Black Hole Search in Computer Networks: State-of-the-Art, Challenges and Future Directions

Mengfei Peng[a], Wei Shi[a], Jean-Pierre Corriveau[b], Richard Pazzi[a], Yang Wang[c]

[a] *Faculty of Business and Information Technology*
University Of Ontario Institute Of Technology, Oshawa, Canada
[b] *School of Computer Science*
Carleton University, Ottawa, Canada
[c] *Shenzhen Institute of Advanced Technology*
Chinese Academy of Science, Shenzhen, China

Abstract

As the size and use of networks continue to increase, network anomalies and faults are commonplace. Consequently, effective detection of such network issues is crucial for the deployment and use of network-based services. In this paper, we focus on one specific severe and pervasive network problem, namely the presence of one or more *black holes*. A black hole models a network node that is accidentally off-line or in which a process deletes any visiting agent or incoming data upon arrival without leaving any observable trace. *Black Hole Search* is the process that leverages mobile agents to locate black holes in a fully distributed way. In this paper, we review the state-of-the-art research in this area. We first distinguish between solutions for synchronous and asynchronous networks. We then consider the communication model between agents, their starting locations and the topological knowledge each may hold. We also report on the proposed algorithms with respect to their complexity and correctness. We remark that most existing work addresses locating a single black hole, multiple black hole search being significantly more complex. We not only summarize major results in this area but also briefly touch on other types of malicious hosts. Finally, we identify some open problems for future research.

Keywords: black hole search, mobile agents, multiple black holes, malicious host, network diagnosis

1. Introduction

Over the past few decades, as network-based services have become prevalent, so has the need for effective diagnosis of all-too-frequent network anomalies and

Email addresses: `mengfei.peng@uoit.ca` (Mengfei Peng), `wei.shi@uoit.ca` (Wei Shi), `jeanpier@scs.carleton.ca` (Jean-Pierre Corriveau), `richard.pazzi@uoit.ca` (Richard Pazzi), `yang.wang1@siat.ac.cn` (Yang Wang)

faults. Among these, a *black hole* is a severe and pervasive problem. A black hole models a computer that is accidentally off-line or a network site in which a resident process (e.g., an unknowingly-installed virus) deletes any visiting agents or incoming data upon their arrival without leaving any observable trace [30]. For example, in a cloud, a node that causes loss of essential data (for the system and/or its users) constitutes a black hole and *de facto* compromises the quality of any service in this cloud. Similarly, any undetectable crash failure of a site in a network transforms that site into a black hole.

A *mobile agent* is an abstract and autonomous software entity. As such, agents are versatile and robust in changing environments, and can be programmed to work in cooperative teams. Members of such teams may have different complementary specialties, or be duplicates of one another [53]. For *black hole search*, one or a team of identical agents are generally used. These agents have limited computing capabilities and bounded storage. They all obey an identical set of behavioural rules (referred to as the "protocol") and can move from a node to a neighbouring one. Also, these agents are anonymous (i.e., do not have distinct identifiers) and autonomous (i.e., each does its own computing and uses its own memory).

Using such agents offers several potential advantages: they can reduce network load, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, and even adapt dynamically [61]. For example, black hole search may instead rely on the use of a central controller. In this case, the latter must constantly send Ping messages to nodes or, alternatively, require that each node send it periodically a message confirming this node's activity. Both of these strategies lead to heavy network traffic that can be avoided when using mobile agents for such a search.

Consequently, in this paper, black hole search is scoped to be a task that allows a team of mobile agents to collaborate with each other to locate black holes within finite time while eventually leaving at least one agent to survive and know all the edges leading to black holes [37]. (We abstract a network into a graph $G(V, E)$ where nodes in V represent computer hosts and edges in E represent network links.) Currently, many distinct approaches to using mobile agents to locate a *single* black hole in a computer network have been studied in many different contexts (e.g., [6, 17, 19, 28, 43, 57]). Generally, existing solutions rest on anonymous agents that all execute the same protocol to identify and report any black hole.

In 2006, Flocchini *et al.* [50] scrutinized the black hole search problem for both asynchronous and synchronous networks. That survey also introduced the black hole search problem as a special case of exploring and mapping an unknown environment. While there exists a large body of literature on unknown graph exploration problems, it is mostly irrelevant to this paper for it generally assumes that the underlying network graph does not contain any type of malicious entities [3]. Conversely, work on *dangerous graph search* (e.g., [18]) does address the detection and localization of malicious hosts (such as black holes), malicious agents, and faulty links. In particular, in their 2012 survey [64], Markou *et al.* discussed previous research on identifying hostile nodes.

They mainly focused on synchronous special trees, arbitrary trees and arbitrary graphs, with a brief mention of asynchronous rings. More recently, Zarrad *et al.* [69] briefly discuss solutions for black hole search in synchronous and asynchronous networks, however without analyzing the underlying assumptions of these solutions.

In this paper, our goal is to review the state-of-the-art in the black hole search field in order to help readers understand the existing work, as well as grasp some of the remaining challenges in this field. We specifically exclude from the scope of this paper the issue of *black hole attacks* [2, 8, 68], which is superficially related to the topic at hand.

First, we introduce the main models and assumptions that are commonly used in the relevant literature with respect to network synchronization, the communication model between agents, their starting locations and the topological knowledge each may hold. In addition to obtaining models, determining their complexity is also critical for the actual deployment of the proposed algorithms. Possibly relevant measures of complexity include the total number of moves, the number of agents, the number of tokens, and the memory footprint, as well as algorithm efficiency *per se*. Time cost is another metric that is usually discussed when considering synchronous networks. Because the time cost of transit (i.e., moving from one node to a neighbouring one) is unpredictable in asynchronous networks, in such networks time complexity can only be measured using additional assumptions such as: it takes an agent an unitary amount of time (i.e., one 'time unit') to traverse a link or explore a node (which amounts to having a global clock) [5, 6, 30].

We then separate the papers of this survey based on their network synchronization (i.e., synchronous or asynchronous). The motivation for this is simple: when considering synchronization, the black hole search problem is very different with respect to it allowed behaviour(s), its inherent difficulty and its limitations, and so are the proposed solutions. For both categories, we further classify the studies based on the agent communication model, the agent starting locations, and knowledge of the network. That is, we contrast the proposed solutions with respect to their choice of assumptions (and resulting complexity) in each of the three areas of variability just mentioned for black hole search. Beyond such comparisons, we also briefly introduce some open problems that persist in this field.

More specifically, the rest of this paper is organized as follows: Frequent assumptions and models for black hole search are introduced in Section 2, then relevant measures of complexity are discussed in Section 3. Solutions for the detection of a single black hole in synchronous and asynchronous networks are respectively addressed in Section 4 and Section 5. In Section 6 we consider multiple black holes search. We then report in Section 7 on the most recent results pertaining to the different types of malicious hosts. In Section 8, we summarize the contributions we survey and mention some open problems stemming from this work. We draw some conclusions in Section 9.

Table 1: Models and Assumptions Frequently Used for Black Hole Search

Network synchronization	Communication model	Agent starting location	Knowledge of Network
Synchronous network	Pure token	Co-located	No knowledge (e.g., unknown)
	Enhanced token		Edge-labelled (e.g., sense of direction)
Asynchronous network	Whiteboard	Dispersed	Network topology (e.g., ring)
	Face-to-face		Complete knowledge (e.g., map)

2. Common Models and Assumptions

Because none of the existing algorithms are able to solve the black hole search problem without some restrictions, it is crucial to gather the assumptions that are typically made in existing research and study the impact of each one. In this section, we introduce a list of such assumptions.

To start with, existing work always assumes that the agents' initial wake-up nodes are safe. Otherwise, all the agents may die before even starting graph exploration, rendering the problem unsolvable. Furthermore, unless the agents are extremely fortunate, (viz., happen to explore all nodes in a graph except the black hole(s)) in order to systematically identify a black hole, we must expect at least one agent to go in a black hole and somehow leave a hint for the other agents before it dies, which eventually allows the surviving agents to know the location of the black hole(s). All other common assumptions are listed in Table 1. We will now provide a detailed explanation of each of these assumptions.

2.1. Network Synchronization

2.1.1. Synchronous Network

A *synchronous network* is a network in which all agents initially wake up at the same time and where it takes a quantum amount of time (called a *time unit*) for an agent to traverse a link or explore a node: All agents are thus synchronized with respect to a global clock. By the end of each time unit, an agent must decide whether to move to a neighbouring node, or stay at its current node, or terminate the algorithm. As such, the complexity of the agent's algorithm in synchronous networks can be measured in terms of the number of time units.

In synchronous networks, a *time-out mechanism* is available to enforce the time synchronization [17, 22, 23, 24, 56]. Such a mechanism allows us to easily identify which agents died in the black hole(s). Suppose a team of agents should meet at a node u after m time units, after this time-out, all other agents know that those that do not show up in node u died in the black hole(s).

Using such a time-out mechanism, the black hole can be located using only 2 agents in any network that has only one black hole present when a network map is available for every agent. In this case the network size is not required to guarantee a solution. For example, let 2 agents, a and b, be at a safe node u. Assume agent a moves to the neighbouring node v and is expected to return to u while agent b waits at node u. As each move takes 1 time unit, if agent a does not come back to node u after 2 units, then agent b knows that agent a is dead and that node v is the black hole. Once agent b knows the location of the black hole, the algorithm can terminate immediately even if there are remaining unexplored nodes in the network.

Furthermore, with this mechanism, it is also possible to know whether or not a black hole exists. More specifically, if all n nodes of the network have been explored by the end of a predefined time-out, we can conclude that there is no black hole in this network, provided that n be known *a priori*. In this case, Klasing *et al.* [57] and Czyzowicz *et al.* [22] solve the black hole search problem under the assumption that there is one or no black holes in the network.

2.1.2. Asynchronous Network

Unlike for synchronous networks, there is no global clock mechanism in asynchronous networks. Thus, the agents could initially wake up at different times. Also, the time that an agent takes for every action (sleep or transit) is finite but unpredictable [50]. Therefore, it is impossible to distinguish whether an agent died in a black hole or is stuck in a slow link/node of the network since the latter possibility takes an unpredictable amount of time [67]. It follows that the only way to locate a black hole in an asynchronous network is to explore the *entire* network [50]. Consequently, the network size n and the number of black holes b must be known *a priori* in order to count the total number of explored nodes (whether for single or multiple black hole search): only when at least $n-b$ nodes are explored may the algorithm terminate.

While knowing network size n is not required for solving black hole search in synchronous networks, generally it is for asynchronous ones. Alternatively, the total number of edges m could possibly be used in lieu of n. In theory, once every edge that leads to the black hole would have been marked as dangerous, the search would complete after an agent finishes traversing all edges except the dangerous ones. Since such an approach has not been yet studied in any research paper, knowing m but not n will not be considered in the rest of this survey. That is, hereafter we will assume n is given in the case of asynchronous networks.

Also, a network may be disconnected due to the presence of a black hole. In the context of an asynchronous network, this makes it is impossible for an agent to finish exploring the entire network and terminate the algorithm. In order to bypass this roadblock, research papers that study the single black hole search problem in asynchronous networks assume that the network is bi-connected or at least that the network remains connected after removing the black hole node. In contrast, synchronous networks need not be bi-connected for single black hole search. For example, Czyzowicz *et al.* [24] study this problem in tree networks.

Finally, we remark that this issue is far more complex when considering multiple black hole search (which we will address in Section 6). Thus, hereafter until that section, we will focus on single black hole search.

2.2. Communication Models

Given the location of the black hole is initially unknown, regardless of network synchronization, an agent may die at any time while it explores. As previously mentioned, in order to systematically identify a black hole, a team of agents is used to locate the black hole. Collaboration between agents is not only necessary; it is essential. To this end, the agents are usually assumed to communicate with each other using one of the four communication models: the *pure token model*, the *enhanced token model*, the *whiteboard model*, and the *face-to-face model*. In the first three of these models, agents have no means for direct communication between themselves.

Before discussing each of these models, we remark that a crucial goal of agent communication is to minimize the number of agents that die in a black hole. To this end, it is assumed that at most one agent should be allowed to enter the same node at the same time via the same link. More specifically, when a port is explored for the first time, this initial exploration must involve only one agent that, before entering this port, must somehow indicate to other exploring agents that the node to which this port leads is currently under exploration and thus is to be considered dangerous until proven otherwise. Such a strategy, called *Cautious Walk*, is commonly used in black hole search algorithms for it prevents other agents from entering a node under exploration via the same link. It was first introduced by Dobrev *et al.* [30] to minimize the number of agents that die in the black hole. It typically requires that a node be conceptualized as having *ports*. A port can be classified as a) *unexplored* - no agent has ever passed through this port, or b) *dangerous* - an agent left via this port but no agent has returned through it, or c) *safe* - an agent has left and returned through this port. How the status of a port is captured differs between communication models. Regardless, Cautious Walk guarantees that no agent leave a node via a dangerous port. Consequently, if node v is a black hole and can be accessed from port p of a neighbour of v, then at most one agent will die via p.

2.2.1. Whiteboard Model

In the *whiteboard model* introduced by Dobrev *et al.* [30], each node has a bounded amount of storage where information can be written and read by agents. All incoming agents can access the whiteboard of a node in a fair mutual exclusion way and communicate with each other via reading/writing on such whiteboards.

When executing the cautious walk, an agent leaves from a node u to a neighbouring node v via an unexplored port p. It marks port p as dangerous by writing on the whiteboard of node u. After visiting node v, this agent immediately returns to node u in order to update its whiteboard so that the status of p is changed from dangerous to safe.

2.2.2. Pure Token Model

In the *pure token model*, each agent has a limited number of tokens that can be placed on or picked up at a node in the course of searching. An agent places one or more tokens at its current node u to indicate that the 'next' node it visits is dangerous. (More precisely, each node has a single location, referred to as its 'center', where to place tokens.) Mutiple tokens may be required in order to capture which of the neighbouring nodes of u is this 'next' node visited by the agent at hand.

The pure token model can be considered as a special whiteboard model with $O(1)$-bit memory on each node. Tokens that can be picked up from a node and placed on another are called *movable tokens*. In contrast, Chalopin *et al.* [16] define *unmovable tokens* as those that cannot be picked up once placed on a node. Unless specified otherwise, the tokens mentioned in this paper are movable ones and are all identical by default (that is, they cannot be distinguished one from the other).

2.2.3. Enhanced Token Model

Clearly, the pure token model has strong limitations, in particular with respect to the limited number of messages that can be expressed using a constant number of tokens. In light of such constraints, many researchers (e.g., [28, 37, 39]) enhance the pure token model in order to increase the information that can be expressed via tokens. More specifically, in the *enhanced token model*, the tokens can be left not only at the 'center' of a node, but also on the ports of a node. But as the number of locations to hold the tokens increases at each node, so does the memory cost of each node. Typically, the memory cost is set to $O(\log n)$ bits in the whiteboard model, $O(\log \Delta)$ in the enhanced token model, and $O(1)$ in the pure token model, where n is the network size and Δ is the maximum node degree in the network graph[1] [17].

When executing cautious walk under this model, an agent marks a port as dangerous by placing a token at this port before moving to the next node. (As for the whiteboard model, no agent will leave via a dangerous port, that is, in this model, a port at which a token is present [37].) Upon its return, this agent will pick up this token to show that this port is not dangerous. Reusing movable tokens in several different nodes helps minimizing the overall number of tokens used, a challenge not faced in the whiteboard model (since, once written to a whiteboard, a message may be repeatedly accessed by agents over a long period of time, and even be modified). However, the use of movable tokens results in a significantly more complex communication model than a) one with only unmovable tokens and b) a whiteboard model (in which messages written to a whiteboard are far easier to use than tokens).

[1] Note that, unless specified otherwise, we will use these definitions of n and Δ throughout this survey paper.

2.2.4. Face-to-Face Model

In the *face-to-face model*, agents move through the network in synchronous steps and communicate with each other only when they meet at a node [19]; no other communication method (e.g., whiteboard or tokens) is available. In contrast to the three communication models mentioned above, face-to-face communication does not require that nodes have memory. Finally, clearly, face-to-face communication only applies to synchronous networks: the unpredictability of wake up times and of time required to move and/or compute in asynchronous networks entails agents may never meet.

2.3. Agent Starting Location

The starting location of an agent is another factor that significantly affects black hole search. Since at least 2 agents are necessary to locate the black hole, the agents could start at the same node or different nodes. More generally, with respect to starting locations, agents may be:

- Co-located: all the agents initially wake up at the same node, and this node is referred to as *homebase*;
- Dispersed: the agents wake up at different nodes. The node in which an agent wakes up is its *homebase*. Dispersed agents are also occasionally referred to as *scattered agents*. Hereafter in this paper, we use the former term.

In both cases, all homebases are assumed to be safe. Otherwise, the black hole search problem is unsolvable.[2] Moreover, each dispersed agent only knows its own homebase and, upon waking up, there is no communication between the dispersed agents. In contrast, upon waking up, co-located agents can communicate, which can can lead to guaranteed coordination [67].

Finally, for synchronous networks, if the face-to-face model is adopted, then only co-located agents must be used: should agents be dispersed, there is a possibility that all will die in the black hole before they ever meet. That is, only co-location guarantees face-to-face communication.

2.4. Network Knowledge

What agents know about the network considerably affects both the design and complexity of a solution to black hole search. This knowledge includes some (if not all[3]) of the following: network size, network topology, network direction, edge-labelling and sense of direction.

[2]Since, if a homebase is a black hole, all agents waking up at that homebase die immediately thereby eliminating the possibility of a successful search.

[3]An unrealistic case for which the black hole search problem becomes much less complex.

2.4.1. Network Size

Network size refers to the total number of nodes in the network (denoted by n in this paper). As mentioned before, if the agents do not know the number of nodes nor the number of edges in the network, then the black hole search problem is unsolvable in an asynchronous network. In addition, the problem is also unsolvable in the asynchronous network if the number of black holes is not known *a priori*.

2.4.2. Network Topology

Network topology refers to the topological structure of the network abstracted as a graph (e.g., a ring, a torus, etc.). Many algorithms are specifically designed for certain network topologies. For example, in [38, 40], Dobrev *et al.* provide a protocol called *shadow check* that only works on ring networks. A ring is a fundamental network topology in the context of black hole search for it is the basis for more complex topologies (e.g., torus and hypercube).

In synchronous networks, when the agents have no knowledge of network size nor possess a network map, the black hole search problem can still be solved with only 2 agents if the network topology is known. For example, such solutions exist for rings [17] and tori [15] networks.

In both synchronous and asynchronous networks, when the agents have no topological knowledge, at least $\Delta + 1$ agents are needed in any generic solution, even if the agents are given the network size n and the maximum node degree Δ [50]. If the black hole is a node with degree Δ, then there are Δ ports leading to the black hole that have to be marked as dangerous. Since one agent dies for each dangerous port to mark, and given at least one agent has to survive to eventually report the black hole location, it follows that at least $\Delta + 1$ agents are necessary.

2.4.3. Network Direction

Network direction refers to whether a graph is directed or undirected (e.g. bi-directional). Most importantly, we remark that most commonly used techniques for black hole search (e.g., the previously mentioned Cautious Walk) can only be used in undirected graphs. Although results for the exploration of directed graphs have appeared since the mid-1990s (e.g. [10, 11, 51]), the first study dealing with black hole search in directed graphs was published by Czyzowicz *et al.* [21] only in 2010. It shows that as many as 2^d agents are possibly required (where d denotes the in-degree of the black hole node) [*Ibid.*]. Additional research on black hole search in directed graphs can be found in [21, 58, 59]. Unless otherwise indicated, in the rest of this paper we will focus on the use of undirected graphs.

2.4.4. Edge-labelling and Sense of Direction

An *edge-labelled graph* is one where at each node x, there is a distinct label associated with each one of its ports and the incident link of each port. Let $\lambda_x(x, z)$ denote the label associated at x with the link $(x, z) \in E$, and λ_x denote

Table 2: Relationships between network direction and sense of direction

<table>
<tr><td rowspan="5">Directed Graph</td><td colspan="4">Undirected Graph</td></tr>
<tr><td rowspan="4">Edge Unlabelled</td><td colspan="3">Edge-labelled</td></tr>
<tr><td>Arbitrarily Labelled</td><td colspan="2">Consistently Labelled</td></tr>
<tr><td rowspan="2">Un-oriented: No Sense of Direction</td><td>Oriented</td><td>Un-oriented</td></tr>
<tr><td>Sense of Direction</td><td>No Sense of Direction</td></tr>
</table>

the overall injective mapping at x. The set $\lambda = \{\lambda_x | x \in V\}$ of those mappings is called a *labelling* and we shall denote by (G, λ) the resulting edge-labelled graph. The nodes of G can be anonymous (e.g., without unique names) [35]. When visiting a node in an edge-labelled network, an agent can distinguish the ports of this node, whereas this is not possible in an edge-unlabelled network.

Sense of direction occurs in an edge-labelled undirected graph if, from any given node u, it is possible to determine whether or not different paths from node u will end in the same node. More precisely, in order to obtain a sense of direction, a *consistent coding function* and a *consistent decoding function* must be defined [49].

For example, in a ring network, if each port is labeled as A or B and such labeling is *consistent*, we say this ring has a sense of direction. Such a labeling is consistent if starting from some specific port and following a specific convention for traversal (e.g., 'A-B-A-B-....-A-B-A' or 'A-A-B-B-....A-A-B-B-A-A'), an agent can traverse the ring of n nodes and return to its starting port. A ring with a consistent labeling (e.g., all ports going in the clockwise direction are labelled *Right*) is commonly referred to as an *oriented* ring. Otherwise a ring is referred to as an *unoriented* one.

We further clarify the relationships between the network direction and the sense of direction in Table 2.

2.4.5. Complete Knowledge

Complete knowledge refers to the case where the agents know the size, topology and sense of direction (e.g., torus with consistent and systematic "N-S-E-W" labelling) of the network. Sometimes, agents are equipped with a network *map* that holds all this knowledge and can also be used to mark the explored nodes during a black hole search [27]. In this model, the black hole search problem becomes much less complex.

3. Cost Analysis Metrics

Complexity analysis is generally used to compare different solutions to black hole search with respect to specific costs. The most frequently measured costs are:

- Number of agents: the minimal number of agents used to solve the black hole search problem.
- Number of agent moves: the total number of moves performed by all agents from the first agent waking up until the black hole has been located.
- Number of tokens: the minimal number of tokens used by each agent (or by the entire agent team) in order to locate the black hole.
- Memory footprint of agents: the memory overhead of agents. Usually, in models relying on tokens, the agents are designed with a small memory footprint (e.g., an agent can only carry a constant number of tokens at any point in time [15, 16]). In other types of models, agents may have a very large memory footprint (e.g., agents carrying a network map [42, 56]).
- Memory footprint of nodes: the memory overhead of each node in the network. For example, a $O(\log n)$-bits whiteboard is sufficient for all the algorithms proposed in [6, 26]. Recall that the pure token model can be viewed as a whiteboard model with $O(1)$-bit memory on each node when assuming that only a constant number of tokens can be placed at a node [16]. However, in practice, memory overhead is considered mostly for whiteboard models. Instead, not surprisingly, in token models, the number of tokens is taken to be much more relevant.
- Time cost: In synchronous networks, this metric is computed as total number of time units used from when algorithm starts until the black hole is found. Given that, in an asynchronous network, a move of an agent costs finite but unpredictable time, generally time cost is not measured. However, some research [5, 6, 30] assumes a unitary time delay for each move, which enables the calculation of time complexity. Such a measure is referred to as *ideal time*. Under this assumption, time cost is almost the same as the number of agent moves.

Beyond measures of complexity, evaluations of correctness are also a commonly presented in black hole search work. Most papers in this area use mathematical proofs (e.g., [6, 15, 22, 28, 43, 44, 57]), while only a few researchers conduct simulations and use the results of such experiments to demonstrate correctness [25, 67]. For example, Shi *et al.* [67] present their simulation results for three proposed algorithms in addition to providing theoretical proofs. Similarly, D'Emidio *et al.* [25] simulate and compare their own algorithms before further analysis is used to decide which one performs better.

4. Black Hole Search in Synchronous Networks

In this section, we overview solutions for black hole search in synchronous networks. Given no existing research has used the enhanced token or the whiteboard models in synchronous networks, our presentation will follow the different possibilities given in Figure 1.

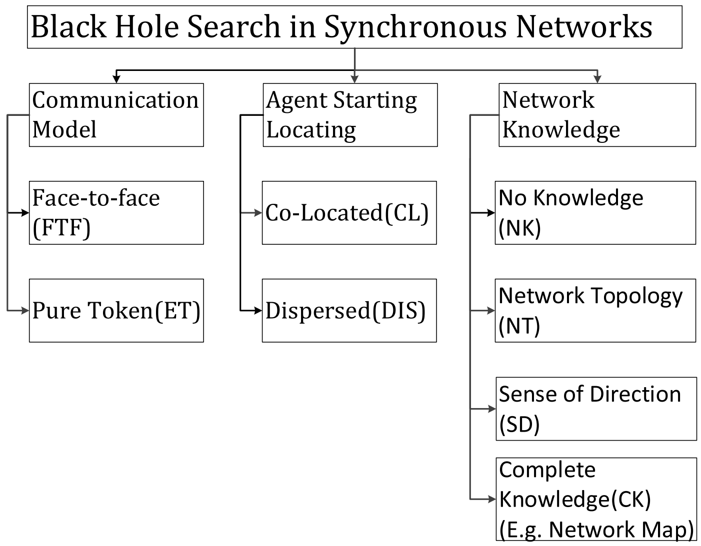

Figure 1: Different Variants for Black Hole Search in Synchronous Networks

4.1. Solutions under Different Communication Models

4.1.1. Face-to-Face Model

Recall the face-to-face model is only possible in synchronous networks. According to this model, agents simultaneously present in the same node can communicate with each other using an unlimited number of messages.

Czyzowicz *et al.* [22, 23, 24] and Klasing *et al.* [55, 56, 57] consider the problem of finding the most efficient solution (in terms of time cost) for the black hole search under the same assumption: 2 co-located agents with maps searching for a black hole in an edge-labelled undirected synchronous network. Instead, Chalopin *et al.* [16] study the problem using a hybrid communication model: agents can carry and place a bounded number of pure tokens *and* can communicate with each other when they meet on a node. Since that work focuses more on the impact of the tokens, we discuss it in the section on the pure token model (Section 4.1.2).

Under the assumptions they make, Czyzowicz *et al.* [23] show that the optimal black hole search problem is NP-hard, and propose a 9.3-approximation algorithm for it. Additionally, Klasing *et al.* [57] prove that this problem cannot be approximated in polynomial-time using a constant factor less than $\frac{389}{388}$ (unless P=NP), and give a 6-approximation algorithm. In both [57] and [23], each agent carries a network map and starts from the same node. But whereas the algorithm proposed by [23] can solve the problem when there is one and only one black hole in the network, the solution in [57] can first detect whether there is a black hole and then locate this black hole if present. (Recall, as previously mentioned, that such detection is only possible in a synchronous network.)

In [22, 24], Czyzowicz *et al.* present a $\frac{5}{3}$-approximation algorithm in an arbitrary tree without a map. This result exemplifies the impact of network knowledge: knowing the topology at hand reduces not only the time complexity but also the memory footprint of each agent. The authors introduce algorithms for two specific classes of trees namely: a) lines and b) trees in which all internal nodes have at least 2 children. The algorithm in [56] follows an intuitive approach of exploring the network graph via a spanning tree. Then, Klasing *et al.* [56] prove that this approach cannot lead to an approximation ratio bound better than $\frac{3}{2}$. Furthermore, they provide [*Ibid.*] a $3\frac{3}{8}$-approximation algorithm for an arbitrary network with the help of a network map. This result is a direct improvement from the $\frac{7}{2}-$ approximation algorithm presented in [55].

4.1.2. Pure Token Model

Chalopin *et al.* [15, 16, 17] and Markou *et al.* [63] focus on locating the black hole using a minimum number of agents and tokens, while the agents have $O(1)$ memory size and carry $O(1)$ pure tokens. Most importantly, in these solutions, agents do not know n or k, where n is the number of nodes in the network and k is the number of agents [4]. The authors consider both movable and unmovable tokens in rings [16] and tori [15, 63] respectively.

As previously mentioned, in [16], Chalopin *et al.* consider the black hole search problem with agents that have hybrid communication capabilities: they can communicate with each other face-to-face when they are in the same node and they can also carry either movable or unmovable tokens. When using movable tokens, 3 agents, each of which carrying only 1 token, are necessary and sufficient for both oriented and un-oriented rings. In contrast, using unmovable tokens, 4 agents are required, each with 2 tokens, for oriented rings and 5 agents, each with 2 tokens, when exploring un-oriented rings. Expressing messages using unmovable tokens is equivalent to writing messages on whiteboards with limited memory. Given this observation, one might expect the use of unmovable tokens to be more 'powerful' than that of movable ones. Interestingly, the results show that using unmovable tokens is more costly than that using movable one with respect to number of agents used. Furthermore, results show that more agents are necessary for un-oriented rings than for oriented rings.

In addition to rings, Chalopin *et al.* [15] also study the oriented torus under the same assumptions: dispersed agents, pure token model and face-to-face communication. They prove that the black hole search problem is unsolvable in synchronous oriented torus in three scenarios: 1) when the number of agents is constant and tokens are unmovable; 2) when using 2 dispersed agents, even if the tokens are movable and the agents have unlimited memory; 3) when using 3 agents with constant memory and 1 movable token each. Ultimately, they show that at least 3 agents, each with 2 movable tokens, are necessary and sufficient to solve the problem in any oriented torus.

[4]Note that, unless specified otherwise, we will use these definition of k throughout this survey paper.

In [63], Markou *et al.* study the black hole search problem under the same assumptions as [15] but in an un-oriented torus. The authors discuss four cases of un-oriented tori: from totally un-oriented to semi-oriented (i.e., without an agreement on the orientation in the horizontal or vertical axis, as explained shortly). The authors prove that the black hole search problem cannot be solved in an un-oriented torus using a constant number of agents and tokens if these tokens are unmovable. The authors then consider the use of movable tokens. They prove that the problem is also unsolvable when using any constant number of dispersed agents with 1 movable token each. The authors provide algorithms, each using 5 agents and 3 tokens, for any semi-oriented torus. Finally, they conjecture that at least 5 scattered agents with constant memory, equipped with at least 2 movable tokens, would be able to locate the black hole in a totally un-oriented torus. However, formal proofs of correctness and complexity are not provided and a tight solution remains an open question.

4.2. Solutions under Different Agent Starting Locations

Some researchers [22, 23, 24, 55, 56, 57] choose to study the black hole search using co-located agents, others with dispersed ones [15, 16, 17, 63]. When all agents wake up in the same node, coordination and communication are guaranteed for these co-located agents. This greatly simplifies graph exploration.

4.2.1. Co-located Agents

As previously mentioned, adopting face-to-face communication entails using co-located agents (since dispersed agents may all die in the black hole before ever meeting.) Given it was shown early that, with complete knowledge of the network, 2 co-located synchronous agents are sufficient to locate the black hole, subsequent work [22, 23, 24, 55, 56, 57] has focused on finding solutions that improve the time cost (as reported in Section 4.1.1). Finally, we remark that when using only 2 co-located agents, whether the agents are anonymous or not is irrelevant since an agent can definitely distinguish itself from the other when they meet.

4.2.2. Dispersed Agents

In order to extend the results obtained for 2 co-located synchronous agents, Chalopin *et al.* [15, 16, 17] and Markou *et al.* [63] consider using dispersed agents under the pure token model (as discussed in Section 4.1.2). In these contributions, in contrast to work on co-located agents, the focus is not on time complexity and agent moves but rather on the minimal number of dispersed agents and tokens being used. It is assumed that network size is unknown *a priori* and that agents are restricted to using pure tokens. Thus, given tokens can be placed only at a node, not its ports, coordination between dispersed agents becomes significantly more complex. For example, in a torus, even when an agent sees a token at a node, it still cannot know from which port the previous agent left.

4.3. Solutions under Different Network Knowledge

As previously hinted, network topology may significantly impact on solutions for black hole search. For example, the above-mentioned work on rings [16, 17] and tori [15, 63] clearly shows that, under the same assumptions, more agents and tokens are needed for a torus than a ring. That is, it appears network topology not only affects the complexity of the network, but also the number of agents and the number of tokens necessary and sufficient to solve the black hole search problem. Furthermore we notice that, with a map of an arbitrary network, [56] offers a $3\frac{3}{8}$-approximation algorithm, whereas [24] presents a $\frac{5}{3}$-approximation algorithm that does not use a map but does know the topology is a tree. This strongly suggests that, even without a map, a solution that relies on knowledge of topology may have better time cost than a solution designed for an arbitrary network (viz., *without* knowledge of topology), even if with the help of a map.

Sense of direction is another important consideration. It offers not only consistent edge-labelling, but also a guaranteed method of systematic exploration of the entire graph. (In contrast, without edge-labelling, an agent may not be able to distinguish the edges incident to a node, and thus a whole part of the graph may not be considered during exploration.) Its importance is clearly demonstrated in the results obtained for oriented rings and tori [15, 16, 17] that, under the same assumptions, improve on those for un-oriented rings and tori [16, 17, 63].

Similarly, in [63], Markou *et al.* discuss four levels of network knowledge in a torus, namely: 1) the agents have no agreement on anything regarding the orientation; 2) the agents perceive orthogonal links but they do not agree on which link is horizontal and which is vertical 3) the agents agree on which link is horizontal and which is vertical, but there is no consensus on the orientationof each link; and 4) the agents agree on which link is horizontal and which is vertical and they also agree on the orientation in one of the links. The three latter are called *semi-oriented*. Their results (reported in 4.1.2) demonstrate solutions for oriented tori are less costly than those for semi-oriented tori, thus emphasizing again the important of orientation.

5. Black Hole Search in Asynchronous Networks

In this section, we overview the state of the art for black hole search in an asynchronous network (which is much more complex and relevant in practice than in a synchronous network). Our presentation will follow the different variants given in Figure 2.

5.1. Solutions under Different Communication Models

As previously mentioned, in asynchronous networks, agents may wake up at different times. Agents may never meet each other regardless whether they die in the black hole or not. Thus, face-to-face communication is not of great use to solve the black hole search problem. We will therefore focus on solutions that use a whiteboard or token model.

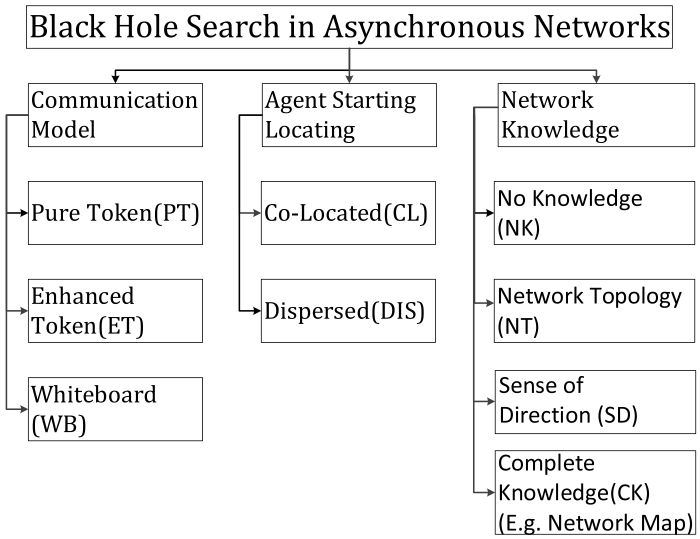

Figure 2: Different Variants for Black Hole Search in Asynchronous Networks

5.1.1. Pure Token Model

Flocchini *et al.* [42] first prove that the pure token model is as powerful as the whiteboard model and that, in an arbitrary network, its complexity is the same as that of the whiteboard model if each of the co-located agents carries a map. They also show that 2 co-located agents, each with 1 token, can locate the black hole in a ring topology using a technique called *ping-pong*. In this specific case, when the network topology is known, the agents can achieve this goal without using a map. They further demonstrate that this ping-pong technique can also be applied to an arbitrary network if a corresponding network map is available to each agent. In the latter case, it costs $\Theta(n \log n)$ moves to locate the black hole. (Additional details are given in [43].)

It is known that $\Delta + 1$ (Recall that Δ is the maximum node degree in the network graph.) agents are necessary to locate the black hole when the topology of an asynchronous network is unknown, regardless of the number of tokens used [50]. With the same number of agents and $O(1)$ tokens in total, it is possible to locate the black hole if each agent has a network map available. Balamohan *et al.* study whether $\Delta + 1$ agents, each with $O(1)$ tokens, can still locate the black hole in an unknown network in [4]. They prove that in order to keep the total number of tokens used to $O(1)$, $\Delta + 1$ agents are not sufficient. They then present a protocol that uses $\Delta + 2$ agents, each carrying 3 tokens, to locate a black in an unknown network.

Finally, we remark that, when using the pure token model for black hole search in an asynchronous network, researchers have exclusively considered co-located agents.

5.1.2. Enhanced Token Model

Due to the limitations of the pure token model, Dobrev *et al.* [28, 37, 39, 40] and Shi *et al.*[66] use the enhanced token model to further improve the move and agent costs. In all these studies, each agent can carry and most importantly can place in the same node more than 1 token at any time. Given these characteristics, Dobrev *et al.* [38, 40] introduce an algorithm to locate the black hole in an un-oriented ring network with dispersed agents. Same as in synchronous networks, coordinating dispersed agents is significantly more complex than using co-located agents. The proposed algorithm demonstrates that using $O(1)$ enhanced tokens is sufficient for successful black hole search in asynchronous networks using dispersed agents. In [37], Dobrev *et al.* demonstrate that the move cost of $O(kn + n \log n)$ of [38, 40] can be reduced to $O(n \log n)$ by using 2 co-located agents with $O(1)$ tokens per agent, when the orientation of the ring is known.

Apart from the ring networks, Shi *et al.* [66] prove that 2 co-located agents, each with $O(1)$ tokens, can locate the black hole in $\Theta(n)$ moves for hypercube, torus and complete networks. (Details are available in [67].) Using dispersed agents, 3 agents and 7 tokens in total are required to locate a black hole within $\Theta(n)$ moves in an oriented torus. When the number of agents increases to $k(k > 3)$ with 1 token per agent, the move cost becomes $O(k^2n^2)$. This result is interesting. It shows that if the number of dispersed agents in a torus increases, the communication between these agents becomes significantly more complicated. This is reflected in the increase of the move cost.

Moreover, for an arbitrary unknown network graph with known n, Dobrev *et al.* [28] present an algorithm using $\Delta + 1$ agents and one token per agent and $O(\Delta^2M^2n^7)$ moves to locate the black hole. Here M is the total number of edges of the graph. This result has been improved by the same authors in [29] to $O(\Delta^2M^2n^5)$ moves. In contrast, under the same assumption in the whiteboard model, the cost of the algorithm is $\Delta + 1$ agents and $\Theta(n^2)$ moves. For arbitrary unknown network graphs, the costs of the enhanced token model are significantly greater than those of the whiteboard model [28]. However, when a network map is available to the agents, the costs of the enhanced token model can be reduced to the same as those for the whiteboard model [37].

5.1.3. Whiteboard Model

In both types of token models, agents can only express very limited messages. This is why the whiteboard model is still the most popular agent communication model and has been studied by many (e.g., [5, 6, 21, 26, 27, 30, 31, 32, 34, 35, 36, 52]).

In addition to presenting solutions to black hole search in asynchronous arbitrary networks [31, 35, 36], Dobrev *et al.* [32] solve a multiple agents rendezvous problem in a ring network that contains a black hole. In their paper, the final goal of the agents is not only to locate the black hole but also to collect all survived dispersed agents in one node. The authors offer a protocol that can rendezvous k agents in $\Theta(n)$ time units. They claim that when k is unknown, this protocol is also a solution to the black hole search problem. In terms of the time complexity in rings, Dobrev *et al.* [30, 34] show that at least $2n-4$ time units are needed in the worst case and give an algorithm, achieving it using $n-1$ co-located agents. (Here movement and exploration are assumed to consume one time unit.) Apart from time complexity, the authors also prove that 2 agents are necessary and sufficient and present an algorithm to locate the black hole in $O(n \log n)$ moves, regardless whether the agents are co-located or dispersed, provided the orientation of the ring is known *a priori*. If the ring is un-oriented, 3 dispersed agents are necessary and sufficient. Apart from rings, Dobrev *et al.* [26] (with additional details in [27]) also present a general strategy to locate the black hole in $O(n)$ moves by using 2 co-located agents for some other common interconnected networks, such as *cube-connected cycles, wrapped butterflies, star graphs, chordal rings, hypercubes, tori of restricted diameter*, and in *multidimensional meshes.*

Based on Dobrev's work, Balamohan *et al.* [5] prove that $3n \log_3 n - O(n)$ moves are necessary in an asynchronous ring when 2 co-located agents are used. As for time complexity, Balamohan*et al.* [6] improve the algorithm of [30] to solve the problem in an average of $\frac{7}{4}n - O(1)$ time units when $n-1$ agents are used (with 2 extra time units required in the worst case). The authors also propose another algorithm to locate the black hole in $\frac{3}{2}n - O(1)$ time units on average, using $2(n-1)$ agents without increasing the time complexity in the worst case.

While all the above studies only consider the case of undirected graphs, Czyzowicz *et al.* [21] study the black hole search in directed graphs. They show that at least 2^d agents are necessary in the worst case, where d is the in-degree of the black hole. If a planar graph with a planar embedding is known to the agents, $2d$ agents are needed, and $2d+1$ agents are sufficient.

5.2. Solutions under Different Agent Starting Locations

As discussed for the synchronous networks, when the homebases of the agents are dispersed, black hole search is more complex than if all agents wake up in the same node. This is even more so for an asynchronous network: given agents may wake up at different times, coordinating them to locate the black hole with minimal resource cost is a challenge. For example, 2 co-located agents suffice to solve the problem in a complete network in $\Theta(n)$ moves in [67]; while using dispersed agents costs $O(n^2)$ moves.

5.2.1. Co-located Agents

The co-located agent model is frequently used in the literature. Many whiteboard based studies adopt this model (e.g., [5, 6, 21, 27, 30, 31, 35, 36, 52]). Similarly, in token-based research, many choose to solve the problem under this model (e.g., [4, 28, 29, 37, 42, 43, 66, 67]). Among these papers, [5, 6, 30, 37, 42, 43] specifically consider ring networks, while [5, 6, 30] instead study time complexity. In particular, [6] offers an algorithm that improves the average time from [30]. Moreover, [37, 42, 43] only use 2 agents, and [37] studies the enhanced token model, while [42, 43] investigate the pure token model.

As previously suggested, when the agents are initially co-located, they can easily establish agreements before any exploration. This can greatly help coordinating agents and eventually reducing the resource costs. For example, in a ring network, when the agents are co-located, the orientation is no longer important. This is because when there are only two directions, the agents can certainly make an agreement at the beginning of the exploration on what direction to take. Furthermore, solving the problem using co-located agents in a ring with n nodes is the same as having each agent carry a network map in asynchronous networks. The situation is different when using dispersed agents. That is, unless the orientation of the ring is known, having a map or not leads to different solutions.

The following example (described in [42, 43]) illustrates how a pair of co-located agents can locate the black hole using such an 'agreement': 2 agents each with one token start to explore the ring using cautious walk; one going right and the other going left. However, only one agent at a time is allowed to explore. To ensure this, one agent must first 'steal' the token from the other before its start its exploration. Stealing is possible because, during cautious walk, an agent has to leave a token before going to the next node. After such a theft, the agent without a token cannot continue exploration and has to go 'back' to look for a token. This is repeated until one agent dies. For example, suppose the right agent goes first. Before the left agent starts, it must first go right and steal the token of the right agent, and then it goes left for exploring. Once the right agent finds its token has gone, it goes left and steals a token from the left agent, and then goes right again. Repeating this process can ensure that only one agent dies in the black hole and that the surviving one knows the location of the dead agent.

5.2.2. Dispersed Agents

Dispersed agents have been adopted by the research based either on the whiteboard model [30, 32, 34, 52] or on the enhanced token model [39, 40, 66, 67]. Furthermore, no one has yet offered solutions to black hole search in asynchronous networks that use dispersed agents carrying pure tokens. The reason for this might be that such a solution is likely to use more pure tokens than one that relies on enhanced tokens.

Both Shi *et al.* [66] and Dobrev *et al.* [30] consider the use co-located agents *and* the use of dispersed ones. More specifically, in [66] authors focus on agent moves in *hypercube, torus*, and *complete networks*, whereas in[30], they measure agent moves and time complexity in ring networks.

Finally, in [39], Dobrev *et al.* solve the black hole search problem using an algorithm called *Pair Elimination* in oriented ring networks. The agents are initially dispersed in the ring and each endowed with $O(1)$ enhanced tokens. This algorithm consists in letting all the agents try to form pairs as soon as they wake up. All paired agents eliminate all the single agents they meet. Each pair has a level. A pair increases its level each time it eliminates another agent. When two pairs meet, the higher level pair always eliminates the lower level pair. Between pairs of the same level, the right pair eliminates the left pair.

Eventually only one pair will survive, and one of the two agents forming that pair will locate the black hole. In contrast to the co-located case (for which each agent carries only 1 pure token), pair elimination requires 4 tokens for each agent even when they use the enhanced token model in the dispersed case. (This stems from the fact that communication/coordination among dispersed agents is significantly more complex than the co-located case.)

5.3. Solutions under Different Network Knowledge

Most existing work on black hole search in asynchronous networks (e.g., [6, 37, 39, 42, 43]) assumes agents have *knowledge of incoming links*, which means that when an agent enters a node, it is 'told' which port it used to do so. In turn, this enables this agent to possibly 'go back' to its previous node. Conversely, Glaus *et al.* [52] study arbitrary, unknown distributed systems without knowledge of incoming links. They present a lower bound on the size of the optimal solution, showing that at least $\frac{d^2+d}{2}+1$ co-located agents are necessary and sufficient to locate the black hole. Here d denotes the number of links leading into the black hole (i.e., the node degree of the black hole).

In an un-oriented network, all ports that lead to a black hole should be marked as dangerous, hence $\Delta+1$ agents are necessary. However, in an oriented network, the number of agents that die in the black hole can be reduced by forcing agents to only enter a node from certain directions. For example, given a torus whose nodes have their ports labelled as *north, south, east*, and *west*, Shi *et al.* [67] assume an agent can only enter a node from the west and come out from the east, or enter from the north and come out from the south. With this assumption, only 3 agents are necessary. In contrast, when agents are allowed to enter a node from all four directions, at least 5 agents are necessary.

Dobrev *et al.* [33] prove that without any knowledge, $\Delta+1$ agents are needed and the cost is $\Theta(n^2)$. However, with a sense of direction but lack of information of the network topology, only 2 agents are required to achieve the same cost. The main idea of that algorithm is as follows: a) the two agents start from the homebase *hb* and construct at *hb* a spanning tree of explored nodes (i.e., those visited by one agent); b) an agent searches this tree and if there is a node with unexplored ports, that agent goes to explore that node in order to make all its ports explored using cautious walk; c) after each such exploration, the agent comes back to *hb* and adds that node to the tree as an explored node. The algorithm depends on a agent leaving navigation instructions (i.e., where it is going) to the other agent, each time the former leaves the homebase. The algorithm terminates when the number of explored nodes reaches $n-1$.

Again we observe that the knowledge of the network topology (e.g., ring, hypercube, torus, complete, tree and arbitrary networks) has great impact on results for black hole search. For example, Balamohan *et al.* [5, 6], Chalopin *et al.* [16] and Dobrev *et al.* [30, 34, 37, 39] propose algorithms based on ring networks. In the same vein, Shi *et al.* [67] design algorithms for hypercube and torus networks with co-located agents, and for torus and complete networks with dispersed agents. Also, Dobrev *et al.* [26, 27]) present a general strategy that allows 2 agents to locate the black hole with $O(n)$ moves in some common interconnected networks. In contrast, [4, 26, 27, 31, 33, 36, 42, 43] search the black hole in arbitrary networks.

As just mentioned, for an arbitrary network, Dobrev *et al.* [31, 33] prove that using the whiteboard model, the black hole search problem can be solved with $\Delta+1$ agents in $\Theta(n^2)$ moves without network maps. Also, recall this result (pertaining to move complexity) can be achieved using only 2 agents provided there is a sense of direction. With complete knowledge of the network, 2 agents are sufficient and the cost can be reduced to $\Theta(n \log n)$. In another paper [35], Dobrev *et al.* present a universal protocol that locates the black hole using at most $O(n + d \log d)$ moves with 2 agents each carrying a network map. Here d is the diameter of the network. Still using 2 agents, the same authors [36] present a strategy that can locate the black hole in $O(\Sigma_{i=1}^{k}|C_i| \log |C_i|)$ moves, here $C = C_1, C_2, ..., C_i..., C_k$ is an open vertex cover by cycles of a 2-connected graph[5].

The point to be grasped is that these results show that having a network map or a sense of direction can significantly reduce the cost complexity in asynchronous networks.

6. Multiple Black Hole Search

As previously mentioned, the only way to locate a black hole in an asynchronous network is to have at least one agent visit all the nodes except the black hole. Therefore, the network minus the black hole has to be connected. Otherwise, the presence of the black hole may partition the network into several disconnected subgraphs, making it impossible to visit all nodes. Also recall that, in synchronous networks, with the help of time-out mechanism, the single black hole search problem can still be solved even if the network is disconnected by the black hole (as is the case for tree networks).

Clearly, the problem becomes much more complex when the network contains multiple black holes. Indeed, in some situations, this problem is unsolvable even in synchronous networks. For example, Figure 3 illustrates a ring network containing 3 black holes that disconnect the ring into 3 sub-graphs. Unless there are enough agents starting at specific nodes, locating these multiple black holes cannot be guaranteed.

Strategies for finding multiple black holes can be intuitively grouped into three categories, each with different assumptions and results, which are discussed next.

6.1. Best Effort without Modifying the Black Hole Search Problem

[5]An *open vertex cover by cycles* (C) is defined as a set of simple cycles such that a) each vertex of G is covered by a cycle from C and b) the connectivity graph of these cycles (where each cycle is represented by a vertex, and 2 vertices are connected if the corresponding cycles share an edge) is connected.

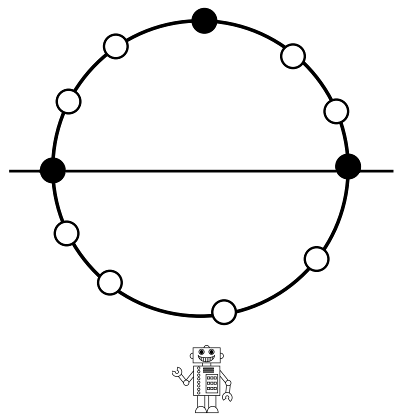

Figure 3: A ring network that is disconnected by multiple black holes (represented as solid circles).

This strategy tries to find as many black holes as possible without modifying the traditional black hole search problem. In a synchronous network, as discussed in Section 2.1, finding out whether there is a single black hole is rather trivial given a time-out mechanism. Should the network be disconnected due to the presence of several black holes, some nodes may never be explored. In this case, finding all the black holes is impossible. Otherwise, Cooper *et al.* [19] offer a solution to finding all possible black holes. First, they study the multiple black hole search problem in synchronous networks using the face-to-face model. They assume that k co-located agents know the topology of the whole network including the size n and number of black holes b. They conclude that any exploration algorithm needs $\Omega(n/k + D_b)$ steps in the worst case to solve a multiple black hole search problem, while D_b is the diameter of the network with at most b nodes deleted. They then provide a general algorithm that performs the exploration in $O(\frac{\frac{n}{k}\log n}{\log\log n} + bD_b)$ steps in an arbitrary network with network maps available to the agents, where $b \leqslant k/2$. In the case where $b \leqslant k/2$, $bD_b = O(\sqrt{n})$ and $k = O(\sqrt{n})$, they give a refined algorithm that performs the exploration in asymptotically optimal $O(n/k)$ steps. Ultimately a node can be identified as a black hole or as a safe node (if and only if it can be reached following a path of safe nodes).

6.2. Variants of the Black Hole Search Problem

In the traditional black hole search problem, the existence of a black hole is persistent. That is, a black hole is not affected by the arrival of any incoming agent. Cooper *et al.* [20] solve a variant of the multiple black hole search problem in synchronous networks by changing this model. They introduce the notion of a *faulty node*, which is a weak form of a black hole. A faulty node is repaired when first visited by an agent (which, however, dies repairing it).

And once repaired, this node will permanently behave as a normal one. Hence, when a network contains more than one faulty node, the agents are still able to explore the whole graph. Also, if more than one agent enters the same faulty node at the same time, only one will repair the faulty node and die while the others can continue their explorations.

The agents used in [20] know the topology of the whole network, move synchronously, use the face-to-face model, and are initially co-located at the same node. Given a network map, the whole network is first divided into equal partitions of size $O(D)$, where D is the diameter of the network. Consequently, an agent should spend $O(D)$ time to explore one such partition. All the agents start from the same homebase and each agent explores a partition. After $O(D)$ time, if an agent returns to the homebase, it is inferred that the partition it explored contains no faulty node. Should an agent not show up on time, it is assumed to be dead in a faulty node (which it repaired) and the partition to which it was assigned is still marked as unsafe and thus in need of further exploration. After one such iteration of exploration, once all surviving agents come back to the homebase, they will start a new iteration of exploration on the remaining unsafe partitions of the network. This process will be repeated until there are no more unsafe partitions. Eventually, this 'faulty node repair' problem can be solved within $O(\frac{n}{k} + \frac{D \log f}{\log \log f})$ time steps, where $f = min(\frac{n}{k}, \frac{n}{D})$, assuming that the number of faulty nodes is at most $k/2$. It must be emphasized that, in [20], because the face-to-face model leaves no mark on the nodes, once an agent dies repairing a node, the other agents cannot know where it died. Therefore, ultimately, all faulty nodes are repaired but their locations remain unknown.

D'Emidio *et al.* [25] study the same problem under the same conditions as [20] with a slight change to one assumption: if more than one agent enters the same faulty node at the same time, all agents die. Trying to make the problem more realistic, the authors however introduce a new behavior: if one agent enters a faulty node u, all agents within distance r from u disappear along with the faulty node. D'Emidio *et al.* first prove that the faulty node repair problem is NP-hard even when $b = k = 1$, where b is the number of faulty nodes and k is the number of agents. Second, when $r = 0$ (which means the agents die only when they physically enter a faulty node), using a simple variation of the algorithm of Cooper *et al.*, the faulty node repair problem can be solved in $\Theta(\frac{n}{k-b} + \frac{D \log f}{\log \log f})$ with $k > b$ always true. Otherwise, all agents will die. Third, for any $r > 0$, the faulty node repair problem requires $\Omega(n)$ time steps in the worst case. Fourth, when $r = 1$, the faulty node repair problem can be solved in $\Theta(n)$ time steps, and the authors provide two strategies to achieve this bound. Finally, the authors report their experimental results to demonstrate correctness.

In a different vein, Flocchini *et al.* [44, 47] solve the multiple black hole search problem via a *subway model* using co-located agents with the whiteboard model, the number b of black holes being known to the agents. The authors use carriers (the subway trains) to transport agents (the passengers) from node to node (the subway stops), and the carriers move asynchronously in a directed graph. When a carrier enters a node, the agents can either get off from the carrier and explore the node, or stay on the carrier to go to another node. In a traditional black hole search, any incoming data will be deleted, including the carrier. However, in this subway model, the black holes no longer affect the carriers and can only eliminate the agents. At the homebase, there is a whiteboard that is used to record all explored, unexplored and dangerous nodes. Initially, all nodes are recorded as unexplored except the homebase. Once an agent chooses to explore a node, the node will be marked as dangerous until the agent comes back and marks it as explored. Eventually, the algorithm terminates when $n - b$ nodes have been explored, the remaining b dangerous nodes being the black holes. In [44, 47], when $k = r + 1$ agents are used (where r is the number of carrier stops at black holes), the number of carrier moves is $O(k \cdot n_C^2 \cdot l_R + n_C \cdot l_R^2)$. Here n_C is the number of subway trains, and l_R is the length of the subway route with the most stops.

Under the same assumption and keeping the same carrier moves as [44, 47], Flocchini *et al.* [46] solve the same problem with dispersed agents. Instead of having a whiteboard at the homebase, these authors put the whiteboard on the carriers. Thus, an agent only has to come back to a carrier to update its exploration information.

6.3. A Simplifying Assumption

As Figure 3 suggests, if the presence of multiple black holes results in the network being effectively partitioned by the latter into several disconnected partitions, then it impossible to visit all nodes without going through a black hole. In order to alleviate this difficulty, some researchers make the assumption that the network minus the black holes is connected.

For example, Flocchini *et al.* explicitly state this simplifying assumption in [45]: "after deleting all the black holes, the network still remains interconnected". (Clearly, without this assumption, it is impossible to locate all the black holes in any given network.) However, these authors also complicate multiple black hole search by adding link failure. That is, in their model, an link failure is locally detectable at an adjacent node. More specifically: 1). an edge is identified by its port number in its incident node and 2). if information about an edge is written on a whiteboard, an agent can notice the absence of an edge with such a port number. If no information about an edge is written (i.e. this edge has disappeared before any agent has visited), it is treated likely it has never existed. It is assumed that any such failure occurs only when no agent is traversing that link, and that the failures do not disconnect the safe part of the network (otherwise dangerous graph exploration is clearly unsolvable). Under this assumption, the authors present an algorithm to solve dangerous graph exploration with link deletions in an arbitrary unknown graph with asynchronous dispersed agents using the whiteboard model. The algorithm can correctly solve the link deletion problem within finite time by marking all safe edges as such, and marking as dangerous every port that is on a safe node leading to a black hole or to a faulty edge (i.e., an edge that has failed). The total number of moves performed by the agents is at most $O(k^2 \cdot n_s + n_s \cdot m + k \cdot n_s \cdot D)$, where k is the number of agents, n_s is the number of safe nodes, and m is the number of edges or links.

Kosowski *et al.* [58, 59] also assume that the graph is strongly connected if all black holes are removed. They find out that $O(d \cdot 2^d)$ co-located agents are sufficient to solve the black hole search problem in a directed graph with an arbitrarily large n, where the network is synchronous and d is the number of edges leading to the black holes. Furthermore, the authors show that when $d = 2$, 4 agents are always sufficient in synchronous networks. However, in asynchronous networks, at least 5 agents are required when $d = 2$. Finally, when $d = 1$, 2 agents are always sufficient and sometimes required in both synchronous and asynchronous networks.

7. Other Types of Malicious Hosts

Beyond studying the traditional black hole and its variants (e.g., a) the repairable black holes introduced in [20] by Cooper *et al.* and b) the new subway model presented by Flocchini *et al.* in [44, 47]), work on other types of malicious hosts exists and is briefly discussed here.

Chalopin *et al.* [18] study a rendezvous of mobile agents in a network with *faulty links*. In that model, some of the edges in the graph are dangerous for the agents: any agent that attempts to traverse such an edge (from either direction) simply disappears, without leaving any trace. Notice that if all the edges incident to a node u are faulty, then node u can never be reached by any agent and thus is essentially equivalent to a black hole.

Královič *et al.* [60] study a *periodic data retrieval problem*, which is equivalent to a) *periodic exploration* in fault-free networks and b) a black hole location problem for which there is only one black hole in the network. The aim of the periodic data retrieval problem is to deliver data from any non-faulty node to the homebase (a node that collects all data) periodically. These authors address a ring network that contains one malicious host that can behave in an 'arbitrary' way (except for the fact that it cannot change the internal state (i.e., the contents of the local variables) of an agent, nor create an agent with a given state).

Luccio *et al.* [62] consider a mobile agents rendezvous problem in spite of a malicious agent. This is similar to [32], which rendezvouses agents in a ring in spite of a black hole. The agents that need to be gathered are called 'honest agents' and their communication model is similar to (albeit much more limited than) the face-to-face one. More specifically, an honest agent can see other honest agents and read their states if they are in the same node, but such agents cannot exchange messages or leave any message on the nodes. Also, while most studies that use the face-to-face model always assume the networks are synchronous (in order to enforce the meeting of agents), Luccio *et al.* instead use an asynchronous network. Finally, we remark that the problem they address is solvable likely due to the fact that, in their work, a malicious agent can only

a) block honest agents from visiting its current node and b) move in the network at arbitrary speed but without deleting honest agents.

Cai *et al.* [13, 14] consider the problem of a *black virus* that, like a black hole, deletes any incoming agent. But, unlike a black hole (which is defined as a static host), a black virus moves from node to node, thus potentially increasing the number of dangerous nodes. Furthermore, unlike a black hole (which can only be located but not removed), a black virus is destroyed if it enters a node that contains an anti-viral system agent. Thus, the only way to remove a black virus is to surround it by anti-viral system agents and force it to move to a neighbouring node that already contains at least one anti-viral system agent. In the same vein, some theoretical work has focused on the *intruder capture* problem (also known as *graph decontamination*): an intruder (a harmful agent) moves through the network infecting nodes and the goal is to remove the intruder from the network using mobile agents. Unlike a black virus, an intruder can only harm nodes, not agents. This problem has been extensively studied in [9, 12, 48].

Finally, *black hole attack* [2, 8, 68] is also a research topic remotely related to black hole search. Most importantly, the networks considered for black hole attack are different from those of black hole search: In the latter, the networks are static, while in the former, the networks can be dynamic (e.g., MANET (Mobile Ad-Hoc networks), wireless networks, mobile networks). For example, in MANET, the network topology is only formed once one node needs to send a data package. Khari *et al.* [54] survey security attacks, as well as secured routing protocols in MANET, and offer a definition for black hole attack. Moreover, their survey mentions a variation of black hole attack called *grey hole attack* [1, 7, 65]: whereas a black hole will delete any incoming data packages, a grey hole only deletes part of the packages.

8. Further Analysis and Future work

In this section, we analyze all the reviewed studies and highlight some future work for:

- single black hole search in both synchronous network (Section 8.1.1) and asynchronous network (Section 8.1.2) and
- multiple black hole search (Section 8.2) and
- black hole search using different types of agents (Section 8.3).

8.1. Further Analysis and Future work for Single Black Hole Search

In this subsection, we list all possible combinations of different assumptions and organize all the single black hole search studies under each such combination. Our findings are presented in Tables 3 and 4. We then consider the remaining combinations/open problems not yet studied and identify some possibilities for future research.

Table 3: Existing work on black hole search in synchronous networks.

Combination	Communication model	Agent starting location	Network knowledge	Paper
1	PT	DIS	Unknown n, oriented torus	[15]
2	PT	DIS	Unknown n, un-oriented torus	[63]
3	FTF + PT	DIS	Unknown n, oriented or un-oriented ring	[16, 17]
4	FTF	CL	tree	[22, 24]
5	FTF	CL	Complete Knowledge	[23, 55, 56, 57]

PT: pure token; FTF: face-to-face; CL: co-located; DIS: dispersed

8.1.1. Single Black Hole Search in Synchronous Networks

Single black hole search in synchronous networks is not studied as often as in asynchronous networks, the latter being a more realistic model. As shown in Table 3, the pure token model is only used with dispersed agents (Combinations $1-3$ in Table 3). These 4 papers only study the minimal number of agents and tokens required to solve the black hole search problem without offering any specific algorithm or complexity analysis. Hence, characterizing the number of agent moves, as well as the time cost, can be further studied. Also, only rings and tori have been studied under the pure token model with a focus on the number of agents and tokens used. Studying the problem in other topologies (such as hypercube or mesh) is required. Additionally, we have observed that, for black hole search in asynchronous networks, using co-located agents always costs fewer moves than using dispersed agents. Studying the use co-located agents with the pure token model in a synchronous network should further support this observation. Moreover, it has been proven that, in asynchronous arbitrary networks, the pure token model can offer the same complexity as the whiteboard model provided a network map is available. Whether this is also true for synchronous networks also needs to be studied.

Finally, one of the several advantages of solving the black hole search problem in synchronous networks versus in asynchronous networks is the possibility of using face-to-face communication. Indeed, a hybrid that combines this model with the use of whiteboards or tokens appears to lead to further reduction on both time costs and agent moves. However, in the face-to-face model, the agents leave no marks on nodes. Consequently, as previously mentioned, if using dispersed agents, it is possible all these agents could die in the black hole before they even meet with each other. Therefore, we repeat, it is of very little interest to consider dispersed agents when relying on face-to-face communication. Conversely, finding a time-optimal algorithm for an arbitrary unknown graph using co-located agents and only the face-to-face communication should be investigated.

8.1.2. Single Black Hole Search in Asynchronous Networks

We do not include edge-labelling in our discussion because it is widely adopted in the field. Also, since network size must be known *a priori* in asynchronous networks, we do not further mention n in this subsection. Also recall that, when using co-located agents to explore a ring, whether or not the ring is oriented does not affect the move cost of the algorithm. Therefore, we do not discuss separately each of these two possibilities below. Finally, as ring is a special topology, namely the sparsest bi-connected graph, we list it separately in Table 4.

We have seen that Glaus *et al.* [52] study the black hole search problem without the knowledge of incoming links in an unknown un-oriented arbitrary network. Under these assumptions, Glaus *et al.* solve the problem when both the agents and the network nodes have distinct IDs. Whether, under the same assumptions, the black hole search problem is still solvable if an anonymous network and anonymous agents are used, remains an open problem. Solving the black hole search problem without the knowledge of incoming links in an unknown un-oriented arbitrary network also remains an open problem if a) a synchronous network and/or b) tokens and/or c) dispersed agents are used (in lieu of the asynchronous network with whiteboard and co-located agents of [52]).

Balamohan *et al.* identify another open problem in [6]: Is there an algorithm that locates the black hole in $\frac{3}{2}n - O(1)$ time (average case) and $2(n-1)$ time (the worst case) using $n-1$ co-located agents and the whiteboard model?

Dobrev *et al.* consider a very difficult condition, namely, no topology knowledge is assumed in [29] (Combination 8 in Table 4). Their algorithm locates the black hole in $O(\Delta^2 M^2 n^5)$, using the enhanced token model with $\Delta+1$ co-located agents. Under the same conditions, solving the problem in the whiteboard model only costs $\Theta(n^2)$ moves. The question these results raise is whether we can find a solution at a lower move cost for the problem using the enhanced token model without any topology knowledge. Relaxing the assumption regarding knowledge of topology (Combinations $9-10$ in Table 4) may lead to further reductions of the move cost when trying to solve this specific problem.

The fact that only ring, hypercube, torus, and complete network have been studied under the enhanced token model suggests yet another research direction, namely the consideration of other topologies under the same assumptions.

Finally, with respect to the use of the pure token model, whether or not the number of moves can be further reduced by increasing the number of agents and/or by knowing the topology at hand is an open question. Other open problems pertaining to this model include: a) whether black hole search is solvable in an arbitrary unknown graph and b) whether dispersed agents can still solve the problem (over co-located agents, which have been the only ones used with this model thus far).

Table 4: Existing work on black hole search in asynchronous networks.

Combination	Communication model	Agent starting location	Network knowledge	Paper
1	WB	CL	No knowledge	[21, 31, 33, 52]
2	WB	CL	NT	[26, 27]
3	WB	CL	SD and no NT	[31, 33]
4	WB	CL	ring	[5, 6, 30, 34]
5	WB	CL	Complete Knowledge	[26, 27, 31, 33, 35, 36]
6	WB	DIS	un-oriented ring	[30, 32, 34]
7	WB	DIS	oriented ring	[30, 32, 34]
8	ET	CL	No Knowledge	[28, 29]
9	ET	CL	oriented ring	[37]
10	ET	CL	SD and NT	[66, 67]
11	ET	DIS	SD and NT	[66, 67]
12	ET	DIS	oriented ring	[39]
13	ET	DIS	un-oriented ring	[38, 40]
14	PT	CL	ring	[42, 43]
15	PT	CL	Complete Knowledge	[42, 43]
16	PT	CL	No Knowledge	[4]

WB: whiteboard; ET: enhanced token; SD: sense of direction; NT: network topology; CL: co-located; DIS: dispersed

8.2. Future Work for Multiple Black Hole Search

In the context of the multiple black hole search problem, recall that that Cooper *et al.* [20] and D'Emidio *et al.* [25] solve a weaker form of this problem in which an agent can repair a black hole by sacrificing itself. In these two papers, the authors assume that each agent carries a map of the synchronous network at hand. One question then is how to deal with repairable faulty nodes if agents have partial or no *a priori* knowledge of the topology of the network. Also, this problem should be investigated for asynchronous networks, with different types of communication models (namely, whiteboard, as well as pure and enhanced tokens). Also recall that D'Emidio *et al.* [25] assume that if an agent enters and repairs a black hole, all agents within a distance r of the black hole will disappear along with it. The authors only solve the problem when $r = 0$ and $r = 1$. Clearly, more work is required for cases where $r > 1$.

As previously mentioned, Flocchini *et al.* [47] solve the multiple black hole search problem with a *subway model* in an asynchronous network using the whiteboard model and co-located agents. These initial results suggest several questions to research such as a) whether a different communication model can be used, b) whether a solution for a synchronous network can be obtained (especially given that it is assumed that a carrier takes the same amount of time to travel between two stations) and c) how such new solutions would compare in costs with the solution put forth by these authors. In particular, with respect to choice of communication model, it may be interesting to assume that no whiteboard is available and that a cell phone network does not work in subways. Consequently, a new model to study would have agents only communicate with each other when they meet in the same subway station or in the same carrier. Alternatively, agents could use 'walkie-talkies' that support two-way communication over short distances. Both of these models-to-study are essentially variants of the face-to-face model. Finally, recall that Flocchini *et al.* a) assume the network is a directed graph and b) distinguish between carrier moves and agent moves. This prompts asking whether or not the use of an undirected graph may help reducing a) the carrier move complexity and/or b) the total move cost of the agents.

8.3. On Limiting the Memory of Agents

Many solutions presented in this survey rest on the use of agents endowed with unlimited memory so that they can carry a network map or build such a map during the network exploration. In reality, however, the memory of a mobile agent is constrained. This observation opens the door to many variations for such a constraint. For example, Flocchini *et al.* [41] introduce agents with *very* limited memory. More generally, an agent is *oblivious* if all the information it holds is cleared at the end of each computing cycle. In essence, such agents are memoryless, that is, have no memory of any past actions and computations: their current behavior can only be based on what has been determined in the current computing cycle. The consequences of using such agents (or agents with similar constrained behavior) remain to be explored. In particular, can the 'absence' of memory in such agents be compensated by the use of a whiteboard or tokens?

9. Conclusions

In this paper, we first introduced the main models and assumptions that are commonly used in the literature on black hole search, namely with respect to a) network synchronization, b) the communication model between agents, c) their starting locations and d) the topological knowledge each may hold. Our goal in doing so was to capture the variability found across the space of existing solutions for black hole search. We then summarized the typical measures of complexity and evaluation approaches found in the relevant literature. For presentation purposes, existing work was organized into two categories: solutions for synchronous networks and solutions for asynchronous ones. In each of these two categories, we contrasted the proposed solutions with respect to the above-mentioned set of four 'vectors' of variability. More specifically, we tried to understand the impact of each assumption (viewed as a choice in one of these four vectors of variability) of a proposed solution on the cost complexity of that solution. We then briefly touched on a) approaches to the multiple black hole search problem and b) recent results pertaining to the different types of malicious hosts. We concluded with two tables summarizing the work we have reviewed, as well as a partial list of open problems stemming from this survey.

10. Acknowledgments

The authors gratefully acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. 371977-2009 RGPIN. Additionally, the authors also would like to thank several anonymous reviewers who gave insightful comments that have helped to improve the quality of the manuscript.

References

[1] Agrawal, P., Ghosh, R. K., Das, S. K., 2008. Cooperative black and gray hole attacks in mobile ad hoc networks. In: Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication. ICUIMC '08. ACM, New York, NY, USA, pp. 310–314. URL http://doi.acm.org/10.1145/1352793.1352859

[2] Al-Shurman, M., Yoo, S.-M., Park, S., 2004. Black hole attack in mobile ad hoc networks. In: Proceedings of the 42nd annual Southeast regional conference. ACM, pp. 96–97.

[3] Albers, S., Henzinger, M. R., 2000. Exploring unknown environments. SIAM Journal on Computing 29 (4), 1164–1188.

[4] Balamohan, B., Dobrev, S., Flocchini, P., Santoro, N., 2012. Asynchronous exploration of an unknown anonymous dangerous graph with o (1) pebbles. In: Proceedings of the 19th International Conference on Structural Information and Communication Complexity. SIROCCO'12. Springer, pp. 279–290.

[5] Balamohan, B., Flocchini, P., Miri, A., Santoro, N., 2011. Improving the optimal bounds for black hole search in rings. In: Proceedings of the 18th International Conference on Structural Information and Communication Complexity. SIROCCO'11. Springer, pp. 198–209.

[6] Balamohan, B., Flocchini, P., Miri, A., Santoro, N., 2011. Time optimal algorithms for black hole search in rings. Discrete Mathematics, Algorithms and Applications 3 (04), 457–471.

[7] Banerjee, S., 2008. Detection/removal of cooperative black and gray hole attack in mobile ad-hoc networks. In: proceedings of the world congress on engineering and computer science. pp. 22–24.

[8] Banerjee, S., Sardar, M., Majumder, K., 2014. Aodv based black-hole attack mitigation in manet. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013. Springer, pp. 345–352.

[9] Barrière, L., Flocchini, P., Fraigniaud, P., Santoro, N., 2002. Capture of an intruder by mobile agents. In: Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures. ACM, pp. 200–209.

[10] Bender, M. A., Fernández, A., Ron, D., Sahai, A., Vadhan, S., 1998. The power of a pebble: Exploring and mapping directed graphs. In: Proceedings of the thirtieth annual ACM symposium on Theory of computing. ACM, pp. 269–278.

[11] Bender, M. A., Slonim, D. K., 1994. The power of team exploration: Two robots can learn unlabeled directed graphs. In: Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on. IEEE, pp. 75–85.

[12] Blin, L., Fraigniaud, P., Nisse, N., Vial, S., 2006. Distributed chasing of network intruders. In: Structural Information and Communication Complexity. Springer, pp. 70–84.

[13] Cai, J., Flocchini, P., Santoro, N., 2013. Network decontamination from a black virus. In: Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International. IEEE, pp. 696–705.

[14] Cai, J., Flocchini, P., Santoro, N., 2014. Decontaminating a network from a black virus. International Journal of Networking and Computing 4 (1), 151–173.

[15] Chalopin, J., Das, S., Labourel, A., Markou, E., 2011. Black hole search with finite automata scattered in a synchronous torus. In: Proceedings of the 25th International Conference on Distributed Computing. DISC'11. Springer-Verlag, Berlin, Heidelberg, pp. 432–446.

[16] Chalopin, J., Das, S., Labourel, A., Markou, E., 2011. Tight bounds for scattered black hole search in a ring. In: Proceedings of the 18th International Conference on Structural Information and Communication Complexity. SIROCCO'11. Springer-Verlag, Berlin, Heidelberg, pp. 186–197.

[17] Chalopin, J., Das, S., Labourel, A., Markou, E., 2013. Tight bounds for black hole search with scattered agents in synchronous rings. Theoretical Computer Science.

[18] Chalopin, J., Das, S., Santoro, N., 2007. Rendezvous of mobile agents in unknown graphs with faulty links. In: Proceedings of the 21st International Conference on Distributed Computing. DISC'07. Springer-Verlag, Berlin, Heidelberg, pp. 108–122.

[19] Cooper, C., Klasing, R., Radzik, T., 2006. Searching for black-hole faults in a network using multiple agents. In: Proceedings of the 10th International Conference on Principles of Distributed Systems. OPODIS'06. Springer-Verlag, Berlin, Heidelberg, pp. 320–332.

[20] Cooper, C., Klasing, R., Radzik, T., Mar. 2010. Locating and repairing faults in a network with mobile agents. Theoretical Computer Science 411 (14-15), 1638–1647.

[21] Czyzowicz, J., Dobrev, S., Královič, R., Miklík, S., Pardubská, D., 2010. Black hole search in directed graphs. In: Proceedings of the 16th International Conference on Structural Information and Communication Complexity. SIROCCO'09. Springer, pp. 182–194.

[22] Czyzowicz, J., Kowalski, D., Markou, E., Pelc, A., 2005. Searching for a black hole in tree networks. In: Proceedings of the 8th International Conference on Principles of Distributed Systems. OPODIS'04. Springer-Verlag, Berlin, Heidelberg, pp. 67–80.

[23] Czyzowicz, J., Kowalski, D., Markou, E., Pelc, A., Aug. 2006. Complexity of searching for a black hole. Fundamenta Informaticae 71 (2,3), 229–242.

[24] Czyzowicz, J., Kowalski, D., Markou, E., Pelc, A., Jul. 2007. Searching for a black hole in synchronous tree networks. Combinatorics, Probability & Computing 16 (4), 595–619.

[25] D'Emidio, M., Frigioni, D., Navarra, A., 2013. Exploring and making safe dangerous networks using mobile entities. In: Ad-hoc, Mobile, and Wireless Network. Springer, pp. 136–147.

[26] Dobrev, S., Flocchini, P., Královič, R., Prencipe, G., Ruzicka, P., Santoro, N., 2002. Black hole search by mobile agents in hypercubes and related networks. In: OPODIS. Vol. 3. pp. 169–180.

[27] Dobrev, S., Flocchini, P., Královič, R., Ružička, P., Prencipe, G., Santoro, N., 2006. Black hole search in common interconnection networks. Networks 47 (2), 61–71.

[28] Dobrev, S., Flocchini, P., Královič, R., Santoro, N., 2006. Exploring an unknown graph to locate a black hole using tokens. In: Fourth IFIP International Conference on Theoretical Computer Science-TCS 2006. Springer, pp. 131–150.

[29] Dobrev, S., Flocchini, P., Královič, R., Santoro, N., 2013. Exploring an unknown dangerous graph using tokens. Theoretical Computer Science 472, 28–45.

[30] Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N., 2001. Mobile search for a black hole in an anonymous ring. In: Proceedings of the 15th International Conference on Distributed Computing. DISC '01. Springer-Verlag, London, UK, UK, pp. 166–179.

[31] Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N., 2002. Searching for a black hole in arbitrary networks: Optimal mobile agent protocols. In: Proceedings of the Twenty-first Annual Symposium on Principles of Distributed Computing. PODC '02. ACM, New York, NY, USA, pp. 153–162.

[32] Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N., 2004. Multiple agents rendezvous in a ring in spite of a black hole. In: Proceedings of the 7th International Conference on Principles of Distributed Systems. Springer, pp. 34–46.

[33] Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N., 2006. Searching for a black hole in arbitrary networks: Optimal mobile agents protocols. Distributed Computing 19 (1), 1–18.

[34] Dobrev, S., Flocchini, P., Prencipe, G., Santoro, N., Mar. 2007. Mobile search for a black hole in an anonymous ring. Algorithmica 48 (1), 67–90.

[35] Dobrev, S., Flocchini, P., Santoro, N., 2004. Improved bounds for optimal black hole search with a network map. In: Proceedings of the 11th International Colloquium on Structural Information and Communication Complexity. Springer, pp. 111–122.

[36] Dobrev, S., Flocchini, P., Santoro, N., 2006. Cycling through a dangerous network: a simple efficient strategy for black hole search. In: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems. ICDCS '06. IEEE Computer Society, Washington, DC, USA, pp. 57–57.

[37] Dobrev, S., Královič, R., Santoro, N., Shi, W., 2006. Black hole search in asynchronous rings using tokens. In: The 6th Conference on Algorithms and Complexity (CIAC '06). Springer, pp. 139–150.

[38] Dobrev, S., Santoro, N., Shi, W., 2007. Locating a black hole in an un-oriented ring using tokens: The case of scattered agents. In: The 13th International Euro-Par Conference European Conference on Parallel and Distributed Computing (Euro-Par 2007). Springer, pp. 608–617.

[39] Dobrev, S., Santoro, N., Shi, W., 2007. Scattered black hole search in an oriented ring using tokens. In: IEEE International Parallel and Distributed Processing Symposium (IPDPS-APDCM 2007). IEEE, pp. 1–8.

[40] Dobrev, S., Santoro, N., Shi, W., 2008. Using scattered mobile agents to locate a black hole in an un-oriented ring with tokens. International Journal of Foundations of Computer Science 19 (06), 1355–1372.

[41] Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N., 2007. Computing without communicating: Ring exploration by asynchronous oblivious robots. In: Proceedings of the 11th International Conference on Principles of Distributed Systems. OPODIS'07. Springer-Verlag, Berlin, Heidelberg, pp. 105–118.

[42] Flocchini, P., Ilcinkas, D., Santoro, N., 2008. Ping pong in dangerous graphs: Optimal black hole search with pure tokens. In: Proceedings of the 22nd International Symposium on Distributed Computing. Springer, pp. 227–241.

[43] Flocchini, P., Ilcinkas, D., Santoro, N., 2012. Ping pong in dangerous graphs: Optimal black hole search with pebbles. Algorithmica 62 (3-4), 1006–1033.

[44] Flocchini, P., Kellett, M., Mason, P. C., Santoro, N., 2010. Mapping an unfriendly subway system. In: Proceedings of the 5th International Conference on Fun with Algorithms. Springer, pp. 190–201.

[45] Flocchini, P., Kellett, M., Mason, P. C., Santoro, N., 2012. Fault-tolerant exploration of an unknown dangerous graph by scattered agents. In: Proceedings of the 14th International Conference on Stabilization, Safety, and Security of Distributed Systems. SSS'12. Springer, pp. 299–313.

[46] Flocchini, P., Kellett, M., Mason, P. C., Santoro, N., 2012. Finding good coffee in paris. In: Fun with Algorithms. Springer, pp. 154–165.

[47] Flocchini, P., Kellett, M., Mason, P. C., Santoro, N., 2012. Searching for black holes in subways. Theory of Computing Systems 50 (1), 158–184.

[48] Flocchini, P., Luccio, F. L., Song, L. X., 2005. Size optimal strategies for capturing an intruder in mesh networks. In: Communications in Computing. pp. 200–206.

[49] Flocchini, P., Mans, B., Santoro, N., 2003. Sense of direction in distributed computing. Theoretical Computer Science 291 (1), 29–53.

[50] Flocchini, P., Santoro, N., 2006. Distributed security algorithms by mobile agents. In: Distributed Computing and Networking. Springer, pp. 1–14.

[51] Fraigniaud, P., Ilcinkas, D., 2004. Digraphs exploration with little memory. In: STACS 2004. Springer, pp. 246–257.

[52] Glaus, P., 2009. Locating a black hole without the knowledge of incoming link. In: Algorithmic Aspects of Wireless Sensor Networks. Springer, pp. 128–138.

[53] Greenberg, M. S., Byington, J. C., Harper, D. G., 1998. Mobile agents and security. Communications Magazine, IEEE 36 (7), 76–85.

[54] Khari, M., et al., 2012. Mobile ad hoc netwoks security attacks and secured routing protocols: A survey. In: Advances in Computer Science and Information Technology. Networks and Communications. Springer, pp. 119–124.

[55] Klasing, R., Markou, E., Radzik, T., Sarracco, F., 2005. Hardness and approximation results for black hole search in arbitrary graphs. In: Proceedings of the 12th International Conference on Structural Information and Communication Complexity. SIROCCO'05. Springer, pp. 200–215.

[56] Klasing, R., Markou, E., Radzik, T., Sarracco, F., 2007. Hardness and approximation results for black hole search in arbitrary networks. Theoretical Computer Science 384 (2), 201–221.

[57] Klasing, R., Markou, E., Radzik, T., Sarracco, F., 2008. Approximation bounds for black hole search problems. Networks 52 (4), 216–226.

[58] Kosowski, A., Navarra, A., Pinotti, C. M., 2009. Synchronization helps robots to detect black holes in directed graphs. In: Proceedings of the 13th International Conference on Principles of Distributed Systems. Springer, pp. 86–98.

[59] Kosowski, A., Navarra, A., Pinotti, C. M., 2011. Synchronous black hole search in directed graphs. Theoretical Computer Science 412 (41), 5752–5759.

[60] Královič, R., Miklík, S., 2010. Periodic data retrieval problem in rings containing a malicious host. In: Proceedings of the 17th International Conference on Structural Information and Communication Complexity. SIROCCO'10. Springer, pp. 157–167.

[61] Lange, D. B., Oshima, M., Mar. 1999. Seven good reasons for mobile agents. Commun. ACM 42 (3), 88–89.
URL `http://doi.acm.org/10.1145/295685.298136`

[62] Luccio, F. L., Markou, E., 2014. Mobile agents rendezvous in spite of a malicious agent. arXiv preprint arXiv:1410.4772.

[63] Markou, E., Paquette, M., 2012. Black hole search and exploration in unoriented tori with synchronous scattered finite automata. In: Proceedings of the 16th International Conference on Principles of Distributed Systems. Springer, pp. 239–253.

[64] Markou, E., et al., 2012. Identifying hostile nodes in networks using mobile agents. Bulletin of the EATCS (108), 93–129.

[65] Sen, J., Chandra, M. G., Harihara, S., Reddy, H., Balamuralidhar, P., 2007. A mechanism for detection of gray hole attack in mobile ad hoc networks. In: Information, Communications & Signal Processing, 2007 6th International Conference on. IEEE, pp. 1–5.

[66] Shi, W., 2009. Black hole search with tokens in interconnected networks. In: The 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2009). Springer, pp. 670–682.

[67] Shi, W., Garcia-Alfaro, J., Corriveau, J.-P., 2014. Searching for a black hole in interconnected networks using mobile agents and tokens. Journal of Parallel and Distributed Computing 74 (1), 1945–1958.

[68] Su, M.-Y., 2011. Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. Computer Communications 34 (1), 107–117.

[69] Zarrad, A., Daadaa, Y., 2013. A review of computation solutions by mobile agents in an unsafe environment. International Journal of Advanced Computer Science & Applications 4 (4).