\Delta = 2$ are given in Section 4. Section 5 contains some concluding remarks.

2. The model

We assume the same scenario as that introduced in [4]. The explored digraph $G = (V, A)$ contains a distinguished node called the *home base* (hb) from which all the robots start the exploration, and a distinguished set of black holes $BH \subseteq V \setminus \{hb\}$. We remark that from the perspective of this work, setting the number of black holes equal to $|BH| = 1$ does not affect any of the stated results. It is assumed that the induced subgraph $G[V \setminus BH]$ is strongly connected. The set of arcs leading into some black hole, $BA = A \cap ((V \setminus BH) \times BH)$, is known as the set of *black hole arcs*. All robots know two parameters of the graph: its order $n = |V|$, and the total number of black hole arcs, $\Delta = |BA|$. We remark that this assumption of the model comes from [4]; the algorithm which we propose in Section 3 only makes use of these parameters to determine the number of released robots and the waiting time between two successive robots are released, and hence knowledge of an upper bound on n and Δ is sufficient. A robot located at a node v has access to its own built-in memory, the whiteboard associated with the node, a global timer counting the steps of the algorithm, and a local labeling of the arcs leaving node v (called an assignment of port numbers), which can be used to select the next arc in the robot's traversal. Note that nodes do not have unique identifiers, although such identifiers can potentially be created by the robots exploring the graph.

For a node v of a directed (multi)graph H, the number of arcs entering v is denoted by $\text{indeg}_H(v)$, and the number of arcs exiting v is denoted by $\text{outdeg}_H(v)$.

3. Algorithm for the synchronous model

In this section we propose a strategy for solving the directed black hole search problem using $O(\Delta \cdot 2^{\Delta})$ synchronous robots. The robots are released from the home base hb one-by-one, at regular intervals of n^4 time steps. The strategy is formulated in such a way as to guarantee that within n^4 steps, i.e., before the next robot is released, the currently active robot is destroyed in some black hole.

Throughout the algorithm, each node v maintains a label $f(v)$ stored in its whiteboard. Let the subgraph $H_{f(v)}$ of G consist of all nodes $u \in V \setminus BH$ having $f(u) = f(v)$, and all arcs $(u_1, u_2) \in A$ such that $f(u_1) = f(u_2) = f(v)$ and (u_1, u_2) has been visited by at least one robot. The procedure is defined in such a way that initially $f(v)$ is unique for all visited nodes (the definition of $f(v)$ for unvisited nodes is irrelevant), whereas throughout the computation each of the graphs $H_{f(v)}$ remains strongly connected. Moreover, for each node v, the structure of the graph $H_{f(v)}$ and its embedding in the ports of G is encoded on the whiteboard of v.

The procedure is divided into Δ phases. The p-th phase, $1 \leq p \leq \Delta$ consists of $(\Delta + 2) \cdot S_p$ robot releases, where $S_p = \lceil 2^{\Delta} \cdot (\sqrt{3}/2)^{p-1} \rceil$. The phase is divided into the following three parts, involving robots with essentially different behavior:

- First part: S_p *searching robots* are released.
- Second part: in each of Δ iterations, S_p *cycle detection robots* are released, following along a slight modification of the paths used by the searching robots from the first part of the phase.
- Third part: S_p *cycle contraction robots* are released.

After the Δ phases have been completed, all nodes (apart from black holes) belong to the same strongly connected component $H_{f(hb)}$, which is encoded on the whiteboards of all these nodes, thus describing all the safe links of the graph. The procedures followed by the searching, cycle detection, and cycle contraction robots are discussed in detail below.

Procedure for the first part of the phase. The procedure for each searching robot r is defined as follows. Directly before and directly after traversing an arc, the robot writes this information on the whiteboards of the starting node and of the end node of the arc, respectively, together with its unique ID and the number of the move in the robot's move sequence. The robot proceeds to make its next move as follows:

1. Let the robot be located at an arbitrary node of some graph H_i. Then, the robot explores all nodes of H_i, identifying all arcs of G which leave or enter these nodes and do not belong to H_i. If there exists an outgoing arc (u_1, v_1) which was used by some robot r' at time t_1, and an incoming arc (v_2, u_2) which was used by the same robot r' at some later time t_2, where $u_1, u_2 \in V(H_i)$ and $v_1, v_2 \notin V(H_i)$, then robot r proceeds to extend component H_i by including in it all the nodes on the route $(v_1, \ldots, v_2)$ used by the robot r' between times t_1 and t_2. (Robot r' was either a previously released robot,
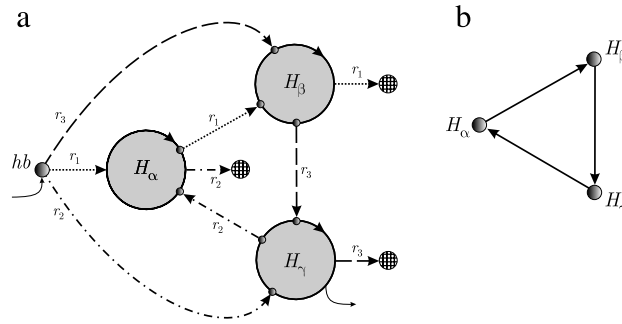
Fig. 1. (a) A schematic representation of the paths followed by three consecutive searching robots r_1, r_2, r_3. The strongly connected components H_α, H_β, H_γ associated with respective black hole arcs are assumed to remain unchanged during the phase. (b) The corresponding graph of the precedence relation ($\prec$) on $H_\alpha, H_\beta, H_\gamma$.

or $r' = r$.) More precisely, robot r visits all the nodes of all the graphs $H_{f(v)}$, with $v \in (v_1, \ldots, v_2)$, setting their labels to $f(v) := i$, and updating the stored graphs $H_{f(v)}$ accordingly.

2. When the extension from Step 1 can no longer be applied, the robot performs the next move as follows.
   - If there exists an arc leaving H_i which has never been visited by any robot, then the robot proceeds along this arc.
   - Otherwise, the robot proceeds along the arc leaving H_i which has not been used by any robot for the longest time.

Note that the actions performed in Step 1 will never lead the robot into a black hole, and do not lead it out of the current explored strongly connected component, whereas either of the actions performed in Step 2 can potentially destroy the robot or lead it into a node which does not belong to component H_i.

The number of steps performed by a searching robot can be upper-bounded by n^4. Indeed, the graph has less than n^2 arcs, and the discovery of each new arc by an agent may require a complete exploration of the current strongly connected component H_i, which can be performed in at most $n^2/2$ steps.

Procedure for the second and third parts of the phase. To avoid confusion, we will write $f^{(2)}$ and $H_i^{(2)}$ to denote the value of labels f and graphs H_i throughout the second part of the phase (the cycle detection robots do not update the labels), and likewise $f^{(3)}$ and $H_i^{(3)}$ to denote these labels and graphs at the end of the whole phase (after releasing all the cycle contraction robots).

Looking at the labels $f^{(2)}$ and graphs $H^{(2)}$, which are defined at the time of the destruction of the last searching robot, observe that each searching robot r can be seen as performing a traversal of some sequence of graphs $(H_{f_1}^{(2)}, \ldots, H_{f_k}^{(2)})$, where a graph $H_{f_i}^{(2)}$ appears in the sequence if at least one of its nodes has been visited by robot r. This implies a natural precedence relation ($\prec$) between graphs $H_{f_i}^{(2)}$ and $H_{f_j}^{(2)}$, that is, we will write $H_{f_i}^{(2)} \prec H_{f_j}^{(2)}$ if for some searching robot r, graph $H_{f_i}^{(2)}$ appears before graph $H_{f_j}^{(2)}$ in the robot's sequence of visits; see Fig. 1 for an illustration.

Relation $\prec$ does not have to be acyclic. The goal of the cycle detection robots is to detect cycles of length at most Δ in the graph of relation $\prec$. To do this, we release the robots in such a way that after the l-th of the Δ iterations in this phase, each node of graph $H_{f_j}^{(2)}$ stores a list of all f_i such that $H_{f_i}^{(2)} \prec^l H_{f_j}^{(2)}$, where ($\prec^l$) is the l-th power of relation ($\prec$). With each graph $H_{f_i}^{(2)}$, we associate a precedence set P_{f_i} whose copies are stored in the whiteboards of all of the nodes of $H_{f_i}^{(2)}$, and initially $P_{f_i} = \emptyset$. In the l-th iteration, for each searching robot r, exactly one cycle detection robot r' will retrace the route of r. Robot r' is defined so as to visit all nodes of all graphs in the same order $(H_{f_1}^{(2)}, \ldots, H_{f_k}^{(2)})$ as robot r, with transitions between adjacent graphs $H_{f_j}^{(2)}$ performed using the same arcs as those visited by robot r (this can be achieved since robot r may be assumed to store the information about its next move on the whiteboards of the nodes it is passing through). Additionally, for each $H_{f_j}^{(2)}$, robot r' now appends to the contents of set P_{f_j} the contents of all sets P_{f_i}, for all $i < j$, and also adds to P_{f_j} the element f_i, for all $i < j$; all the copies of set P_{f_j} stored in the nodes of $H_{f_j}^{(2)}$ are updated accordingly. Clearly, we have $H_{f_i}^{(2)} \prec^l H_{f_j}^{(2)}$ if and only if $f_i \in P_{f_j}$. For easier manipulation of data in the next part of the phase, the precedence sets should store additional information about the ID of the searching robots r inducing the respective precedence relations.

Finally, the goal of the cycle contraction robots is to contract into a single connected component all graphs $H_{f_i}^{(2)}$ forming cycles of length at most Δ in the precedence graph induced by relation ($\prec$). The cycle contraction once again retraces the route of corresponding searching robots from r. Upon detecting a cycle of length at most Δ in the precedence relation, i.e., $f_i \in P_{f_i}$ for some graph $H_{f_i}^{(2)}$, they follow the arcs of the directed cycle of G which induces this relation (the identifiers of these arcs should be stored in set P_{f_i} as auxiliary information). For all the nodes v of all the graphs $H_{f_j}^{(2)}$ encountered in this cycle, the robots then update their labels, setting the new value $f^{(3)}(v) = f_i$ and constructing the new contracted component $H_{f_i}^{(3)}$ accordingly. These labels and components are then used by the searching robots in the next phase of the algorithm.

We remark that in all three phases of the algorithm, the number of steps performed by a robot before it is destroyed in a black hole or it returns to the home base, can be upper-bounded by n^4 steps, hence the release times of successive robots can be scheduled in intervals of n^4 steps.

Theorem 3.1. *The proposed strategy solves the directed black hole search problem using $O(\Delta \cdot 2^\Delta)$ synchronous robots.*

Proof. The number of robots used by the algorithm can indeed be bounded as follows: $\sum_{p=1}^{\Delta}(\Delta+2)S_p = (\Delta+2)\sum_{p=1}^{\Delta}\lceil 2^\Delta \cdot (\sqrt{3}/2)^{p-1}\rceil \in O(\Delta \cdot 2^\Delta)$. We now show that our strategy solves all instances of the black hole search problem.

Let $\{v_1, \ldots, v_D\}$, where $D \leq \Delta$, be the set of vertices from which it is possible to exit by a black hole arc. At any given time step, consider the set of labels $\{f(v_1), \ldots, f(v_D)\} = \{f_1, f_2, \ldots, f_d\}$, with $1 \leq d \leq D$, and let δ_i denote the number of black hole arcs exiting from nodes of H_{f_i}, with $\sum_{1 \leq i \leq d} \delta_i = \Delta$. As in the case of the other parameters, we will use the notation $d^{(2)}, d^{(3)}, f_i^{(2)}, f_i^{(3)}$, and $\delta_i^{(2)}, \delta_i^{(3)}$ to denote the respective values during the second part and at the end of the third part of the phase.

First, we will prove by induction that at the end of the p-th phase, $1 \leq p < \Delta$, we have $d^{(3)} \leq \Delta - p$. The claim is clearly true for $p = 0$; we will show that in each subsequent phase, either the black hole search problem has already been successfully solved (i.e., $d^{(2)} = 1$ and $H_{f_1}^{(2)} = G[V \setminus BH]$), or the value of $d^{(3)}$ decreases by at least 1 (i.e., some two components H_{f_i} and H_{f_j} are contracted into each other during the phase). Consider for a moment the situation during the second part of the phase, and define the directed multigraph $(\mathcal{H}, R)$, where the set of nodes is $\mathcal{H} = \{s, H_{f_1}^{(2)}, H_{f_2}^{(2)}, \ldots, H_{f_d}^{(2)}\}$, with s being a special source node, while arcs are defined as follows. For each searching robot r released in the current phase, we add one arc from node $H_{f_i}^{(2)}$ to $H_{f_j}^{(2)}$ if and only if robot r visited at least one node of $H_{f_i}^{(2)}$ before visiting a node of $H_{f_j}^{(2)}$, and did not visit any node of any other graph from $\mathcal{H} \setminus \{H_{f_i}^{(2)}, H_{f_j}^{(2)}\}$ in between these two visits. For each robot r, we also add an arc from the special source node s to the first of the graphs $H_{f_i}^{(2)}$ visited by the robots, containing the homebase hb. Note that $\text{indeg}_{(\mathcal{H},R)}(s) = 0$, and $\text{outdeg}_{(\mathcal{H},R)}(s) = S_p$.

We consider the following cases.

(a) Multigraph $(\mathcal{H}, R)$ contains a directed cycle. Then, since the relation R restricted to nodes from $\mathcal{H} \setminus \{s\}$ is a sub-relation of precedence relation ($\prec$), and $|F| \leq \Delta$, the graph of relation ($\prec$) also contains a cycle of length at most Δ on the connected components from $\mathcal{H}$. Since all such components are contracted into one in the cycle contraction phase, we immediately obtain that the value of $d^{(3)}$ is smaller than that of $d^{(2)}$ for the current phase, hence $d^{(3)}$ decreases with respect to the previous phase.

(b) Multigraph $(\mathcal{H}, R)$ is acyclic, and no contractions are performed ($d^{(2)} = d^{(3)}$); we can then extend R to a linear order, simply writing $f_1 < f_2 < \cdots < f_d$; each searching robot of the current phase visits some subsequence of graphs $H_{f_i}^{(2)}$ in ascending orders of labels. For each f_i, consider the set of black hole arcs exiting the component $H_{f_i}^{(2)}$. If at the start of the current phase some of these arcs exited different strongly connected components H_a, H_b (which were contracted into one by some searching robot), then the inductive claim holds, since for the current phase the value of $d^{(2)}$ is less than that of $d^{(3)}$ at the end of the previous phase. Thus, w.l.o.g. we can assume that $H_{f_i}^{(2)}$ is obtained from some component H_{f_i}, which is perhaps enlarged by searching robots, but has the same set of exiting black hole arcs as $H_{f_i}^{(2)}$, throughout the phase. With this assumption, we now proceed to show the following auxiliary lemma.

Lemma 3.1. *If multigraph $(\mathcal{H}, R)$ is acyclic, then we either have $d = 1$ and $H_{f_1}^{(2)} = G[V \setminus BH]$, or for all i, $1 \leq i \leq d^{(2)}$, the following condition holds:*

$$\text{outdeg}_{(\mathcal{H},R)} H_{f_i}^{(2)} \geq \left\lfloor \frac{\text{indeg}_{(\mathcal{H},R)} H_{f_i}^{(2)}}{\delta_i + 1} \right\rfloor.$$

Proof of Lemma. Let i be arbitrarily chosen, and consider an arbitrary sequence of $\delta_i + 1$ searching robots $r_1, \ldots, r_{\delta_i+1}$ which consecutively went through $H_{f_i}^{(2)}$, ordered by increasing release times. We will show that at least one of the robots did not use a black hole arc to leave $H_{f_i}^{(2)}$. Let $H_{f_i}(r_j)$ denote component H_{f_i} directly after the destruction of robot r_j; recall that we have $H_{f_i}(r_1) \subseteq H_{f_i}(r_2) \subseteq \cdots \subseteq H_{f_i}(r_{\delta_i+1}) \subseteq H_{f_i}^{(2)}$, and each $H_{f_i}(r_j)$ has the same set of black hole arcs as $H_{f_i}^{(2)}$. Suppose now, to the contrary, that each of the robots r_j left $H_{f_i}^{(2)}$ by a black hole arc; this means that each of these robots also left $H_{f_i}(r_j)$ by a black hole arc; moreover, since preference is given to arcs which were not used for a longer time, each of the robots $r_1, \ldots, r_{\delta_i}$ will be destroyed by a different black hole arc. Consider now the robot r_{δ_i+1}. By the strong connectivity property of graph $G[V \setminus BH]$, unless $H_{f_i}(r_{\delta_i+1}) = G[V \setminus BH]$, there exists at least one arc e outgoing from $H_{f_i}(r_{\delta_i+1})$ which is not a black hole arc. This arc could not have been visited by any robot r released between r_1 (inclusive) and r_{δ_i+1} (exclusive). Indeed, r never returns to component $H_{f_i}(r_{\delta_i+1})$ after leaving it by arc e (since otherwise arc e would have been included into $H_{f_i}(r_{\delta_i+1})$ by robot r_{δ_i+1}), and since we have $r \in \{r_1, \ldots, r_{\delta_i}\}$ (because r visits $H_{f_i}(r_{\delta_i+1}) \subseteq H_{f_i}^{(2)}$), one of the robots $\{r_1, \ldots, r_{\delta_i}\}$ would have not been destroyed by any black hole arc leading out of $H_{f_i}(r_{\delta_i+1})$, a contradiction. Hence, robot

r_{δ_i+1} finally leaves component $H_{f_i}(r_{\delta_i+1})$ by an arc which is not a black hole arc, so it must contribute to the out-degree of node $H_{f_i}^{(2)}$ in multigraph $(\mathcal{H}, R)$. Since the sequence of $\delta_i + 1$ consecutive robots going through $H_{f_i}^{(2)}$ was arbitrarily chosen, the claim follows directly. $\square$

Now, we make a simple structural claim which relies only on properties of directed acyclic multigraphs.

Lemma 3.2. *Let M be any directed acyclic multigraph having set of nodes $\{s, v_1, v_2, \ldots, v_d\}$, such that $\mathrm{indeg}(s) = 0$, and for all v_i, $1 \leq i \leq d$, we have: $\mathrm{outdeg}_M(v_i) \geq \left\lfloor \frac{\mathrm{indeg}_M(v_i)}{\delta_i+1} \right\rfloor$. Then: $\mathrm{outdeg}_M(s) < \prod_{i=1}^{d}(\delta_i + 1)$.*

Proof of Lemma. Without loss of generality, we may assume that $(s, v_1, v_2, \ldots, v_d)$ is a valid linear ordering of the vertices of M obtained by topological sorting of the vertices. Let m_i, $1 \leq i \leq d$, be the cardinality of the set of arcs of M having their starting point in a vertex from the set $\{s, v_1, \ldots, v_{i-1}\}$ and their endpoint in the set $\{v_i, \ldots, v_d\}$. We also put $m_{d+1} = 0$. Observe that by assumption $\mathrm{indeg}_M(v_i) \leq (\delta_i + 1)\,\mathrm{outdeg}_M(v_i) + \delta_i$, and for all i, $1 \leq i \leq d$, we have $m_{i+1} \geq \mathrm{outdeg}_M(v_i)$. Hence:

$$m_i = m_{i+1} + \mathrm{indeg}_M(v_i) - \mathrm{outdeg}_M(v_i)$$
$$\leq m_{i+1} + (\delta_i + 1)\,\mathrm{outdeg}_M(v_i) + \delta_i - \mathrm{outdeg}_M(v_i)$$
$$= m_{i+1} + \delta_i\,(\mathrm{outdeg}_M(v_i) + 1) \leq m_{i+1} + \delta_i(m_{i+1} + 1) = (\delta_i + 1)(m_{i+1} + 1) - 1.$$

From the above inequalities, we obtain by induction starting from $i = d$ the relation $m_i \leq \prod_{j=i}^{d}(\delta_j + 1) - 1$. Hence, $\mathrm{outdeg}_M(s) = m_1 < \prod_{j=1}^{d}(\delta_j + 1)$. $\square$

Lemma 3.3. *Let $\delta_1, \ldots, \delta_d$ be positive integers such that $\sum_{i=1}^{d} \delta_i = \Delta$. Then $\prod_{i=1}^{d}(\delta_i + 1) \leq 2^{\Delta} \cdot (\sqrt{3}/2)^{\Delta-d}$.*

Proof of Lemma. We will show a more general claim, namely, that for any sequence $\delta_1, \ldots, \delta_t$ of positive integers containing at most d elements equal to 1 and such that $\sum_{i=1}^{t} \delta_i = \Delta$, we have $\prod_{i=1}^{t}(\delta_i + 1) \leq 2^{\Delta} \cdot (\sqrt{3}/2)^{\Delta-d}$. Assume w.l.o.g. that $\delta_1 \leq \cdots \leq \delta_t$. First, we consider the case when $\delta_i \in \{1, 2\}$ for all $1 \leq i \leq t$, that is, for some $c \leq d$, $\delta_1 = \cdots = \delta_c = 1$ and $\delta_{c+1} = \cdots = \delta_t = 2$. Then, $\prod_{i=1}^{t}(\delta_i + 1) = 2^c \cdot 3^{(\Delta-c)/2} = \sqrt{3}^{\Delta}(2/\sqrt{3})^c \leq \sqrt{3}^{\Delta}(2/\sqrt{3})^d = 2^{\Delta} \cdot (\sqrt{3}/2)^{\Delta-d}$, and the claim holds. Analogously we show that the claim holds when $\delta_i \in \{1, 2\}$ for all $1 \leq i \leq t - 1$ and $\delta_t = 3$. Finally, for any sequence $\delta_1, \ldots, \delta_t$ which does not belong to one of the two considered classes it is immediate to construct a sequence $\delta'_1, \ldots, \delta'_{t'}$ containing the same number of elements equal to 1, having the same sum, and such that $\prod_{i=1}^{t}(\delta_i + 1) \leq \prod_{i=1}^{t'}(\delta'_i + 1)$. Indeed, if the sequence $\delta_1, \ldots, \delta_t$ contains an element $\delta_i > 3$ then we can replace this element by some number of elements $\delta'_{i_1}, \ldots, \delta'_{i_k} \in \{2, 3\}$ such that $\delta'_{i_1} + \cdots + \delta'_{i_k} = \delta_i$, without affecting the sum and the number of ones in the sequence, and increasing the considered product. Moreover, if the sequence $\delta_1, \ldots, \delta_t$ contains some two elements $\delta_{i_1} = \delta_{i_2} = 3$, then we similarly replace them by three elements $\delta'_{i_1} = \delta'_{i_2} = \delta'_{i_3} = 2$, retaining all the desired properties. This completes the proof of the lemma. $\square$

Now, recall that the number of searching robots released in the current p-th phase is chosen as $S_p = \lceil 2^{\Delta} \cdot (\sqrt{3}/2)^{p-1} \rceil$. Since by the inductive claim from phase $p - 1$, we have that $d \leq \Delta - p + 1$, we obtain from the preceding lemma: $\mathrm{outdeg}_{(\mathcal{H}, R)}(s) = S_p \geq 2^{\Delta} \cdot (\sqrt{3}/2)^{\Delta-d} \geq \prod_{i=1}^{d}(\delta_i + 1)$. This means that the claim of Lemma 3.2 does not hold, and so the assumption $\mathrm{outdeg}_{(\mathcal{H}, R)}H_{f_i}^{(2)} \geq \left\lfloor \frac{\mathrm{indeg}_{(\mathcal{H}, R)}H_{f_i}^{(2)}}{\delta_i+1} \right\rfloor$ is violated for some component $H_{f_i}^{(2)}$. From Lemma 3.1 we thus immediately conclude that $d = 1$ and $H_{f_1}^{(2)} = G[V \setminus BH]$, hence in this case the black hole search problem has been successfully solved. This completes the inductive proof that at the end of the p-th phase, $d^{(3)} \leq \Delta - p$. Thus, after phase $p = \Delta - 1$, we have $d^{(3)} \leq \Delta - p \leq 1$, and so at the start of phase Δ, $d = 1$, and all the nodes belong to the same strongly connected component H_{f_1}. The black hole search problem is thus solved in phase Δ. $\square$

Theorem 3.2. *In the synchronous model, 4 robots are always sufficient and sometimes required to solve the directed black hole search problem with $\Delta = 2$.*

Proof. The lower bound on the number of required robots is a direct consequence of Theorem 1.1. The upper bound is obtained by a similar strategy as that used in the proof of Theorem 3.1, simply releasing 4 searching robots r_1, r_2, r_3, r_4 in a single phase (without any cycle detection or cycle contraction robots). Let $\{e_1, e_2\} = BA$ be the two black hole arcs, originating from nodes v_1 and v_2, respectively. Robot r_1 may either successfully solve the black hole search instance, or may be destroyed on one of the black hole arcs, say e_1. In the latter case, robot r_2 will either successfully solve the instance or be destroyed in the other black hole arc e_2 (recall that robots choose to use unvisited arcs, whenever possible, and the graph is strongly connected). Suppose that robot r_3 is also destroyed in some black hole; this is possible, e.g., in the worst-case example from Theorem 1.1. Now, taking into account Lemma 3.2, we observe that at some point during the traversal performed by robot r_4, nodes v_1 and v_2 will necessarily belong to the same strongly connected component H_f ($f(v_1) = f(v_2) = f$). Once robot r_4 has reached this component, it will always exit it by unvisited arcs. Since both arc e_1 and e_2 have already been used (by robots r_1 and r_2, respectively), robot r_4 will never enter a black hole and will proceed to enlarge component H_f, until the whole of the graph has been explored, $H_f = G[V \setminus BH]$, and will thus successfully complete the task. $\square$
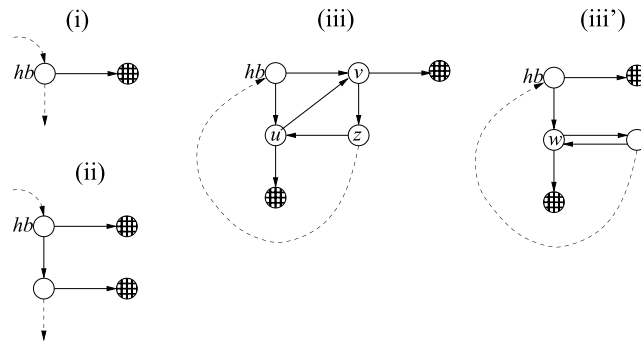
Fig. 2. The possible ways that an adversary has to defeat any strategy which aims to solve the directed black hole search problem when using less than 5 robots.

4. Results for the asynchronous model

Theorem 4.1. *In both the asynchronous and synchronous models, 2 robots are always sufficient and sometimes required to solve the directed black hole search problem with $\Delta = 1$.*

Proof. The lower bound on the number of required robots is a direct consequence of Theorem 1.1. The upper bound (which we obviously need to show for the asynchronous model, only) is obtained by simultaneously releasing two robots, each of which applies a strategy identical to that used by searching robots in the proof of Theorem 3.1. It is easy to observe that, throughout the process, each robot is either traversing arcs of an already identified strongly connected subgraph H_f of $G[V \setminus BH]$, or is exiting such a subgraph by some outgoing arc e. Arc e has either never been visited before by any robot, or is the unique arc exiting H_f, which, due to the strong connectivity property of $G[V \setminus BH]$, has to lead to some node of $V \setminus BH$. Thus, a robot may potentially enter a black hole only when using an arc which has been never visited before by any robot. Since $\Delta = 1$, this means that the black hole may destroy at most one robot, since the remaining robot will not traverse the unique black hole arc at any later time. The surviving robot(s) perform a search of the graph, exploring unvisited arcs, until the whole of the strongly connected component $G[V \setminus BH]$ has been discovered and the search is complete. □

Theorem 4.2. *In the asynchronous model, at least 5 robots are sometimes required to solve the directed black hole search problem with $\Delta = 2$.*

Proof. The proof considers some possible cases that must be solved by any strategy $\mathcal{A}$ for the black hole search problem with $\Delta = 2$. We first assume that 4 robots are sufficient for solving the problem and then we obtain the claim by showing how an adversary can defeat all the robots. As usual, all the 4 robots start from hb which has two outgoing arcs. All of the examples we consider are such that the only arc of the graph returning to the homebase (marked in Fig. 2 with a dashed line) will in some cases never be traversed by a robot. Consequently, we confine ourselves to an analysis of strategies in which all the robots leave the homebase without waiting for any robots to return.

According to $\mathcal{A}$, there are three possible cases for the first move of each robot and three corresponding strategies of the adversary (see Fig. 2 for a visualization): (i) all of them follow one direction; (ii) one goes through one direction and the remaining three to the other; (iii and iii') the four robots are equally distributed among the two directions.

If (i), then the adversary can locate one black hole on the reached node, and all the robots get destroyed. If (ii), then the adversary can locate one black hole on the node reached by the majority of the robots, hence destroying three of them. The surviving robot reaches a safe node admitting another branching off. One of the two options leads to the black hole, while the other one to hb. As the robot has no further knowledge to distinguish between the two options, the branch it chooses will be set by the adversary as the one leading to the black hole, and again all the robots get destroyed. If (iii), by referring to Fig. 2 we have that after the first move, two robots are on node u and two robots are on node v. From the point of view of the pair of robots located at u (respectively, v), this situation is indistinguishable from that of the pair of robots located at w in graph (iii'), where allowing any robot to wait could lead to an infinite deadlock, since all other robots may be destroyed. Hence, all the four robots at u and v in (iii) must move, without waiting for any other robot. The pair of robots on u (resp. v) cannot choose to move along the same arc because they may both be destroyed by the adversary. So, they have to move on distinct arcs. This implies that one robot from u and one from v get destroyed. The snapshot of the network now gives a robot on v and another on z. From v, both outgoing arcs have been already traversed and hence the robot has no further knowledge to distinguish among the two options apart from the identifiers of the robots that have been passing through such arcs. However, these IDs can be exchanged by the adversary by swapping the labels of ports exiting v, thus making this information useless. The robot is thus forced either to wait at v, or enters the black hole. The other robot, which is on z, has again two unexplored options, one leading to hb, and the other to u. The adversary forces the robot to reach u, from there it is analogously forced either to wait or to enter the black hole. Thus, all the robots are either destroyed or deadlocked, and the claim holds. □

5. Conclusions

In this paper, we have considered the black hole search in directed graphs under both the synchronous and the asynchronous settings. The obtained results reflect the difficulty of the problem on directed graphs as well as the difference between the synchronous and the asynchronous settings. It is worth noting how the behavior of the robots must change even for small values of Δ, such as 2. This was not the case for undirected graphs. This makes the study of the problem on directed graphs even more intriguing. One major remaining open problem concerns the possibility of bounding the number of required robots in terms of Δ in the asynchronous case.

Many variations of the problem still deserve investigation. Changing the assumptions to the initial knowledge of the robots, for instance by (partially) revealing the graph to the robots or providing topological information about the location of the nodes and ordering of the edges, could lead to completely different results.

References

[1] B. Balamohan, P. Flocchini, A. Miri, N. Santoro, Time optimal algorithms for black hole search in rings, in: Proc. of 4th International Conference on Combinatorial Optimization and Applications, COCOA, LNCS vol. 6508, 2010, pp. 58–71.
[2] C. Cooper, R. Klasing, T. Radzik, Locating and repairing faults in a network with mobile agents, Theoretical Computer Science 411 (14–15) (2010) 1638–1647.
[3] J. Czyzowicz, D. Kowalski, E. Markou, A. Pelc, Searching for a black hole in synchronous tree networks, Combinatorics, Probability and Computing 16 (2007) 595–619.
[4] J. Czyzowicz, S. Dobrev, R. Kralovic, S. Miklik, D. Pardubska, Black hole search in directed graphs, in: Proc. of 16th International Colloquium on Structural Information and Communication Complexity, SIROCCO, LNCS, vol. 5869, 2009, pp. 182–194.
[5] S. Dobrev, P. Flocchini, R. Kralovic, G. Prencipe, P. Ruzicka, N. Santoro, Black hole search in common interconnection networks, Networks 47 (2) (2006) 61–71.
[6] S. Dobrev, P. Flocchini, G. Prencipe, N. Santoro, Searching for a black hole in arbitrary networks: optimal mobile agents protocols, Distributed Computing 19 (1) (2006) 1–18.
[7] S. Dobrev, P. Flocchini, G. Prencipe, N. Santoro, Mobile search for a black hole in an anonymous ring, Algorithmica 48 (1) (2007) 67–90.
[8] S. Dobrev, P. Flocchini, N. Santoro, Improved bounds for optimal black hole search in a network with a map, in: Proc. of 11th International Colloquium on Structural Information and Communication Complexity, SIROCCO, LNCS, vol. 3104, 2004, pp. 111–122.
[9] L. Gąsieniec, R. Klasing, R.A. Martin, A. Navarra, X. Zhang, Fast periodic graph exploration with constant memory, Journal of Computer and System Sciences (JCSS) 74 (5) (2008) 802–822.
[10] P. Glaus, Locating a black hole without the knowledge of incoming link, in: Proc. of 5th International Workshop on Algorithmic Aspects of Wireless Sensor Networks, ALGOSENSORS, LNCS, vol. 5804, 2009, pp. 128–138.
[11] R. Klasing, E. Markou, T. Radzik, F. Sarracco, Approximation bounds for Black Hole Search problems, Networks 52 (4) (2008) 216–226.
[12] A. Kosowski, A. Navarra, Graph Decomposition for Improving Memoryless Periodic Exploration, in: Proc. of 34th International Symposium on Mathematical Foundations of Computer Science, MFCS, LNCS, vol. 5734, 2009, pp. 501–512.
[13] A. Kosowski, A. Navarra, C.M. Pinotti, Synchronization Helps Robots to Detect Black Holes in Directed Graphs, in: Proc. of 13th International Conference on Principles of Distributed Systems, OPODIS, LNCS, vol. 5923, 2009, pp. 86–98.