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 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 any constant factor less than $\frac{389}{388}$ (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.

* Research supported in part by the European Research Training Network COMBSTRU HPRN-CT-2002-00278 and the European COST Action TIST 293 (GRAAL), the Royal Society Grant ESEP 16244, and the ACI Masses de données. Part of this work was done while E. Markou, T. Radzik and F. Sarracco were visiting the LaBRI (Laboratoire Bordelais de Recherche en Informatique) in Bordeaux.

** LaBRI - Université Bordeaux 1 - CNRS, 351 cours de la Libération, 33405 Talence cedex, France. Email: `Ralf.Klasing@labri.fr`

*** School of Computational Engineering & Science, McMaster University, 1280 Main St. West, Hamilton, Ontario L8S 4K1, Canada. Email: `markoue@mcmaster.ca`

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

‡ DIS - Dipartimento di Informatica e Sistemistica, CTL - Centro di ricerca per il Trasporto e la Logistica, Sapienza - Università di Roma. Email: `Fabiano.Sarracco@dis.uniroma1.it`

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 (if not virtually impossible) to fully protect mobile agents from malicious hosts attacks.

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 on the other nodes of the network. 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. However, no hint about the presence of a black hole can be deduced by visiting its neighborhood, and it is also assumed that an agent visiting a black hole has no way of communicating with other agents before being terminated. Therefore, it should be clear that it is possible to locate a black hole only by "sacrificing" one agent and by using another agent to indirectly infer the existence of a black hole. An agent which is to visit an unknown node can, for instance, have a meeting scheduled with another agent after such a visit, or write on a white-board in a neighboring node the label of the unknown node that he is visiting. If the visited node is a black hole, then the destroyed agent will neither turn up at the node where the meeting was scheduled nor write back to the white-board that the node has been successfully visited. In both cases, the surviving agents can deduce that the visited node is a black hole.

In this paper, we investigate the case when there may be at most one black hole in the network, and the search is performed by exactly two agents, which start from the same node s and can communicate only when they are in the same node. At least one agent must report back to s the information on where exactly the black hole is or that there is none. 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, may be at any node not in S.

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, there is no obvious and interesting measure of the time needed by the agents to find the black hole. Hence, the complexity measure considered is the total number of moves performed by the agents. For arbitrary networks of n nodes, 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 there is an upper bound on the time needed by an agent for traversing any edge. This assumption makes a dramatic change to the problem. The black hole can be located by two agents in any network and moreover the agents can decide if there is a black hole or not. To measure the efficiency of a black hole search, it is assumed 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 it is discovered that the network contains no black hole.

The running time of an algorithm producing a black hole search scheme should be distinguished from the cost (the worst-case time) of the search based on this scheme. Informally, the former is the time of preparing (planning) the walk, while the latter is the time of walking. Here, we study the optimization problem of computing a minimum-cost black hole search scheme for a given network, a given starting node and a given set of nodes initially known to be safe. From now on, the Black Hole Search problem refers to this optimization problem.

Previous Results. In [2] the authors prove that the Black Hole Search problem is **NP**-hard, and show a 9.3-approximation algorithm for it. The restricted case of this problem, when the starting node is the only node initially known to be safe ($S = \{s\}$), is considered in $[3, 10]$. In [10] the authors prove that this restricted case is also **NP**-hard, and give a $3\frac{3}{8}$-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 $G = (V, E)$, where nodes denote hosts and edges denote communication links.[1] With no loss of generality, we assume that G has no multiple edges or self-loops.

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 $B \subseteq U$ denote the (unknown) location of the black hole, with either $B = \emptyset$ or $B = \{b\}$. We formalize the general version of the Black Hole Search problem (set S can be any proper subset of V including s) in the following way.

(General) Black Hole Search problem (BHS problem)

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.

[1] In the following we will use the terms graph and network, host and node, and link and edge interchangeably, although we tend to use the term graph to mean an abstract representation of a network.

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 $\mathcal{E}_{G,S,s}$ (defined below).

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 $V \setminus B$.

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 the next time the agents meet and communicate; recall that the agents communicate with each other, exchanging their full knowledge, when and only when they meet at a node. Intuitively, the explored territory is a territory that both agents know is safe. As we will see, in light of Constraint 4, the definition of explored territory coincides with this intuition. 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, in each meeting step j, the agents must be in the same node ($x_j = y_j$), which we call a MEETING POINT. Note that the opposite is not necessarily true, i.e., there can exist non-meeting steps during which the agents are in the same node. For example, the agents could be following together a path of already explored nodes to get to a new part of the network. They would be at the same time in the same nodes, but would not be increasing the explored territory. 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 steps 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 more detailed discussion). From now on an exploration scheme means a feasible exploration scheme.

For an exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ and a subset B of unexplored nodes (with $|B| \leq 1$), the EXECUTION TIME is defined as the number of time units needed to perform the BHS based on $\mathcal{E}_{G,S,s}$, in the case that B is the set of black holes. 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. In this case the agents must perform the full exploration (spending one time unit per step) and then get back to the starting node to report that there is no black hole in the network. 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(a), *Agent*-1 has visited only one unexplored node during the phase (node b), and, by Constraint 4(b), *Agent*-2 has not visited that node, 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 $B = \emptyset$).

We recall from [10] the next two simple observations.

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

Proof. Let j be the last meeting step before step k, and hence $S_j = S_{j+1} = \cdots = S_{k-1}$. By definition $x_k = y_k \in S_k$. If $x_k = y_k$ is not in S_{k-1}, then it is in both $\{x_{j+1}, \ldots, x_k\} \setminus S_j$ and $\{y_{j+1}, \ldots, y_k\} \setminus S_j$. In this case, at least one of the conditions of Constraint 4 is violated, since either the two sets are the same or at least one of the two contains more than one node. □

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

Proof. Let us suppose, by contradiction, that there exists in $\mathcal{E}_{G,S,s}$ a phase of length 1, and hence two adjacent meeting steps j and $j+1$. The step $j+1$ is a meeting if and only if $S_{j+1} \supsetneq S_j$, but, by Lemma 1, $x_{j+1} = y_{j+1} \in S_j$, and hence $S_{j+1} = S_j$. Therefore there cannot exist in $\mathcal{E}_{G,S,s}$ a phase of length 1. □

Phases of length 2 which expand the explored territory by 2 nodes are of particular interest to us since they advance the exploration of the network at the fastest possible rate. Any phase $\langle j+1, j+2 \rangle$ of this kind 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 we denote the phase as **a-split**(m, v_1, v_2, m').

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* BHS problem *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, 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*.*

Proof. Let us consider any exploration scheme $\mathcal{E}_{G,S,s}$ and the case $B = \{b\} \neq \emptyset$ (for any b). By hypothesis, we can remove b from G and have a partition of the nodes in two subsets: nodes becoming disconnected from s, and nodes maintaining the distance from s. The meeting point m at the end of the phase of $\mathcal{E}_{G,S,s}$ during which b is visited for the first time must be in this latter subset. Therefore, the path from m to s defined in $\mathcal{E}_{G,S,s}$ for the case $B = \emptyset$ cannot be shorter than the shortest path from m to s the surviving agent can follow in the case $B = \{b\}$. □

Corollary 1. *Let* (G, S, s) *be an input instance for the* BHS problem*. If* G *is a tree rooted at* s*, then the case* $B = \emptyset$ *yields the maximum execution time for any exploration scheme in* (G, S, s)*.*

Proof. This assertion straightforwardly follows from the property that in any tree there is always a unique path from any node to the root. □

Note that in our model we do not account for the time of computing the shortest path that the surviving agents are to follow to return to s at the end of the exploration. We assume that either this time is negligible or the whole set of required shortest paths is pre-computed and stored in the agents' memory.

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]. For a constant integer M, TSP(1,M) is defined in the following way.

TSP(1,M)

Instance : a pair (G, d), where $G = (V, E)$ is a complete graph (with $V = \{v_1, \ldots, v_n\}$) and $d : V^2 \to \{0, 1, \ldots, M\}$ is a distance function such that $d(v, v) = 0$, for each $v \in V$, and $d(v, u)$ is a positive integer between 1 and M (where M is a constant), for each $u, v \in V, u \neq v$. Function d is symmetric (i.e., $d(u, v) = d(v, u)$) and satisfies the triangle inequality (i.e., $d(i, j) + d(j, k) \geq d(i, k)$, $\forall i, j, k \in V$). For an edge (v, u) in G, we refer to number $d(v, u)$ as the length of this edge.

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. We assume that $\pi(n + 1) = \pi(1) = 1$.

Measure : the *length* (or *cost*) of the tour, i.e.,

$$\mathit{cost}(\tau) = \sum_{i=1}^{n} d(v_{\pi(i)}, v_{\pi(i+1)}).$$

Goal : minimization.

Lemma 4. *[7] It is* ***NP****-hard to approximate* TSP(1,8) *within* $1 + \varepsilon$ *for any* $\varepsilon < \frac{1}{388}$.

Our approach to prove the **APX**-hardness of the BHS problem is the following. We first provide a reduction from instances (G, d) of TSP(1,M) to instances (G', S, s) of the BHS problem. Given a solution τ for an instance (G, d) of the first problem, we construct a solution $\mathcal{E}^{\tau}_{G',S,s}$ for the corresponding instance (G', S, s) of the BHS problem. We show that $\mathit{cost}(\mathcal{E}^{\tau}_{G',S,s}) = 2\mathit{cost}(\tau)$ (Lemma 7). Then, by introducing the concept of *regular* exploration schemes, we show that given any exploration scheme $\mathcal{E}_{G',S,s}$, we can find a tour τ in G such that $\mathit{cost}(\mathcal{E}^{\tau}_{G',S,s}) \leq \mathit{cost}(\mathcal{E}_{G',S,s})$ (Lemma 11 and Lemma 12). Finally, we show that if, for any instance of the BHS problem constructed by reduction from an instance of TSP(1,M), we can approximate the optimal solution within a $(1+\varepsilon)$ factor, then we can approximate the optimal solution of the corresponding instance of TSP(1,M) within the same factor (Lemma 13).

Reduction from instances (G, d) of TSP(1,M) to instances (G', S, s) of the BHS problem. 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. Recall that $V = \{v_1, v_2, \ldots, v_n\}$. We begin the construction with $V' = \{v_1\}$, $S = \emptyset$ and define $s = v_1$ as the starting point of the BHS. 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 refer to node v_1 as the ISLAND I_1, and to each pair of nodes v'_j, v''_j (with $j = 2, \ldots, n$) as the ISLAND I_j. For each edge (v_i, v_j) in E of length $d(v_i, v_j)$, we add to G' a path of $2 \cdot d(v_i, v_j) - 1$ new nodes, and edges connecting one endpoint of this path to nodes v'_i and v''_i (or v_1, if $i = 1$) and the other to nodes v'_j and v''_j (or v_1 if $j = 1$). We denote such a path connecting island I_i with island I_j as BRIDGE $i \leftrightarrow j$. 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 $b_{i,j} \equiv b_{j,i}$). Observe that each bridge is composed of at least one (safe) node, and that $|V' \setminus S| = 2(n-1)$. An example of this reduction is presented in Figure 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 $2 \cdot d(v_i, v_j)$.*

Proof. By construction, bridge $i \leftrightarrow j$ is composed of $2 \cdot d(v_i, v_j) - 1$ nodes. Hence the length of the path from I_i to I_j which uses such a bridge is $2 \cdot d(v_i, v_j)$. Suppose, by contradiction, that there exists a path in G' from I_i to I_j of length less than $2 \cdot d(v_i, v_j)$. This path starts from I_i, visits some other islands (say $\langle I_{k_1}, \ldots, I_{k_k} \rangle$) and then ends in I_j. The length of such a path is $2\,[d(v_i, v_{k_1}) + d(v_{k_1}, v_{k_2}) + \cdots + d(v_{k_k}, v_j)]$. This would mean that $d(v_i, v_{k_1}) + d(v_{k_1}, v_{k_2}) + \cdots + d(v_{k_k}, v_j) < d(v_i, v_j)$. By the triangle inequality on the distances in G, this is a contradiction. □

The following lemma gives a useful property of the constructed instance (G', S, s) of the BHS problem.

Lemma 6. *For any exploration scheme for the constructed instance (G', S, s) of the* BHS problem*, the case $B = \emptyset$ yields the maximum execution time.*

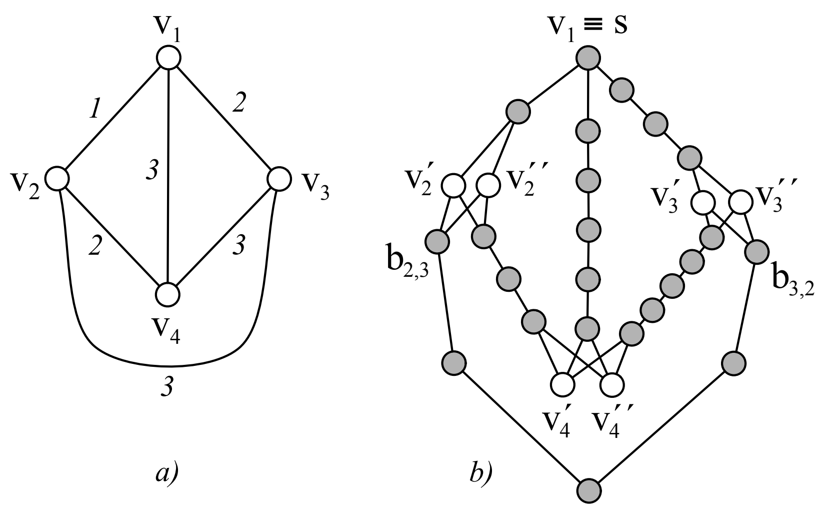


Fig. 1. An example of the reduction from an instance (G, d) of TSP(1,M) (in a)) to an instance (G', S, s) of BHS problem (in b)). The nodes in S are filled with gray color.

Proof. Let v_i' be any node in U. By removing v_i' from G', no node becomes disconnected from s. Moreover, the node v_i'' (the other unexplored node in the same island) is at the same distance as v_i' from s, and has exactly the same set of neighbors as v_i'. Therefore, each node in G' which has v_i' in its shortest path to s, can replace v_i' with v_i'' in the path and remain at the same distance from s. By Lemma 3 the assertion is proved. □

Having a mapping from the instances of TSP(1,M) to instances of the BHS problem, we define now a mapping from the solutions for an instance of the TSP(1,M) problem to solutions of the corresponding instance of the BHS problem. Given an instance (G, d) of TSP(1,M), a corresponding instance (G', S, s) of the BHS problem, and a tour τ in G, we define an exploration scheme on G' which explores the islands in G' in the order defined by τ. In the following definition we introduce a new term: **walk**. By **walk**(b) we mean that both agents, which are supposed to be currently in the same node w, move to b by following a shortest safe path from w to b. Observe that such a walk is not a complete phase (no new nodes are explored), but we use it as 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. Recall that we assume $\pi(n+1) = \pi(1) = 1$, and that node $b_{i,j}$ is the node adjacent to I_i on the bridge $i \leftrightarrow j$. A τ-BASED EXPLORATION SCHEME $\mathcal{E}^{\tau}_{G',S,s}$ on G' consists of the following sequence of steps:

1. **walk**($b_{1,\pi(2)}$);
2. for each $i = 2, \ldots, n$:
   (a) **walk**($b_{\pi(i),\pi(i-1)}$);
   (b) **a-split**($b_{\pi(i),\pi(i-1)}, v'_{\pi(i)}, v''_{\pi(i)}, b_{\pi(i),\pi(i+1)}$).

In other words, the two agents walk together along the bridges, then they separate to visit the two nodes of each unexplored island, and finally meet again on the first node of the next bridge. Given the tour τ in G, the τ-based exploration scheme $\mathcal{E}^{\tau}_{G',S,s}$ can be obviously constructed in linear time. The following lemma gives the cost of the BHS based on $\mathcal{E}^{\tau}_{G',S,s}$.

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 $\mathcal{E}^{\tau}_{G',S,s}$ satisfies* $\text{cost}(\mathcal{E}^{\tau}_{G',S,s}) = 2 \cdot l$.

Proof. By Lemma 6, we can compute *cost*($\mathcal{E}^{\tau}_{G',S,s}$) as the execution time of $\mathcal{E}^{\tau}_{G',S,s}$ in the case $B = \emptyset$. The *walk* in (1) requires 1 step. For the i-th iteration in (2) $(i = 2, \ldots, n)$:

- the *walk* in $(2.a)$ requires $2 \cdot d(v_{\pi(i-1)}, v_{\pi(i)}) - 2$ steps;
- the split defined in $(2.b)$ requires 2 steps.

The exploration scheme $\mathcal{E}^{\tau}_{G',S,s}$ ends in $b_{\pi(n),1}$, and hence the agents have to get back to s. By Lemma 5, the distance from $b_{\pi(n),1}$ to s is $2 \cdot d(v_{\pi(n)}, v_1) - 1$, therefore:

$$\textit{cost}(\mathcal{E}^{\tau}_{G',S,s}) = 1 + 2 \sum_{i=2}^{n} d(v_{\pi(i-1)}, v_{\pi(i)}) + 2 \cdot d(v_{\pi(n)}, v_1) - 1 = 2 \cdot l.$$

□

Corollary 2. *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. *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* $\text{cost}(\mathcal{E}^{*}_{G',S,s}) \leq 2 \cdot \text{cost}(\tau^*)$.

Proof. Lemma 7 implies

$$2\textit{cost}(\tau^*) = \textit{cost}(\mathcal{E}^{\tau^*}_{G',S,s}) \geq \textit{cost}(\mathcal{E}^{*}_{G',S,s}).$$

□

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 define a notion of equivalence between exploration schemes which is based on such an operation.

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., phase ϕ is replaced by phase $\phi' = (\mathbb{Y}_\phi, \mathbb{X}_\phi)$. 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.*

The following lemma is a direct consequence of Definition 1.

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,* $\text{cost}(\mathcal{E}'_{G,S,s}) = \text{cost}(\mathcal{E}_{G,S,s})$.

Corollary 3. *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;

2d-phase : if two nodes in two distinct islands are explored during ϕ;

1-phase : if only one node (of one island) is explored during ϕ.

Definition 2. *Given an exploration scheme $\mathcal{E}_{G',S,s}$, we define the* PHASE GRAPH *of $\mathcal{E}_{G',S,s}$ as 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 additional node which we call x. The following edges are added to $P(\mathcal{E}_{G',S,s})$:*

– *a directed edge $\langle v_i, x\rangle$ is added for each node in island I_i which is explored during a 1-phase by* Agent-1*;*
– *a directed edge $\langle x, v_i\rangle$ is added for each node in island I_i which is explored during a 1-phase by* 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 $P(\mathcal{E}_{G',S,s})$ other than node x has degree (the in-degree plus the out-degree) equal to* 2*.*

Proof. It follows from Definition 2 that, for any node v_i in $P(\mathcal{E}_{G',S,s})$, there is an outgoing edge for each node in I_i of G' which is explored by *Agent*-1, and there is an incoming edge for each node in I_i of G' which is explored by *Agent*-2. Since each island I_i ($i = 2, \ldots, n$) has two unexplored nodes, the statement follows. □

Thus, for the graph $P(\mathcal{E}_{G',S,s})$, all edges of the underlying undirected multigraph form edge-disjoint simple cycles. 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 $outdeg(v_i) = 1$.*

Proof. By Lemma 9, any node v_i in $P(\mathcal{E}_{G',S,s})$ has degree 2. Hence, three cases may occur:

1. $indeg(v_i) = 1$ and $outdeg(v_i) = 1$: in this case one node of island I_i is explored by *Agent*-1 (the outgoing edge) and the other one is explored by *Agent*-2 (the incoming edge). Therefore, the island is explored in the regular way.

2. $indeg(v_i) = 0$ and $outdeg(v_i) = 2$: in this case both nodes of I_i are explored by *Agent*-1; the island is not explored in the regular way.
3. $indeg(v_i) = 2$ and $outdeg(v_i) = 0$: in this case both nodes of I_i are explored by *Agent*-2; the island is not explored in the regular way.

$\square$

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

Proof. It suffices to prove that we can find in linear time a finite sequence of phase-swaps in $\mathcal{E}_{G',S,s}$, which transforms $\mathcal{E}_{G',S,s}$ into a regular exploration scheme. By Lemma 10, this means transforming $P(\mathcal{E}_{G',S,s})$ into a graph where, for each node v_i, $indeg(v_i) = 1$ and $outdeg(v_i) = 1$. We can observe that each phase-swap in $\mathcal{E}_{G',S,s}$ changes the orientation of the corresponding edge in $P(\mathcal{E}_{G',S,s})$. For each (undirected) cycle in $P(\mathcal{E}_{G',S,s})$, we change the orientation of some edges to obtain a directed cycle, and thus make regular the graph $P(\mathcal{E}_{G',S,s})$, and the corresponding exploration scheme. $\square$

Lemma 12. *Given an exploration scheme $\mathcal{E}_{G',S,s}$, we can find in linear time a tour τ on (G, d) such that, for the τ-based exploration scheme $\mathcal{E}^{\tau}_{G',S,s}$,* $\text{cost}(\mathcal{E}_{G',S,s}) \geq \text{cost}(\mathcal{E}^{\tau}_{G',S,s})$.

Proof. By Corollary 3 and Lemma 11, we can assume without loss of generality 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 $l = cost(\tau)$. We show that the τ-based exploration scheme $\mathcal{E}^{\tau}_{G',S,s}$ is such that $cost(\mathcal{E}_{G',S,s}) \geq cost(\mathcal{E}^{\tau}_{G',S,s})$. Consider the BHS based on $\mathcal{E}_{G',S,s}$ in the case when $B = \emptyset$. *Agent*-1 starts from s, visits islands in $I_{\mathbb{X}}$ and then returns to s. By Lemma 5, the length of this tour is at least

$$2 \cdot \left(d(v_1, v_{\pi(2)}) + \sum_{i=2}^{n-1} d(v_{\pi(i)}, v_{\pi(i+1)}) + d(v_{\pi(n)}, v_1) \right) = 2 \cdot cost(\tau) = 2 \cdot l\,.$$

(The path followed by *Agent*-1 may actually be longer than $2l$ if the agent waits in a node or visits the same node more than once.) Thus, $cost(\mathcal{E}_{G',S,s})$ is at least $2 \cdot l$. By Lemma 7, $cost(\mathcal{E}^{\tau}_{G',S,s}) = 2 \cdot l$. Therefore, $cost(\mathcal{E}_{G',S,s}) \geq cost(\mathcal{E}^{\tau}_{G',S,s})$. $\square$

A straightforward corollary of Lemma 12 is that, for any optimal exploration scheme, there exists a corresponding tour-based exploration scheme having the same cost.

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

Proof. Suppose that, given the instance (G', S, s), 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 a tour τ in G such that for the exploration scheme $\mathcal{E}^{\tau}_{G',S,s}$, $cost(\mathcal{E}^{\tau}_{G',S,s}) \leq cost(\mathcal{E}_{G',S,s}) \leq cost(\mathcal{E}^*_{G',S,s})(1+\varepsilon)$. Therefore:

$$
\begin{aligned}
2cost(\tau) &= cost(\mathcal{E}^{\tau}_{G',S,s}) \quad \text{[by Lemma 7]} \\
&\leq cost(\mathcal{E}^*_{G',S,s})(1+\varepsilon) \\
&\leq 2cost(\tau^*)(1+\varepsilon) \quad \text{[by Corollary 2]}\,.
\end{aligned}
$$

Hence, $cost(\tau) \leq cost(\tau^*)(1+\varepsilon)$. □

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

Theorem 1. *The* BHS 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 this problem as the rBHS problem). We show that the BHS problem with this restriction remains **APX**-hard. Note that the input of the rBHS problem is fully specified by providing a graph G and the starting node s. In this section, we will hence use the simpler notation $\mathcal{E}_{G,s}$ to refer to an exploration scheme and (G, s) to refer to an instance of the rBHS problem. We will prove **APX**-hardness of the rBHS problem by showing a reduction from TSP(1,2) which preserves the non-approximability. We first recall Lemma 6.3 from [1]:

Lemma 14. *[1] 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 $\varepsilon_0 = \frac{1}{1290}$.*

With a small abuse of notation we define the cost of a tour in G as the cost of the corresponding TSP tour in the complete graph $\overline{G}$. We show a polynomial-time reduction algorithm $\mathcal{A}$ from TSP(1,2) to the rBHS problem, which takes as input an instance G of TSP(1,2), computes an instance (G', s) of the rBHS problem, and has the following property.

Lemma 15. *Let $0 < \varepsilon < \frac{4}{7}\varepsilon_0$, let G be an n-node cubic graph (an instance of* TSP(1,2)*), and let (G', s) be the corresponding instance of the* rBHS problem *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 $0 < \varepsilon < \frac{4}{7}\varepsilon_0$ 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 is either equal to n or at least $n(1+\varepsilon_0)$, then we can decide which of these two cases happens, if we have a $(1+\varepsilon)$-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 approximate the* rBHS problem *within a factor $1+\varepsilon$ for any $\varepsilon < \frac{1}{2258}$.*

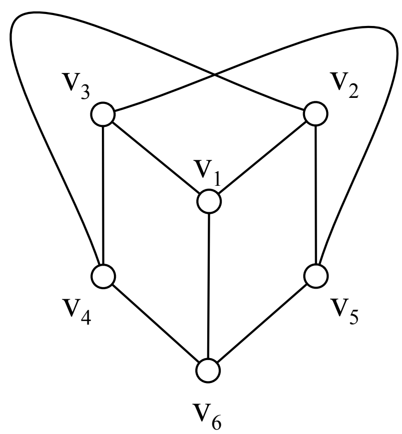
Description of the reduction algorithm $\mathcal{A}$. Let an n-node graph $G = (V, E)$ be the input instance of TSP(1,2). The construction of the instance (G', s) of the rBHS problem is similar to the construction presented in Section 3. The main differences are that here we do not add bridges corresponding to edges of weight 2 and that all nodes but the starting node s are initially unexplored. More precisely, the construction of (G', s) proceeds as follows. We add node v_1 to G' and make it the starting node ($s \equiv v_1$). For each node v_i in G, $2 \le i \le n$, 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). Note that s is adjacent to all bridge nodes and is not adjacent to any "island" nodes. Note also that the obtained graph G' is bipartite with nodes v_1 and v'_i and v''_i, $2 \le i \le n$, on one side of the partition and nodes $b_{i,j}$ and node b_s (if it exists) on the other side.

An example of this reduction is presented in Figure 2.

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 x, $odd(x)$ is equal to 1, if x 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 $\mathcal{E}^{\tau}_{G',s}$ 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 $\mathcal{E}^{\tau}_{G',s}$ 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 $I_{\pi(i)}$ 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 $I_{\pi(i)}$ otherwise.



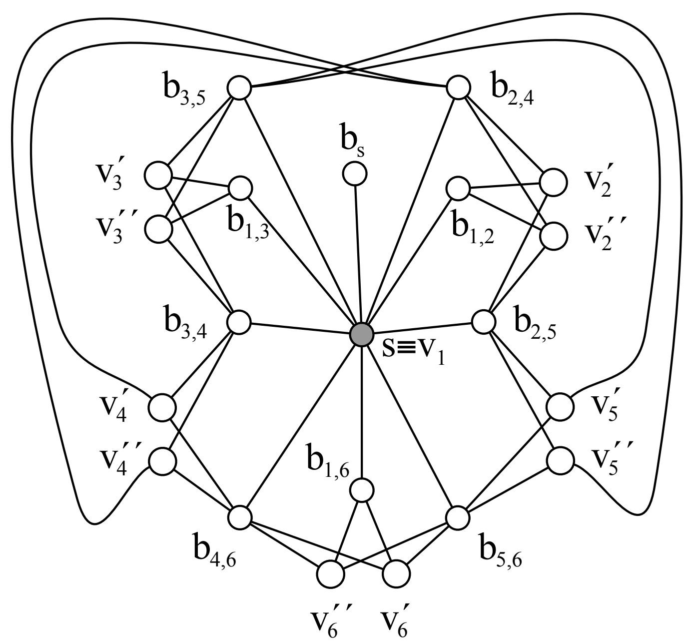


Fig. 2. An example of the reduction $\mathcal{A}$. from an instance G of TSP(1,2) (in a)) to an instance (G', s) of rBHS problem (in b)). Observe that the only explored node is s (filled with gray color), and that, since the number of edges in G is odd, an unexplored node b_s is added.

Note that 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 $v_{\pi(i-1)}, v_{\pi(i)}$ are adjacent in G, or 2, if nodes $v_{\pi(i-1)}, v_{\pi(i)}$ 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 $\mathcal{E}^{\tau}_{G',s}$ 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 $\mathcal{E}^{\tau}_{G',s}$ ends in a bridge node, which is adjacent to s. This is also an upper bound on the cost of the BHS based on $\mathcal{E}^{\tau}_{G',s}$, since Lemma 3 holds for (G', s). If the cost of tour τ is n, then $d = 0$ and the cost of $\mathcal{E}^{\tau}_{G',s}$ is at most $\frac{7}{2}n + 1$, so the first part of Lemma 15 holds.

To prove the second part of Lemma 15, we consider an arbitrary exploration scheme $\mathcal{E}_{G',s}$, and show that if the cost of this scheme is at most $(\frac{7}{2}n + 1)(1 + \varepsilon)$, then there is a tour in G of length less than $n(1+\varepsilon_0)$. By using a similar approach as the one described 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. We further assume, also by symmetry of the agents, that *Agent*-1 explores at least half of the bridge nodes. Let q_i denote the number of bridge nodes explored by *Agent*-1 between the explorations of node $v'_{\pi(i)}$ and node $v'_{\pi(i+1)}$, for $i = 2, \ldots, n-1$. Let q_1 and q_n denote the number of bridge nodes explored by *Agent*-1 before the exploration of node $v'_{\pi(2)}$, and after the exploration of node $v'_{\pi(n)}$, respectively. We have $\sum_{i=1}^{n} q_i \ge \lceil \frac{m}{2} \rceil$. *Agent*-1 needs at least $2q_1 + 1$ steps to reach node $v'_{\pi(2)}$. Then it needs at least $2q_i + 2$ steps to move from node $v'_{\pi(i)}$ to node $v'_{\pi(i+1)}$, for each $i = 2, \ldots, n-1$. And finally, it needs $2q_n + 1$ steps to reach the last meeting point. Thus the length of the exploration scheme $\mathcal{E}'_{G',s}$ is at least:

$$(2q_1 + 1) + \sum_{i=2}^{n-1}(2q_i + 2) + 2q_n + 1 \ge 2\left(\left\lceil \frac{m}{2} \right\rceil\right) + 2n - 2 = \frac{7}{2}n + odd(m) - 2. \quad (1)$$

We will show that for each index i, $2 \le i \le n-1$, such that $(v_{\pi(i)}, v_{\pi(i+1)})$ is not an edge in G, *Agent*-1 takes in fact at least $2q_i + 4$ steps to move from node $v'_{\pi(i)}$ to node $v'_{\pi(i+1)}$. This will imply that the length of the exploration scheme $\mathcal{E}'_{G',s}$ is at least

$$\frac{7}{2}n + odd(m) - 2 + 2(d-2) \tag{2}$$

where d is the number of edges in tour τ which are not in G.

Consider an index i ($2 \le i \le n-1$) such that $(v_{\pi(i)}, v_{\pi(i+1)})$ is not an edge in G. If $q_i = 0$, that is, if *Agent*-1 does not explore any bridge node between the explorations of nodes $v'_{\pi(i)}$ and $v'_{\pi(i+1)}$, then it needs at least 4 steps to move from node $v'_{\pi(i)}$ to node $v'_{\pi(i+1)}$ because the distance between these two nodes is 4.

If $q_i > 0$, then let $b_1, b_2, \ldots, b_{q_i}$ be the bridge nodes explored by *Agent*-1 between the explorations of nodes $v'_{\pi(i)}$ and $v'_{\pi(i+1)}$. *Agent*-1 visits node $v'_{\pi(i)}$ (for the first time), then it goes to a meeting point z (which cannot be b_1), and then to node b_1. This takes at least 3 steps because the length of a path from $v'_{\pi(i)}$ to b_1 containing at least one intermediate node (node z) is at least 3 as nodes $v'_{\pi(i)}$ and b_1 are on the opposite sides of the bipartite graph G'. Similarly, *Agent*-1 needs at least 3 steps to move from node b_{q_i} to node $v'_{\pi(i+1)}$. To move from node b_j to node b_{j+1}, for $j = 1, \ldots, q_i - 1$, *Agent*-1 needs at least 2 steps. Thus *Agent*-1 needs at least $3 + 2(q_i - 1) + 3 = 2q_i + 4$ steps to reach node $v'_{\pi(i+1)}$ from the first visit to node $v'_{\pi(i)}$.
The bound given by (2) on the length of the exploration scheme $\mathcal{E}'_{G',s}$ implies that

$$cost(\mathcal{E}'_{G',s}) \ge \frac{7}{2}n + 2d - 6.$$

This implies that if $cost(\mathcal{E}'_{G',s}) \le \left(\frac{7}{2}n + 1\right)(1+\varepsilon)$, then

$$d \le \frac{7}{4}\varepsilon n + \frac{\varepsilon}{2} + \frac{7}{2},$$

and

$$\begin{aligned} cost(\tau) &= n + d \\ &\le n + \frac{7}{4}\varepsilon n + \frac{\varepsilon}{2} + \frac{7}{2} \\ &= n(1+\varepsilon_0) - (\varepsilon_0 - \frac{7}{4}\varepsilon)n + \frac{\varepsilon}{2} + \frac{7}{2} \\ &< n(1+\varepsilon_0), \end{aligned}$$

provided that $\frac{7}{4}\varepsilon < \varepsilon_0$ and $n \ge n_0 = \lceil (\frac{\varepsilon}{2} + \frac{7}{2})/(\varepsilon_0 - \frac{7}{4}\varepsilon) \rceil$. □

5 A 6-approximation algorithm for the General BHS Problem

Let $G = (V, E)$ be the network to be explored, with the sets S and U defined as in Section 2. Let $u = |U|$. 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). An example of $\widehat{G}$ is presented in Figure 3. Note that weights in $\widehat{G}$ satisfy the triangle inequality. Let T be a

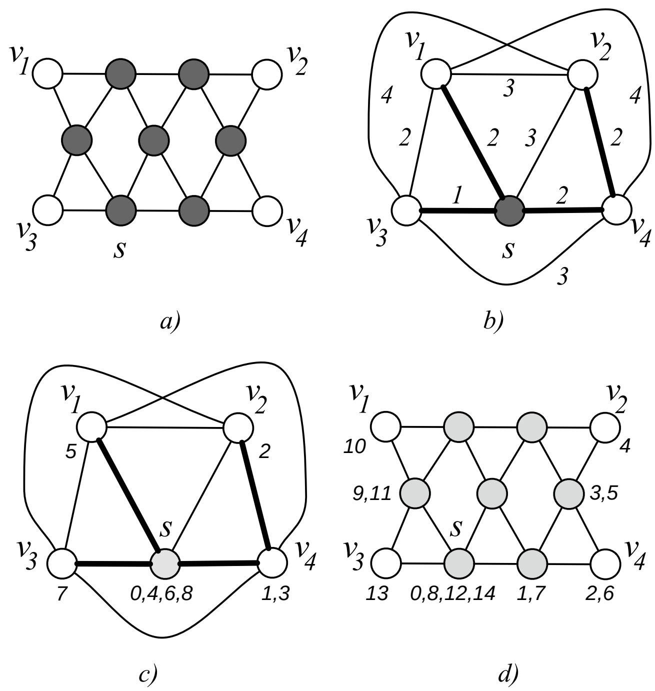


Fig. 3. $a)$ An instance (G, S, s) of the BHS problem. The gray nodes are the nodes in S. $b)$ The corresponding distance graph $\widehat{G}$. $c)$ The ordering in the Euler tour L_T of the nodes of $\widehat{G}$ (the numbers in italic). $d)$ The L_G sequence of the nodes of G. Note that some nodes of G may not be in L_G while some may occur more than once.

minimum spanning tree (MST) 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 $L_T = \langle z_0 = s, z_1, \ldots, z_{2u} = s\rangle$ be an Euler tour of T. Let $L_G = \langle w_0 = s, w_1, w_2, \ldots, w_q = s\rangle$ be the sequence obtained from L_T by inserting between each pair of consecutive nodes z_i and z_{i+1}, for $i = 0, 1, \ldots, z_{2u-1}$, the inner nodes of a shortest path in G between z_i and z_{i+1}. The length of L_G is twice the cost of T.

We now construct the exploration scheme $\mathcal{E}_{G,S,s} = (\mathbb{X}, \mathbb{Y})$ for G based on the walk L_G. Initially $\mathbb{X} = \mathbb{Y} = \langle s \rangle$. Then, for $i = 1, 2, \ldots, q$, the currently last node in $\mathbb{X}$ and $\mathbb{Y}$ is w_{i-1}, we consider node w_i in L_G, and extend the sequences $\mathbb{X}$ and $\mathbb{Y}$ in the following way. If node w_i is in S or has already occurred in L_G before, then append w_i to both $\mathbb{X}$ and $\mathbb{Y}$. Otherwise, append $\langle w_i, w_{i-1}, w_i \rangle$ to $\mathbb{X}$ and $\langle w_{i-1}, w_{i-1}, w_i \rangle$ to $\mathbb{Y}$. That is, if w_i is a new unexplored node, then *Agent*-1 visits w_i and goes back to node w_{i-1}, while *Agent*-2 waits for *Agent*-1 in node w_{i-1}. We call such two steps *probing*. The length of $\mathcal{E}_{G,S,s}$ is equal to $2\mathit{cost}(T) + 2u$.

Lemma 16. *The exploration scheme $\mathcal{E}_{G,S,s}$ is feasible and can be constructed in polynomial time. Moreover,* $\mathrm{cost}(\mathcal{E}_{G,S,s}) \leq 2\mathrm{cost}(T) + 2u$.

Proof. Constraints 1 and 2 can be easily checked by observing that sequence L_G (from which $\mathbb{X}$ and $\mathbb{Y}$ are derived) is a concatenation of paths in G, starting from s. All the nodes in U are in $\widehat{G}$, in L_T and thus in L_G; moreover the insertion of *probing* phases does not alter the set of visited nodes, hence the agents visit all the unexplored nodes (Constraint 3). Observe that the agents always move together along explored nodes except for probing. Thus in each phase *Agent*-1 visits exactly one unexplored node, while *Agent*-2 does not visit any unexplored node. This implies that Constraint 4 is also satisfied.

If $B = \emptyset$, then the agents spend $2u$ steps on probing and $2\mathit{cost}(T)$ steps on following the Euler tour L_T. If there is a black hole somewhere in the network, then the agents spend at most $2u$ steps on probing, and when *Agent*-2 finds out where the black hole is, it can return to node s by skipping some parts of the Euler tour L_T. Thus the execution time in this case is at most $2\mathit{cost}(T) + 2u$. Hence the cost of the exploration scheme $\mathcal{E}_{G,S,s}$ is $2\mathit{cost}(T) + 2u$.

Graph $\widehat{G}$ can be constructed by computing all-pairs shortest paths in G; by using the best known algorithm [14], this operation has cost $O(n^{\omega} \log n)$, where $O(n^{\omega})$ is the cost of a matrix product computation. This is the dominating cost of the whole algorithm, since the computation of the spanning tree T of $\widehat{G}$, as the computation of L_T and L_G, can be all performed in linear time. $\square$

Let us 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 a lower bound, the execution time of $\mathcal{E}^*_{G,S,s}$ in the case $B = \emptyset$. Let $L' = \langle x_k, \ldots, s\rangle$ 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. The length $|L''|$ of L'' is at most twice the execution time of $\mathcal{E}^*_{G,S,s}$ in the case $B = \emptyset$, since L'' is the concatenation of the paths the two agents follow during the exploration in this case; hence $2cost(\mathcal{E}^*_{G,S,s}) \geq |L''|$. 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:

$$L^* = \langle s\rangle \circ P(s, u_1\rangle \circ P(u_1, u_2\rangle \circ \cdots \circ P(u_u, s\rangle$$

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 $P(x, y\rangle$ denotes the shortest path from node x (excluded) to node y in G. Since weights in G satisfy the triangle inequality, the length of L^* is equal to the length of the minimum traveling salesman tour in $\widehat{G}$, which is, by a well-known relation, at least the cost of the minimum spanning tree T of $\widehat{G}$. Therefore, $|L^*| \geq cost(T)$, and

$$cost(\mathcal{E}^*_{G,S,s}) \geq \frac{cost(T)}{2}. \tag{3}$$

Moreover, since the agents cannot explore more than two nodes every two steps, the trivial lower bound still holds:

$$cost(\mathcal{E}^*_{G,S,s}) \geq u. \tag{4}$$

We compute the approximation ratio of the algorithm presented in this section, by choosing a suitable balance for Equations (3) and (4). Therefore:

$$\frac{cost(\mathcal{E}_{G,S,s})}{cost(\mathcal{E}^*_{G,S,s})} \leq \frac{2\,cost(T) + 2u}{\frac{2}{3}\,\frac{cost(T)}{2} + \frac{1}{3}u} = 6\,. \tag{5}$$

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

6 Conclusions

We showed that it is **NP**-hard to approximate within any factor less than $\frac{389}{388}$ the problem of computing the fastest exploration scheme for the BHS with two agents (the BHS problem). We have also shown that for the restricted version of this problem (the rBHS problem), when initially only the starting node is known to be safe, approximating within any factor less than $\frac{2259}{2258}$ is **NP**-hard. We have presented a polynomial-time 6-approximation algorithm for the BHS problem (while a polynomial-time $3\frac{3}{8}$-approximation algorithm for the rBHS problem was previously shown in [10]).

It seems difficult to reduce significantly the gap between the upper and lower bounds on the approximation ratios for the BHS problem and the rBHS problem. Since our lower bounds are based on reductions from 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 can show that the analysis of our approximation algorithm for the BHS problem is tight, i.e., the algorithm does not have a better approximation ratio than 6. However, we believe that one can find an approximation algorithm for the BHS problem with an approximation ratio better than 6. This might be achievable by considering the following two cases separately. If an MST T is such that $cost(T)/(2u)$ is not close to 1, say it is outside the range $[1-\delta, 1+\delta]$ for some small constant δ, then it can be shown that the ratio of costs given on the left-hand side of (5) is less than $6-\delta$. If $cost(T)/(2u)$ is within this range, then, using a similar analysis as in [10], one might try to show that there is some other tree which gives a better bound for the ratio of costs in (5) than 6. This approach would however lead most likely only to a small improvement, while requiring substantial expansion and refinement of technical details.

It would be interesting to investigate how one could model and analyze the more practical and more general case of multiple black holes search, possibly performed by more than two agents. It is interesting to observe that the assumption of having at most one black hole in the network does not make the algorithm presented in Section 5 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 simultaneously coexist in the network, the total number of agents needed is still related to the total number of black holes in the network.

Acknowledgements. The authors wish to thank the anonymous referees for their helpful suggestions.

References

[1] B. Csaba, M. Karpinski, and P. Krysta, Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems, Proc 13th Ann ACM-SIAM Symp Discr Algorithms (SODA '02), Philadelphia, PA, USA, 2002, pp. 74–83.

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

[3] J. Czyzowicz, D. Kowalski, E. Markou, and A. Pelc, Searching for a black hole in synchronous tree networks, Combin, Probability Comput 16 (2007), 595–619, Preliminary version under the title "Searching for a Black Hole In Tree Networks" in *Proc. 8th International Conference on Principles of Distributed Systems (OPODIS 2004)*, Springer LNCS vol. 3544, 2004, pp. 67-80.

[4] S. Dobrev, P. Flocchini, R. Kràlovic, G. Prencipe, P. Ruzicka, and N. Santoro, Black hole search in common interconnection networks, Networks 47 (2006), 61–71, Preliminary version under the title "Black Hole Search by Mobile Agents in Hypercubes and Related Networks" in *Proc. 6th Int. Conf. on Principles of Distributed Systems (OPODIS 2002)*, 2002, pp. 169–180.

[5] S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro, Searching for a black hole in arbitrary networks: Optimal mobile agents protocols, Distrib Comput 19 (2006), 1–18, Preliminary version in *Proc. 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, 2002, pp. 153–161.

[6] S. Dobrev, P. Flocchini, G. Prencipe, and N. Santoro, Mobile search for a black hole in an anonymous ring, Algorithmica 48 (2007), 67–90, Preliminary version in *Proc. 15th Int. Symposium on Distributed Computing (DISC 2001)*, Springer LNCS vol. 2180, 2001, pp. 166–179.

[7] L. Engebretsen and M. Karpinski, TSP with bounded metrics, J Comput System Sciences 72 (2006), 509–546.

[8] F. Hohl, Time limited black box security: Protecting mobile agents from malicious hosts, Proc Conference Mobile Agent Security, Vol. 1419 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 90–111.

[9] F. Hohl, A framework to protect mobile agents by using reference states, Proc 20th Int Conference Distrib Comput Syst (ICDCS '00), 2000, pp. 410–417.

[10] R. Klasing, E. Markou, T. Radzik, and F. Sarracco, Hardness and approximation results for black hole search in arbitrary graphs, Theoret Comput Sci 384 (2007), 201–221, Preliminary version in *Proc. 12th Int. Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, Springer LNCS vol. 3499, 2005, pp. 200–215.

[11] S. Ng and K. Cheung, Protecting mobile agents against malicious hosts by intention of spreading, Proc Int Conference Parallel Distrib Process Appl (PDPTA'99) Vol. II, 1999, pp. 725–729.

[12] T. Sander and C. Tschudin, Protecting mobile agents against malicious hosts, Proc Conference Mobile Agent Security, Vol. 1419 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 44–60.

[13] K. Schelderup and J. Ølnes, Mobile agent security – issues and directions, Proc 6th Int Conference Intelligence Services in Networks, Vol. 1597 of *Lecture Notes in Computer Science*, Springer, 1999, pp. 155–167.

[14] R. Seidel, On the all-pairs-shortest-path problem in unweighted undirected graphs, J Comput System Sciences 51 (1995), 400–403.