Approximation bounds for Black Hole Search problems[⋆]

Ralf Klasing[⋆⋆], Euripides Markou[⋆⋆⋆], Tomasz Radzik[†], Fabiano Sarracco[‡]

Abstract. A black hole is a highly harmful stationary process residing in a node of a network and destroying all mobile agents visiting the node without leaving any trace. The Black Hole Search is the task of locating all black holes in a network, through the exploration of its nodes by a set of mobile agents. In this paper we consider the problem of designing the fastest Black Hole Search, given the map of the network, the starting node and, possibly, a subset of nodes of the network initially known to be safe. We study the version of this problem that assumes that there is at most one black hole in the network and there are two agents, which move in synchronized steps. We prove that this problem is not polynomial-time approximable within $\frac{389}{388}$ (unless **P**=**NP**). We give a 6-approximation algorithm, thus improving on the 9.3-approximation algorithm from [2]. We also prove **APX**-hardness for a restricted version of the problem, in which only the starting node is initially known to be safe.

Keywords: approximation algorithm, black hole search, graph exploration, mobile agent, inapproximability.

1 Introduction

The Background and the Problem. The problem of protecting mobile agents from malicious hosts, i.e., nodes of a network which store harmful processes in them, has been widely studied ([8, 9, 11, 12]). Even though various countermeasures have been proposed, the general belief (see [8, 13]) is that it is very hard (when not virtually impossible) to fully protect mobile agents from malicious hosts attacks.

⋆ Research supported in part by the European projects IST FET AEOLUS and COST Action TIST 293 (GRAAL), the Royal Society Grant ESEP 16244, and the ACI Masses de données. Part of this work was done while T. Radzik and F. Sarracco were visiting the LaBRI (Laboratoire Bordelais de Recherche en Informatique) in Bordeaux.

⋆⋆ LaBRI - Université Bordeaux 1, 351 cours de la Libération, 33405 Talence cedex, (France). email `klasing@labri.fr`

⋆⋆⋆ Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, email `emarkou@softlab.ece.ntua.gr`

† Department of Computer Science, King's College London, London, WC2R 2LS, UK, email `Tomasz.Radzik@kcl.ac.uk`

‡ Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", email `Fabiano.Sarracco@dis.uniroma1.it`

We consider here malicious hosts of a particularly harmful nature, called *black holes* [2–6]. A black hole is a node in a network which contains a stationary process destroying all mobile agents visiting this node, without leaving any trace. Since agents cannot prevent being annihilated once they visit a black hole, the only way of protection against such processes is identifying the hostile nodes and avoiding further visiting them. In order to locate a black hole, at least one agent must visit it. In the model we considered, the agents communicate only when they are in the same node (and not, e.g., by leaving messages at nodes). Therefore, the black hole can be identified by scheduling a meeting between the agents after any visit of an unknown node. If such node is a black hole, then the agent which visits that node gets destroyed and cannot turn up at a node where the other agents expect it. This allows the surviving agents to infer the existence and location of a black hole.

In this paper we investigate the case in which there are exactly two agents, starting from the same node s, to which at least one agent has to report back the exact locations of the black holes. We assume that there is at most one black hole in the network. We consider the problem of designing a black hole search scheme for a given network, a given starting node s, and a given subset $S \supseteq \{s\}$ of nodes which are initially known to be safe. The black hole, if present, is at any node not in S. It is interesting to observe that the assumption of having at most one black hole in the network does not make the algorithm presented here unsuitable for the general case. A (single black hole) search can be restarted for each new black hole found, on the network obtained by removing all the black holes already found and by inserting into S the nodes already explored. This can be iterated until all the network nodes become explored. Obviously, even if at most two agents can simultaneously coexist in the network, the total number of agents needed is related to the total number of black holes in the network.

The issue of efficient black hole search was extensively studied in [4–6] under the scenario of totally asynchronous networks, i.e., while every edge traversal by a mobile agent requires finite time, there is no upper bound on this time. To solve the problem in this setting, the network must be 2-connected. Moreover, in an asynchronous network it is impossible to answer the question of whether a black hole actually exists, hence it is assumed in [4–6] that there is exactly one black hole and the task is to locate it. Due to the asynchronous setting, it is not possible to provide a simple and easy to compute measure of the time needed by the agents to find the black hole. Hence, the complexity measure taken into account for the algorithms is the total number of moves performed by the agents. In the general case, the authors show that $\Theta(n \log n)$ moves are necessary and sufficient.

In this paper we study the problem under the scenario of synchronous networks, previously considered in [2, 3, 10]. In this scenario it is possible to fix the time needed by an agent for traversing any edge. This assumption makes dramatic changes to the problem. First, the black hole can be located by two agents in any network and the agents can decide if there is a black hole or not. Moreover, it is possible in this case to compute exactly the time needed by the agents to find the black hole. With respect to the total number of moves, this is a more relevant measure in the cases in which there is no cost associated with each agent's traversal, but the target is to determine as quickly as possible the location of the black hole. In order to measure the efficiency of a black hole search, we assume that each agent takes exactly one time unit (one synchronized step) to traverse one edge (and to make all necessary computations associated with this move). Then the cost of a given black hole search (scheme) is defined as the total number of time units the search takes under the worst-case location of the black hole in the network, or when the network contains no black hole.

Previous Results. In [2] the authors prove that the Black Hole Search problem is **NP**-hard, and show a 9.3-approximation algorithm. The restricted case of this problem, when the starting node is the only node initially known to be safe ($S = \{s\}$), is considered in [3] and [10]. In [10] the authors prove that this restricted case is also **NP**-hard, and give a $\frac{7}{2}$-approximation algorithm. In [3] the problem is studied in tree topologies, and the main results are an exact linear-time algorithm for some sub-class of trees and a 5/3-approximation algorithm for arbitrary trees. The existence of an exact polynomial-time algorithm for arbitrary trees is left open.

Our Results. We show that the Black Hole Search problem is not approximable in polynomial time within a $1 + \varepsilon$ factor for any $\varepsilon < \frac{1}{388}$, unless **P**=**NP**. Moreover, we give a 6-approximation algorithm for this problem, i.e., a polynomial time algorithm which, for any input instance, produces a black hole search scheme with cost at most 6 times the best cost of a black hole search scheme for this input. This improves on the 9.3-approximation algorithm shown in [2]. Finally we prove that the restricted case in which only the starting node is initially known to be safe is also **APX**-hard.

2 Model and Terminology

We represent a network as a connected undirected graph $G = (V, E)$, without multiple edges or self-loops, where nodes denote hosts and edges denote communication links. The two agents, called *Agent*-1 and *Agent*-2, start the black hole search from a STARTING NODE $s \in V$ and explore graph G by traversing its edges. Together with the starting node s, a subset of nodes S which are initially known to be safe is given. Let $U = V \setminus S$, and let $B \subseteq U$, $|B| \leq 1$, denote the (unknown) set of nodes containing a black hole (we have either $B = \emptyset$ or $B = \{b\}$). We recall the formalization of the Black Hole Search problem given in [10], extending it to the case of S containing more nodes than only s, in the following way.

(General) Black Hole Search problem (gBHS)

Instance : a connected undirected graph $G = (V, E)$, a subset of nodes $S \subset V$ and a node $s \in S$.
Solution : a feasible EXPLORATION SCHEME $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ for (G, S, s), where $\mathbb{X} = \langle x_0, x_1, \ldots, x_T \rangle$ and $\mathbb{Y} = \langle y_0, y_1, \ldots, y_T \rangle$ are two equal-length sequences of nodes in G. The feasibility of $\mathcal{E}_{G,S,s}$ is determined by constraints 1–4 given below. The length of $\mathcal{E}_{G,S,s}$ is defined to be T.
Measure : the cost of the Black Hole Search (BHS) based on $\mathcal{E}_{G,S,s}$.
Goal : minimization.

When the BHS based on a given exploration scheme $\mathcal{E}_{G,S,s}$ is performed in G, *Agent*-1 follows the path defined by $\mathbb{X}$ while *Agent*-2 follows the path defined by $\mathbb{Y}$. At the end of the i-th step of the search (at time i), *Agent*-1 is in node x_i while *Agent*-2 is in node y_i. As soon as an agent deduces the value of B, it "aborts" the exploration and returns to the starting node s by traversing nodes in $V \setminus B$. The cost of the BHS based on $\mathcal{E}_{G,S,s}$ is defined later in this section.

If $\mathbb{X} = \langle x_0, x_1, \ldots, x_T \rangle$ and $\mathbb{Y} = \langle y_0, y_1, \ldots, y_T \rangle$ are two equal-length sequences of nodes in G, then $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ is a feasible exploration scheme for the input (G, S, s) (and can be effectively used as a basis for a BHS in G) if the constraints 1–4 stated below are satisfied.

Constraint 1: $x_0 = y_0 = s$, $x_T = y_T$.

Constraint 2: for each $i = 0, \ldots, T-1$, either $x_{i+1} = x_i$, or $(x_i, x_{i+1}) \in E$; and similarly either $y_{i+1} = y_i$ or $(y_i, y_{i+1}) \in E$.

Constraint 3: $U \subseteq \bigcup_{i=0}^{T} \{x_i\} \cup \bigcup_{i=0}^{T} \{y_i\}$.

Constraint 1 corresponds to the fact that both agents start from the given starting node s. The requirement that the sequences $\mathbb{X}$ and $\mathbb{Y}$ end at the same node provides a convenient simplification of the reasoning without loss of generality. Constraint 2 models the fact that during each step, each agent can either WAIT in the node v where it was at the end of the previous step, or traverse an edge of the network to move to a node adjacent to v. Constraint 3 assures that each node in U is visited by at least one agent during the exploration. We need additional definitions to state Constraint 4.

Given an exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$, for each $i = 0, 1, \ldots, T$, we call the EXPLORED TERRITORY at step i the set S_i defined in the following way:

$$S_i = \begin{cases} S \cup \bigcup_{j=0}^{i} \{x_j\} \cup \bigcup_{j=0}^{i} \{y_j\}, & \text{if } x_i = y_i; \\ S_{i-1}, & \text{otherwise.} \end{cases}$$

Thus $S_0 = S$ by Constraint 1, $S_T = V$ by Constraint 1 and Constraint 3, and $S_{j-1} \subseteq S_j$ for each step $1 \leq j \leq T$. A node v is EXPLORED at step i if $v \in S_i$, or UNEXPLORED otherwise. An unexplored node v may have been already visited by one of the agents, but it will become explored only when the agents meet, and communicate, next time (the agents communicate with each other, exchanging their full knowledge, when and only when they meet at a node). If both agents are alive at the end of step i, then the explored nodes at this step are all nodes which are known to *both* agents to be safe. Note that the explored territory is defined for an exploration scheme $\mathcal{E}_{G,S,s}$, not for the BHS based on $\mathcal{E}_{G,S,s}$, and does not take into account the possible existence of the black hole. This is taken into account in the definition of the cost of the BHS based on $\mathcal{E}_{G,S,s}$.

A MEETING STEP (or simply MEETING) is the step 0 and every step $1 \leq j \leq T$ such that $S_j \neq S_{j-1}$. Observe that, for each meeting step j, we must have $x_j = y_j$, but not necessarily the opposite, and we call this node a MEETING POINT. The meeting steps are the steps when the agents meet and add at least one new node to the explored territory. A sequence of steps $\langle j+1, j+2, \ldots, k\rangle$ where j and k are two consecutive meetings is called a PHASE of length $k - j$. We give now the last constraint on a feasible exploration scheme.

Constraint 4: for each phase with a sequence of steps $\langle j+1, j+2, \ldots, k\rangle$,
(a) $|\{x_{j+1}, \ldots, x_k\} \setminus S_j| \leq 1$ and $|\{y_{j+1}, \ldots, y_k\} \setminus S_j| \leq 1$; and
(b) $\{x_{j+1}, \ldots, x_k\} \setminus S_j \neq \{y_{j+1}, \ldots, y_k\} \setminus S_j$.

Constraint 4(a) means that during each phase, one agent can visit at most one unexplored node. If it visited two or more unexplored nodes and one of them was a black hole, then the other, surviving, agent would not know where exactly the black hole is. Constraint 4(b) says that the same unexplored node cannot be visited by both agents during the same phase, or otherwise they both may end up in a black hole (see [3] for a formal proof of this fact). From now on an exploration scheme means a feasible exploration scheme. We recall from [10] the next two simple observations.

Lemma 1. *If $k \geq 1$ is a meeting step for an exploration scheme $\mathcal{E}_{G,S,s}$, then $x_k = y_k \in S_{k-1}$.*

Lemma 2. *Each phase of an exploration scheme $\mathcal{E}_{G,S,s}$ has length at least* 2.

Any phase $\langle j+1, j+2\rangle$ of length 2 which expands the explored territory by 2 nodes has to have the following structure. Let m be the meeting point at step j. During step $j+1$, *Agent*-1 visits an unexplored node v_1 adjacent to m, while *Agent*-2 visits an unexplored node v_2 adjacent to m as well, and $v_1 \neq v_2$. In step $j+2$, the agents meet in a node which has been already explored and is adjacent to both v_1 and v_2. This node can be either m, and in this case we denote the phase as

b-split(m, v_1, v_2), or a different node $m' \neq m$, and in this case the phase is denoted as **a-split**(m, v_1, v_2, m').

For an exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ and a location of a black hole B, the EXECUTION TIME is defined as follows. If $B = \emptyset$, then the execution time is equal to the length T of the exploration scheme, plus the shortest path distance from $x_T (= y_T)$ to s. If $B = \{b\} \subseteq U$, then let j be the first step in $\mathcal{E}_{G,S,s}$ such that $b \in S_j$. Observe that j must be a meeting step and $1 \leq j \leq T$, since $S_0 = S$ and $S_T = V$. The execution time in this case is equal to j plus the length of the shortest path from $x_j (= y_j)$ to s not including b. In this case one agent, say *Agent*-1, vanishes into the black hole during the phase ending at step j, so it does not show up to meet *Agent*-2 at node $x_j = y_j$. Since, by Constraint 4, *Agent*-1 has visited only one unexplored node during the phase, the surviving *Agent*-2 learns the exact location of the black hole and thus it goes back to s, obviously omitting the black hole.

The COST of the BHS based on an exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ is denoted by *cost*$(\mathcal{E}_{G,S,s})$ and defined as the worst (maximum) execution time of $\mathcal{E}_{G,S,s}$ over all possible values of B (including $B = \emptyset$). The target of the black hole search problem is to find an exploration scheme $\mathcal{E}_{G,S,s}$ which yields a minimum cost BHS over all possible exploration schemes for (G, S, s).

The following lemma helps to simplify, at least in some cases, the computation of the cost of the BHS based on a given exploration scheme.

Lemma 3. *Let (G, S, s) be an input instance for the* gBHS *problem, and let U be the set of initially unexplored nodes ($U = V \setminus S$). The case $B = \emptyset$ yields the maximum execution time for any exploration scheme in (G, S, s), if and only if, by removing any node $u \in U$ from G, each node in $V \setminus \{u\}$ either becomes disconnected from s, or maintains its shortest path distance from s.* [1]

3 Approximation Lower Bound for the General BHS Problem

In this section we provide an explicit lower bound on the approximability of the General Black Hole Search problem by showing an approximation preserving reduction from a particular subcase of the Traveling Salesman Problem, presented in [7], and defined in the following way.

(1,M)-Traveling Salesman Problem (TSP(1,M))

Instance : a pair (G, d), where $G = (V, E)$ is a complete graph (with $n = |V|$) and $d : V^2 \rightarrow \{1, \ldots, M\}$ is a distance function associating to each pair of nodes (v, u) a positive integer length $d(v, u)$ between 1 and M (where M is a constant).

[1] Due to space constraints, the proofs of some lemmas have been omitted in this extended abstract.

Function d is symmetric (i.e., $d(u,v) = d(v,u)$) and satisfies the triangle inequality (i.e., $d(i,j) + d(j,k) \geq d(i,k)$, $\forall i,j,k \in V$).

Solution : a tour τ of G, i.e., a permutation $\tau = \langle v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$ of the nodes in V.

Measure : the *length* (or *cost*) of the tour, i.e., $cost(\tau) = \sum_{i=1}^{n-1} d(v_{\pi(i)}, v_{\pi(i+1)}) + d(v_{\pi(n)}, v_{\pi(1)})$.

Goal : minimization.

In [7] it is also presented a lower bound on the approximability of such problem.

Lemma 4. *It is* ***NP****-hard to approximate* TSP(1,8) *within* $1+\varepsilon$ *for any* $\varepsilon < \frac{1}{388}$.

Reduction from instances (G,d) of TSP(1,M) to instances (G', S, s) of gBHS. Let (G,d) be an instance of TSP(1,M). We define the graph $G' = (V', E')$, the set $S \subset V'$, and the starting node s, in the following way. Let v_1 be an arbitrary node in V. We add v_1 to V' and to S, and we define $s = v_1$. For each node v_i $(2 \leq i \leq n)$ in V, we add to V' a pair of nodes v'_i, v''_i. We denote node v_1 as the ISLAND I_1, and each pair of nodes v'_j, v''_j as the ISLAND I_j. For each edge (v_i, v_j) in E of length $d(v_i, v_j)$, we add to V' (and to E') a path of $2 \cdot d(v_i, v_j) - 1$ nodes (BRIDGE $i \leftrightarrow j$), whose endpoints are adjacent respectively to v'_i, v''_i (or v_1 if $i = 1$) and to v'_j, v''_j (or v_1 if $j = 1$). We add all the nodes of the bridge to S. We call as $b_{i,j}$ and as $b_{j,i}$ the endpoints of bridge $i \leftrightarrow j$ adjacent respectively to island I_i and island I_j (note that if $d(v_i, v_j) = 1$, then $b_{i,j} \equiv b_{j,i}$). Each bridge is composed by at least one (safe) node, and $|V' \setminus S| = 2(n-1)$.

Lemma 5. *The distance in G' between any node of island I_i and any node of island I_j (where $i \neq j$ and $i, j = 1, \ldots, n$) is equal to $2 \cdot d(v_i, v_j)$.*

The following lemma gives a useful characterization of G'.

Lemma 6. *Let G' be a graph produced with the above mentioned construction. The case $B = \emptyset$ yields the maximum execution time for any exploration scheme in G'.*

Now we define an exploration scheme on G' which explores the islands in G', in the order defined by a tour on G. In the following definition we introduce a new keyword: **walk**. By **walk**(b) we mean that both agents (which are supposed to be already in the same node w), move to b by following a shortest (safe) path from w to b. Actually, the walk is not a complete phase (no new nodes are explored), but it is the initial part of a phase.

Let $\tau = \langle v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$ be a tour on G of length l. We assume w.l.o.g. that $\pi(1) = 1$. A τ-BASED EXPLORATION SCHEME $\mathcal{E}^{\tau}_{G',S,s}$ on G' consists of the following sequence of steps:

1. **walk**$(b_{1,\pi(2)})$, where $b_{1,\pi(2)}$ is the node adjacent to s on the bridge $1 \leftrightarrow \pi(2)$;
2. for each $i = 2, \ldots, n$:
   (a) **walk**$(b_{\pi(i),\pi(i-1)})$, where $b_{\pi(i),\pi(i-1)}$ is the node adjacent to $I_{\pi(i)}$ on the bridge $\pi(i-1) \leftrightarrow \pi(i)$;
   (b) **a-split**$(b_{\pi(i),\pi(i-1)}, v'_{\pi(i)}, v''_{\pi(i)}, b_{\pi(i),\pi(i+1)})$, where $b_{\pi(i),\pi(i+1)}$ is the node adjacent to $I_{\pi(i)}$ on the bridge $\pi(i) \leftrightarrow \pi(i+1)$ (or bridge $\pi(n) \leftrightarrow 1$ if $i = n$).

Given the tour τ in G, the τ-based exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',S,s}$ can be obviously constructed in linear time. In the following lemma we compute the cost of the Black Hole Search based on $\mathcal{E}^{\mathcal{T}}_{G',S,s}$.

Lemma 7. *Given a tour $\tau = \langle v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$ on G of length l, the τ-based exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',S,s}$ satisfies* $\mathrm{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s}) = 2 \cdot l$.

Corollary 1. *Let (G, d) be an instance of the* TSP(1,M) *problem, and let (G', S, s) be the corresponding instance of the* BHS problem *where the graph G' is constructed as explained before. Moreover, let τ^* be an optimal solution for (G, d) and let $\mathcal{E}^*_{G',S,s}$ be an optimal solution for (G', S, s). Then* $\mathrm{cost}(\mathcal{E}^*_{G',S,s}) \leq 2 \cdot \mathrm{cost}(\tau^*)$.

In what follows we show a method to modify an exploration scheme without altering its properties (i.e., feasibility, length, sequence of explored territories and the cost of the BHS based on it). We then use this technique to impose a characteristic of "regularity" to any exploration scheme on G'.

Definition 1. *Let $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ be an exploration scheme for (G, S, s), and let $\phi = (\mathbb{X}_\phi, \mathbb{Y}_\phi)$ be a phase in $\mathcal{E}_{G,S,s}$. Let $\mathcal{E}'_{G,S,s}$ be the exploration scheme obtained from $\mathcal{E}_{G,S,s}$ by swapping the paths of the two agents in phase ϕ, i.e., $\phi' = (\mathbb{Y}_\phi, \mathbb{X}_\phi)$. We call this operation a* PHASE-SWAP. *Two exploration schemes are* EQUIVALENT *if and only if one is obtained from the other by applying a finite sequence of phase-swaps.*

Lemma 8. *Let $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ be an exploration scheme for (G, S, s). Let $\mathcal{E}'_{G,S,s}$ be the exploration scheme obtained from $\mathcal{E}_{G,S,s}$ by applying a phase swap on $\mathcal{E}_{G,S,s}$. Then, the exploration scheme $\mathcal{E}'_{G,S,s}$ is feasible, has exactly the same meeting points, the same sequence of explored territories and the same length as $\mathcal{E}_{G,S,s}$. Moreover,* $\mathrm{cost}(\mathcal{E}'_{G,S,s}) = \mathrm{cost}(\mathcal{E}_{G,S,s})$.

Corollary 2. *Two equivalent exploration schemes have exactly the same meeting points, the same sequence of explored territories and the same length. Moreover the cost of the BHS based on them is the same.*

We now turn back our focus to instances (G', S, s) constructed by reduction from instances (G, d). We give a classification of each phase of any exploration scheme in G'. A phase ϕ is a *2s-phase* if the two nodes of the same island are explored during ϕ. It is a *2d-phase*, if two nodes in two distinct islands are explored during ϕ. Finally, it is a *1-phase* if only one node is explored during ϕ.

Definition 2. *Given an exploration scheme $\mathcal{E}_{G',S,s}$, we define the* PHASE GRAPH *of $\mathcal{E}_{G',S,s}$, the following directed multigraph $P(\mathcal{E}_{G',S,s})$. The graph $P(\mathcal{E}_{G',S,s})$ has the nodes $v_2, \ldots, v_n$ corresponding to the islands $I_2, \ldots, I_n$ in G', plus one further node which we call x. The following edges are added to $P(\mathcal{E}_{G',S,s})$:*

- *a directed edge $\langle v_i, x\rangle$ ($\langle x, v_i\rangle$) is added for each node in island I_i which is explored during a 1-phase by Agent-1 (Agent-2);*
- *a directed edge $\langle v_i, v_j\rangle$ is added for each 2d-phase exploring a node of island I_i with Agent-1 and a node of island I_j with Agent-2;*
- *a directed self-loop $\langle v_i, v_i\rangle$ is added if the nodes of island I_i are explored by a 2s-phase.*

Lemma 9. *Given any exploration scheme $\mathcal{E}_{G',S,s}$, each node of the phase graph $P(\mathcal{E}_{G',S,s})$ has degree (= in-degree + out-degree) equal to 2.*

The graph $P(\mathcal{E}_{G',S,s})$ is thus a set of connected components. In the underlying undirected multigraph, these components are either cycles or isolated nodes. Now we give a new characterization of an exploration scheme in G'.

Definition 3. *An exploration scheme $\mathcal{E}_{G',S,s}$ is* REGULAR *if and only if each agent explores exactly one node of each island I_j, with $j = 2, \ldots, n$.*

Notice that any τ-based exploration scheme is regular; we can observe that each node in $P(\mathcal{E}^{\tau}_{G',S,s})$ is an isolated node (the only adjacent edge is a self-loop). Indeed, we can prove a tighter relation between regular exploration schemes and their corresponding phase graph.

Lemma 10. *An exploration scheme $\mathcal{E}_{G',S,s}$ is regular if and only if, in the corresponding phase graph $P(\mathcal{E}_{G',S,s})$, for each node v_i, $indeg(v_i) = 1$ and $outdeg(v_i) = 1$.*

Lemma 11. *For any exploration scheme $\mathcal{E}_{G',S,s}$ there is an equivalent regular one that can be found in linear time.*

Sketch of the proof. By Lemma 10, this corresponds to transform $P(\mathcal{E}_{G',S,s})$ into a graph where, for each node v_i, $indeg(v_i) = 1$ and $outdeg(v_i) = 1$. This can be easily achieved in linear time, since each phase-swap in $\mathcal{E}_{G',S,s}$ produces a change in the orientation of the corresponding edge in $P(\mathcal{E}_{G',S,s})$. □

Lemma 12. *Given an exploration scheme $\mathcal{E}_{G',S,s}$, we can find in linear time a tour τ on G such that, for the τ-based exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',S,s}$,* $\mathrm{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s}) \leq \mathrm{cost}(\mathcal{E}_{G',S,s})$.

Proof. By Corollary 2 and Lemma 11, we can assume w.l.o.g. that $\mathcal{E}_{G',S,s}$ is a regular exploration scheme. By regularity, *Agent*-1 explores a node of each island in G'. Let $I_{\mathbb{X}} = \langle I_{\pi(2)}, \ldots, I_{\pi(n)} \rangle$ be the sequence of the islands in G' in the order they are explored by *Agent*-1. Let τ be the tour in G corresponding to $I_{\mathbb{X}}$ (i.e., $\tau = \langle v_1, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$), and let $l = \mathit{cost}(\tau)$. We show that the τ-based exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',S,s}$ is such that $\mathit{cost}(\mathcal{E}_{G',S,s}) \geq \mathit{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s})$. Consider the case $B = \emptyset$ (Lemma 6). *Agent*-1 starts from s, visits islands in $I_{\mathbb{X}}$ and then gets back to s. By Lemma 5, the length of this tour is at least $2 \cdot l$. The execution time of $\mathcal{E}_{G',S,s}$ cannot be shorter than $2 \cdot l$. Therefore, $\mathit{cost}(\mathcal{E}_{G',S,s}) \geq 2 \cdot l \geq \mathit{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s})$. □

Lemma 13. *Let G be an instance of the* TSP(1,M) *problem, and let G' be the corresponding instance of the* gBHS *problem. Moreover, let τ^* be an optimal tour in G, and let $\mathcal{E}^*_{G',S,s}$ be an optimal exploration scheme for G'. Let $\varepsilon > 0$. If we can find in polynomial time an exploration scheme $\mathcal{E}_{G',S,s}$ such that* $\mathrm{cost}(\mathcal{E}_{G',S,s}) \leq \mathrm{cost}(\mathcal{E}^*_{G',S,s})(1+\varepsilon)$, *then we can find in polynomial time a tour τ in G such that* $\mathrm{cost}(\tau) \leq \mathrm{cost}(\tau^*)(1+\varepsilon)$.

Proof. Suppose that, given a graph G', we can construct in polynomial time an exploration scheme $\mathcal{E}_{G',S,s}$ such that its cost is at most $1+\varepsilon$ times the cost of an optimal exploration scheme. By Lemma 12, we can find an exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',S,s}$, based on a tour τ in G, such that $\mathit{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s}) \leq \mathit{cost}(\mathcal{E}_{G',S,s}) \leq \mathit{cost}(\mathcal{E}^*_{G',S,s})(1+\varepsilon)$. Supposing that the length of the tour τ is l, then, by Lemma 7: $\mathit{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s}) = 2 \cdot l$. Supposing that the length of the optimal tour τ^* is l^*, then, by Corollary 1: $\mathit{cost}(\mathcal{E}^*_{G',S,s}) \leq 2 \cdot l^*$. Therefore, by hypothesis: $2 \cdot l = \mathit{cost}(\mathcal{E}^{\mathcal{T}}_{G',S,s}) \leq \mathit{cost}(\mathcal{E}^*_{G',S,s})(1+\varepsilon) \leq 2 \cdot l^*(1+\varepsilon)$, and hence, $l \leq l^*(1+\varepsilon)$. □

The main theorem immediately follows from Lemma 4 and Lemma 13.

Theorem 1. *The* gBHS *problem is not approximable in polynomial time within a factor of $1+\varepsilon$ for any $\varepsilon < \frac{1}{388}$, unless **P=NP**.*

4 The Restricted BHS Problem is APX-hard

In this section we consider the restricted version of the BHS problem in which $S = \{s\}$, i.e., the starting point is the only node initially known to be safe (we denote it as rBHS). We show that the BHS problem with this restriction remains **APX**-hard. The input of rBHS is fully specified by providing a graph G and the starting node s. We will hence use now the notation $\mathcal{E}_{G,s}$ to refer to an exploration scheme.

We will prove **APX**-hardness of the rBHS problem using **APX**-hardness of the TSP(1,2) problem. We first recall Lemma 6.3 from [1]:

Lemma 14. *Assume we are given an instance of* TSP(1,2) *on the n-node complete graph $\overline{G}$, in the form of the subgraph G of $\overline{G}$ containing the edges of weight* 1*. Assume that G has max degree* 3*. Assume that we know that its minimum cost TSP tour is either of cost n or at least $(1+\varepsilon_0)n$, for some fixed ε_0. Then there exists such a constant ε_0 for which it is* ***NP****-hard to decide which of the two cases holds. The claim holds for $\varepsilon_0 = \frac{1}{786}$. If G is cubic then the claim holds for $\varepsilon_0 = \frac{1}{1290}$.*

We show a polynomial-time reduction algorithm $\mathcal{A}$ from TSP(1,2) to rBHS, which takes as input an instance G of TSP(1,2), computes an instance (G', s) of rBHS, and has the following property.

Lemma 15. *Let $0 < \varepsilon < \varepsilon_0/7$, let G be an n-node cubic graph (an instance of* TSP(1,2)*), and let (G', s) be the corresponding instance of* rBHS *computed by the reduction algorithm $\mathcal{A}$. Then the following two conditions hold.*

1. *If the optimal cost of a tour in G is equal to n, then the optimal cost of an exploration scheme for (G', s) is at most $\frac{7}{2}n + 1$.*
2. *There exists $n_0 = n_0(\varepsilon_0, \varepsilon)$ such that for $n \geq n_0$, if the optimal cost of a tour in G is at least $n(1+\varepsilon_0)$, then the optimal cost of an exploration scheme for (G', s) is greater than $\left(\frac{7}{2}n + 1\right)(1+\varepsilon)$.*

This lemma implies that for $0 < \varepsilon < \varepsilon_0/7$ and $n \geq n_0$, if we have an n-node cubic graph G and we know that the optimal cost of a tour in G either is equal to n or is at least $n(1+\varepsilon_0)$, then we can decide which of these two cases happens, if we have an $(1+\varepsilon)$-approximation of the optimal cost of an exploration scheme for (G', s). Thus Lemmas 14 and 15 imply the following theorem.

Theorem 2. *It is* ***NP****-hard to compute $(1+\varepsilon)$-approximate exploration schemes for the* rBHS *problem for any $\varepsilon < \frac{1}{9030}$.*

Description of the reduction algorithm $\mathcal{A}$. Let an n-node graph $G = (V, E)$ be the input instance of TSP(1,2). The construction of the instance (G', s) of rBHS proceeds as follows. We pick an arbitrary node in G (say v_1) as the starting node ($s \equiv v_1$) and we add it to G' (as before, this is island I_1). For each node v_i in G, $2 \leq i \leq n$, we add in G' a pair of unexplored nodes v'_i, v''_i (as before, we denote this pair as island I_i). For each edge (v_i, v_j) in G, we put in G' an unexplored node $b_{i,j}$ (*bridge node*), connected to v'_i, v''_i (if $i > 1$), to v'_j, v''_j (if $j > 1$) and to s. If the number of bridge nodes (that is, the number of edges in G) is odd, then we add another unexplored node b_s adjacent to s (to ensure that s is adjacent to an even number of unexplored nodes). Node s is adjacent to all bridge nodes and is not adjacent to any "island" nodes.

Sketch of Proof of Lemma 15. Let G be an n-node cubic graph. Since G has $m = \frac{3}{2}n$ edges, the total number of nodes in G' is $\frac{7}{2}n - 1 + odd(m)$, and all of them but one are initially unexplored. For an integer k, $odd(k)$ is equal to 1, if k is odd, and to 0 otherwise. As in Section 3, we define for a tour $\tau = \langle v_1, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$ in G, the exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',s}$ for (G', s), which explores two by two the nodes of each island in the order $\langle I_{\pi(2)}, \ldots, I_{\pi(n)} \rangle$. Here, however, the scheme first explores the bridge nodes.

More formally, the scheme $\mathcal{E}^{\mathcal{T}}_{G',s}$ has the following sequence of steps.

1. While there are two unexplored nodes b', b'' adjacent to s: **b-split**(s, b', b'').
2. For each $i = 2, \ldots, n$:
   (a) **walk**(b'), where b' is either the bridge node $b_{\pi(i-1),\pi(i)}$, if nodes $v_{\pi(i-1)}$ and $v_{\pi(i)}$ are adjacent in G, or any bridge node adjacent to I_i otherwise.
   (b) **a-split**$(b', v'_{\pi(i)}, v''_{\pi(i)}, b'')$, where b'' is either the bridge node $b_{\pi(i),\pi(i+1)}$, if $i < n$ and nodes $v_{\pi(i)}$ and $v_{\pi(i+1)}$ are adjacent in G, or any bridge node adjacent to I_i otherwise.

The first **walk** operation, for $i = 2$, has length 1. For each $3 \le i \le n$, the **walk** operation has length either 0, if nodes $v_{\pi(i-1),\pi(i)}$ are adjacent in G, or 2, if nodes $v_{\pi(i-1),\pi(i)}$ are not adjacent in G. Therefore, if the tour τ has cost $n + d$ (that is, contains d edges of weight 2), then the exploration scheme $\mathcal{E}^{\mathcal{T}}_{G',s}$ has length at most $\frac{3}{2}n + odd(m) + 1 + 2d + 2(n-1) \le \frac{7}{2}n + 2d$. The execution time for the case $B = \emptyset$ is at most $\frac{7}{2}n + 2d + 1$, since $\mathcal{E}^{\mathcal{T}}_{G',s}$ ends in a bridge node, which is adjacent to s. This is also the cost of the BHS based on $\mathcal{E}^{\mathcal{T}}_{G',s}$. When an agent realizes that there is a black hole, then this agent must be at a meeting point, and each meeting point is either node s or a bridge node, which is adjacent to s. Hence, if the cost of tour τ is n, then $d = 0$ and the cost of $\mathcal{E}^{\mathcal{T}}_{G',s}$ is at most $\frac{7}{2}n + 1$, so the first part of Lemma 15 holds.

To prove the second part of Lemma 15, consider an arbitrary exploration scheme $\mathcal{E}_{G',s}$. By using a similar approach as in Section 3, we can find, through a sequence of phase swaps, a "regular" exploration scheme $\mathcal{E}'_{G',s}$, equivalent to $\mathcal{E}_{G',s}$, where each agent explores exactly one node of each island I_j for $j = 2, \ldots, n$, and $cost(\mathcal{E}'_{G',s}) = cost(\mathcal{E}_{G',s})$. We assume by symmetry that scheme $\mathcal{E}'_{G',s}$ is such that *Agent*-1 explores nodes v'_j, $j = 2, \ldots, n$, and that $\langle v'_{\pi(2)}, \ldots, v'_{\pi(n)} \rangle$ is the order in which *Agent*-1 explores these nodes. We consider the tour $\tau = \langle v_1, v_{\pi(2)}, \ldots, v_{\pi(n)} \rangle$ in G.

Let d be the number of weight 2 edges in τ. Thus the number of indices i, $2 \le i \le n-1$, such that $(v_{\pi(i)}, v_{\pi(i+1)})$ is not an edge in G is at least $d - 2$. Consider any of these indices i and two consecutive phases ϕ_{j_i} and ϕ_{j_i+1} in $\mathcal{E}'_{G',s}$, where ϕ_{j_i} is the phase during which node $v'_{\pi(i)}$ is explored by *Agent*-1. It can be shown that at least one of the two phases ϕ_{j_i} and ϕ_{j_i+1} is not a split, so at least $(d-2)/2$ phases in scheme $\mathcal{E}'_{G',s}$ are not splits.

The cost of any exploration scheme is at least the number of unexplored nodes plus the number of phases other than splits. Therefore, we have $cost(\mathcal{E}'_{G',s}) \geq \frac{7}{2}n - 3 + \frac{d}{2}$. This implies that if $cost(\mathcal{E}'_{G',s}) \leq \left(\frac{7}{2}n + 1\right)(1+\varepsilon)$, then $d \leq 7\varepsilon n + 2(4+\varepsilon)$, and

$$cost(\tau) = n + d \leq n + 7\varepsilon n + 2(4+\varepsilon) \leq n(1+\varepsilon_0) - (\varepsilon_0 - 7\varepsilon)n + 2(4+\varepsilon) < n(1+\varepsilon_0),$$

provided that $\varepsilon < \varepsilon_0/7$ and $n \geq n_0 = \lceil 2(4+\varepsilon)/(\varepsilon_0 - 7\varepsilon) + 1 \rceil$.

5 A 6-approximation algorithm for the General BHS Problem

Let G, S and U be defined as in Section 2. We define the distance graph $\widehat{G}$ as the complete weighted graph in which the set of nodes corresponds to the nodes in $U \cup \{s\}$ and the weight of edge (v_i, v_j) is the shortest path distance from v_i to v_j in G (considering both safe and unexplored nodes). Weights in $\widehat{G}$ satisfy triangle inequality. Let T be the minimum spanning tree of $\widehat{G}$ rooted at s, and let $cost(T)$ be its cost, i.e., the sum of the weights of all its edges. Let L_G be the sequence obtained from the depth first traversal of T, $\langle v_0 \equiv s, v_1, \ldots, v_u \rangle$, by replacing each pair of adjacent nodes v_i, v_{i+1} with the shortest path in G from v_i to v_{i+1}. Since the distance from v_i to v_{i+1} is at most the (weighted) cost of path $v_i, \ldots, v_{i+1}$ in T, the length of L_G is at most $2cost(T) - d(v_u, s)$.

We now construct the exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ for G. Initially $\mathbb{X} = \mathbb{Y} = L_G$. Then, the pairs of adjacent steps $\langle x_i, x_{i+1} \rangle$ and $\langle y_i, y_{i+1} \rangle$ are considered from $i = 1, \ldots, k$. If $x_i = y_i = v'$ and $x_{i+1} = y_{i+1} = v''$, where v'' is an unexplored node occurring for the first time in the sequences, we replace $\langle v', v'' \rangle$ in $\mathbb{X}$ with the sequence $\langle v', v'', v', v'' \rangle$ and we replace $\langle v', v'' \rangle$ in $\mathbb{Y}$ with the sequence $\langle v', v', v', v'' \rangle$. This is to assure that each time the agents have to visit an unexplored node, *Agent*-1 first explores it by using the technique of *probing*. Since $|U|$ is the number of unexplored nodes, $2|U|$ steps are added to exploration scheme $\mathcal{E}_{G,S,s}$. The length of $\mathcal{E}_{G,S,s}$ is therefore at most $2cost(T) - d(v_u, s) + 2|U|$, while the execution time in the case $B = \emptyset$ is at most $2cost(T) + 2|U|$ since the surviving agents have to get back from v_u to s. Observing that $B = \emptyset$ yields the worst case for the execution time since we are operating on a tree, we can derive the following lemma.

Lemma 16. *The exploration scheme $\mathcal{E}_{G,S,s}$ is feasible and* $\mathrm{cost}(\mathcal{E}_{G,S,s}) \leq 2\mathrm{cost}(T) + 2|U|$.

Consider now an optimal exploration scheme $\mathcal{E}^*_{G,S,s} = (\mathbb{X}^*, \mathbb{Y}^*)$. In computing $cost(\mathcal{E}^*_{G,S,s})$ we consider, as lower bound, the execution time of $\mathcal{E}^*_{G,S,s}$ in the case $B = \emptyset$. Let $L' = (x_k, \ldots, s)$ be the shortest path in G from the last node x_k in $\mathbb{X}^*$ to the starting node, excluding the endpoints x_k and s. Let $L'' = \mathbb{X}^* \circ L' \circ \mathbb{Y}^* \circ L' \circ \langle s \rangle$. The sequence L'' starts from s, visits all the nodes in U and ends in s; its length

$|L''|$, is at most twice the execution time of $\mathcal{E}^*_{G,S,s}$ in the case $B = \emptyset$, and hence $2\,cost(\mathcal{E}^*_{G,S,s}) \geq |L''|$. Let L^* be the minimum (shortest) tour in G starting from s and visiting all the nodes in U, and let $|L^*|$ be its length; obviously, $|L''| \geq |L^*|$.

Due to its optimality, L^* has the following structure: $L^* = \langle s \rangle \circ P(s, u_1\rangle \circ P(u_1, u_2\rangle \circ \ldots \circ P(u_u, s\rangle$ where $\langle u_1, \ldots, u_u\rangle$ is the sequence of unexplored nodes in the order they are visited for the first time in L^*, and $P(x, y\rangle$ is the shortest path from node x (excluded) to node y in G. Since weights in G satisfy triangle inequality, the length of L^* is equal to the length of the minimum traveling salesman tour in $\widehat{G}$, which is at least the cost of the minimum spanning tree T of $\widehat{G}$. Therefore, $|L^*| \geq cost(T)$, and $cost(\mathcal{E}^*_{G,S,s}) \geq \frac{cost(T)}{2}$. Moreover, the trivial lower bound holds: $cost(\mathcal{E}^*_{G,S,s}) \geq |U|$. We compute the approximation ratio of the algorithm presented in this section, by choosing a suitable balance for the two bounds on the optimal cost. Therefore: $\frac{cost(\mathcal{E}_{G,S,s})}{cost(\mathcal{E}^*_{G,S,s})} \leq \frac{2\,cost(T)+2\,|U|}{\frac{2}{3}\frac{cost(T)}{2}+\frac{1}{3}\,|U|} = 6\,.$

Theorem 3. *The* gBHS *problem is approximable within* 6*.*

6 Conclusions

We showed that the problem of computing an optimal exploration scheme for a BHS with two agents (the gBHS problem) is not approximable within $\frac{389}{388}$ (unless **P**=**NP**). We also showed that for the restricted version of this problem (the rBHS problem), when initially only one, starting node is known to be safe, approximating within any factor less than $\frac{9031}{9030}$ is **NP**-hard. We have presented a polynomial-time 6-approximation algorithm for the gBHS problem (while a polynomial-time $3\frac{1}{2}$-approximation algorithm for the rBHS problem was previously shown in [10]).

It seems very difficult to reduce significantly the gap between the upper and lower bounds on the approximation ratios for the gBHS and rBHS problems. However, some small improvements can be achieved, for example, by showing, with a more detailed analysis, that Lemma 15 holds also for $0 < \varepsilon < 2\varepsilon_0/7$ and for graphs G of maximum degree 3. This improves the constant in the lower bound for the rBHS problem to $\frac{2752}{2751}$. Since our lower bounds are based on reductions from problems TSP(1,8) and TSP(1,2), any improvements of the inapproximability results for those problems will directly lead to improved lower bounds for our problems.

We believe that we can improve the 6 approximation ratio, by a more detailed analysis of the bad case, when the two lower bounds on the optimal cost of an exploration scheme are similar. More precisely, if the ratio $cost(T)/2|U|$ is in the range $[1-\delta, 1+\delta]$, for some small constant $\delta > 0$, then one should be able to derive a lower constant than 6 for the bound (5) using a similar analysis as in [10]. If $cost(T)/2|U|$

is outside of this range, then the left-hand side of (5) is less than $6 - \delta$. This approach would however lead most likely only to a small improvement, while requiring substantial expansion and refinement of technical details.

As already observed in Section 1, it would be interesting to investigate how one could model and analyse the more practical and more general case of multiple black holes search, possibly performed by more than two agents.

References

1. B. Csaba, M. Karpinski, and P. Krysta. Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM Symposium on Discrete algorithms*, pages 74–83, Philadelphia, PA, USA, 2002.
2. J. Czyzowicz, D. Kowalski, E. Markou, and A. Pelc. Complexity of searching for a black hole. 2004. manuscript.
3. J. Czyzowicz, D. Kowalski, E. Markou, and A. Pelc. Searching for a black hole in tree networks. In *Proc. 8th Int. Conf. on Principles of Distributed Systems (OPODIS 2004)*, pages 35–45, 2004.
4. S. Dobrev, P. Flocchini, R. Kralovic, G. Prencipe, P. Ruzicka, and N. Santoro. Black hole search by mobile agents in hypercubes and related networks. In *Proc. 6th Int. Conf. on Principles of Distributed Systems (OPODIS 2002)*, pages 169–180, 2002.
5. S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro. Mobile agents searching for a black hole in an anonymous ring. In *Proc. 15th Int. Symposium on Distributed Computing (DISC 2001)*, Springer LNCS vol. 2180, pages 166–179, 2001.
6. S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro. Searching for a black hole in arbitrary networks: Optimal mobile agents protocols. In *Proc. 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 153–161, 2002.
7. L. Engebretsen and M. Karpinski. TSP with bounded metrics: stronger approximation hardness. Technical Report 85264, University of Bonn, February 2005.
8. F. Hohl. Time limited black box security: Protecting mobile agents from malicious hosts. In *Proc. Conf. on Mobile Agent Security*, Springer LNCS vol. 1419, pages 92–113, 1998.
9. F. Hohl. A framework to protect mobile agents by using reference states. In *Proc. 20th Int. Conf. on Distributed Computing Systems (ICDCS 2000)*, pages 410–417, 2000.
10. R. Klasing, E. Markou, T. Radzik, and F. Sarracco. Hardness and approximation results for black hole search in arbitrary graphs. In *Proc. 12th Int. Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, Springer LNCS vol. 3499, pages 200–215, 2005.
11. S. Ng and K. Cheung. Protecting mobile agents against malicious hosts by intention of spreading. In H. Arabnia, editor, *Proc. Int. Conf. on Parallel and Distributed Processing and Applications (PDPTA'99) Vol. II*, pages 725–729, 1999.
12. T. Sander and C.F. Tschudin. Protecting mobile agents against malicious hosts. In *Proc. Conf. on Mobile Agent Security*, Springer LNCS vol. 1419, pages 44–60, 1998.
13. K. Schelderup and J. Olnes. Mobile agent security – issues and directions. In *Proc. 6th Int. Conf. on Intelligence and Services in Networks*, Springer LNCS vol. 1597, pages 155–167, 1999.