Edge Exploration of a Graph by Mobile Agent

Amit Kumar Dhar[1], Barun Gorain[1(✉)], Kaushik Mondal[2], Shaswati Patra[1], and Rishi Ranjan Singh[1]

[1] Indian Institute of Technology Bhilai, Sejbahar, Raipur, India
{amitkdhar,barun,shaswatip,rishi}@iitbhilai.ac.in
[2] Indian Institute of Information Technology Vadodara, Gandhinagar, India
kaushikmondal85@gmail.com

Abstract. In this paper, we study the problem of edge exploration of an n node graph by a mobile agent. The nodes of the graph are unlabeled, and the ports at a node of degree d are arbitrarily numbered $0, \ldots, d-1$. A mobile agent, starting from some node, has to visit all the edges of the graph and stop. The time of the exploration is the number of edges the agent traverses before it stops. The task of exploration can not be performed even for a class of cycles if no additional information, called advice, is provided to the agent a priori. Following the paradigm of algorithms with advice, this priori information is provided to the agent by an Oracle in the form of a binary string. The Oracle knows the graph, but does not have the knowledge of the starting point of the agent. In this paper, we consider the following two problems of edge exploration. The first problem is: "how fast is it possible to explore an n node graph regardless of the size of advice provided to the agent?"

We show a lower bound of $\Omega(n^{\frac{8}{3}})$ on exploration time to answer the above question. Next, we show the existence of an $O(n^3)$ time algorithm with $O(n \log n)$ advice. The second problem then asks the following question: "what is the smallest advice that needs to be provided to the agent in order to achieve time $O(n^3)$?" We show a lower bound $\Omega(n^\delta)$ on size of the advice, for any $\delta < \frac{1}{3}$, to answer the above question.

Keywords: Algorithm · Graph · Exploration · Mobile agent · Advice

1 Introduction

Exploration of a network by mobile agents is a well studied problem [28] which has various applications like treasure hunt, collecting data from some node in the network or samples from contaminated mines where corridors along with the crossings forms a virtual network. Many real life applications require collection of information from edges of a network as well. In such scenarios, edge explorations are essential to retrieve the required knowledge.

In this paper, we consider the edge exploration problem where a mobile agent, albeit with advice, aims to explore all the edges in a network and stop when

Y. Li et al. (Eds.): COCOA 2019, LNCS 11949, pp. 142–154, 2019.
https://doi.org/10.1007/978-3-030-36412-0_12

done. By *advice* we mean some prior information provided to the agent for the exploration, by an Oracle in the form of a binary string. The length of the string is called the size of advice. We analyze the lower bound on the exploration time with arbitrary size of advice before providing an efficient exploration algorithm.

The network is modeled as a simple connected undirected graph $G = (V, E)$ consisting of n nodes. Nodes are anonymous but all the edges associated to a node of degree d are arbitrarily numbered $0, 1, \cdots, d-1$ at the node. The mobile agent starts from an arbitrary node which we call as the starting node. Before starting the exploration, the agent knows the degree of the starting nodes. When the agent takes the port i at a node u and reaches node v, it learns the degree of v, and the port of the edge at v through which it reached v. The agent does not have the capability to mark any edge or node.

The time of the exploration is the number of edges the agent traverse before it stops. It is evident that some prior information needs to be provided to the agent in order to complete the task of edge exploration. For example, in the class of rings with ports numbered 0,1 in clockwise order at all the nodes, the agent can not learn the size of the ring only by exploring edges if no prior information is provided. Hence, it cannot distinguish between any two oriented rings of different size $k_1, k_2 \geq 3$. Therefore, any exploration algorithm that stops after t steps will fail to explore all the edges a ring with size $t+2$ or more. In this paper we study the problem of how much knowledge the agent needs to have a priori, in order to explore all the edges of a given graph in given time t by any deterministic algorithm.

Following the paradigm algorithm with advice [5,8], this prior information is provided to the agent by an Oracle. According to the literature [21], there are two kind of Oracles, instance Oracle and map Oracle. The entire instance of the exploration problem, i.e., the port-numbered map of the underlying graph and the starting node of the agent in this map is known by the instance Oracle, where as the map Oracle knows the port-numbered map of the underlying graph but does not know the starting node of the agent. In this work, we consider map Oracle.

Hence to prove possibility of such an exploration, we have to show existence of an exploration algorithm which uses advices of length at most x, one for each graph, provided by a map Oracle and explores all the graphs in $\mathscr{G}$ within time t starting from any node. On the other hand, to prove such an exploration is impossible in time t with advice of length x, we need to show existence of at least one graph and a starting point, such that no algorithm successfully explores all the edges of this graph within time t with any advice of length at most x.

It is natural to investigate the trade-off between exploration time and size of advice for edge exploration. In this paper, we provide two lower bound results, one on exploration time, and another on the size of advice for the edge exploration problem.

2 Contribution

Our main result consists of two lower bound results, one on exploration time and the other on size of advice. We prove that it is not possible to complete the task of edge exploration within time $o(n^{\frac{8}{3}})$, regardless of the size of advice. Next, we show the existence of an algorithm which works in time $O(n^3)$ with advice of size $O(n \log n)$. We also show that the minimum size of the advice necessary to explore all the edges of a graph in time $O(n^3)$ is $\Omega(n^\delta)$, for any $\delta < \frac{1}{3}$.

2.1 Related Work

Exploration of unknown environments by mobile agents is an extensively studied problem (cf. the survey [28]). We work on a model where the graph is undirected, nodes are anonymous and the mobile agent have some information a priori. Accordingly the mobile agent may traverse in any direction along an edge. The agent either has restricted tank [1] and needs to return to the base for refueling or already attached to the base with a cable of restricted length [9]. Usually in literature, most of the works analyze the time of completing the exploration by measuring the number of edges (counting multiple traversals) the agent traverses. This is considered as the efficiency measure of the algorithms.

Exploring any anonymous graph and to stop when done is impossible due to the anonymities of the nodes. As a solution, agents can have a finite number of *pebbles* [2,3] to drop on nodes which helps recognizing already visited ones or even put a stationary token at the starting node [6,27].

The problem of exploring anonymous graphs without node marking has been studied in several literature [7,16] where the termination condition after exploring all the edges is removed. Hence, in such variation of problems, the number of edge traversal becomes meaningless, instead, finding the minimum memory required for exploration appears to be the key.

For termination after successful exploration, further knowledge about the graph is essential, e.g., an upper bound on its size [6,29]. These information are usually known as advice and the approaches as *algorithms with advice*.

The paradigm of algorithm with advice is also extensively studied for other problems like graph coloring, broadcasting, leader election and topology recognitions where external information is provided to the nodes of a network or a external entity like mobile agent to perform the task efficiently [10–15,17–20,22–24,26].

In [13], the authors studied comparison of advice size in order to solve two information dissemination problems where the number of messages exchanged is linear. In [15], distributed construction of a minimum spanning tree in logarithmic time with constant size advice is discussed. In [10], authors shows that in order to broadcast a message from a source node in a radio network, 2-bits labeling, which also can be view as external advice, to the nodes are sufficient.

The algorithms with advice in the context of online algorithms is studied in [5,8,11]. In [8], online algorithm with advice is considered for a labeled weighted graph. In [25], authors did online exploration assuming upon visiting a node for the first time, the searcher learns all incident edges and their respective traversal costs. In weighted graphs, treasure hunt with advice, which is also a variation of exploration problem, was studied in [25]. Exploration with advice was studied for trees [14] and for general graphs in [21]. In [21] authors have described node exploration of an anonymous graph with advice. Two kind of Oracles are considered in this paper. Map Oracle, that knows the unlabeled graph, but does not know the starting node of the agent, and Instance Oracle, that knows the graph as well as the starting node of the agent. Trade-off between exploration time and size of advice is shown for both type of oracles.

3 Lower Bound on Exploration Time

In this section, we give a lower bound on time for edge exploration on a graph. More precisely, we show an exploration time of $\Omega(n^{\frac{8}{3}})$ for exploring all the edges of some graph regardless of the size of advice. To establish the lower bound, we construct an n node graph $\widehat{G}$ such that even if the agent is provided the map of the graph as advice, the time taken by the agent to explore all the edges is $\Omega(n^{\frac{8}{3}})$. The construction of the graph $\widehat{G}$ is given below.

Construction of $\widehat{G}$: We use the graphs discussed in [4] as building blocks to construct the graph for the lower bound result. For the sake of completeness, we discuss below the construction of the graphs (discussed in [4]).

Let $m > 0$ be an even positive integer. Let H be a m node regular graph with degree $\frac{m}{2}$. In this case we take H as a complete bipartite graph with the partition U and V of same size. Let T be any spanning tree of H with $E(T)$ being the spanning tree edges. Let S be the set of edges in H which are not in $E(T)$. Let $S = \{e_1, e_2, \cdots, e_s\}$ where $s = \frac{m^2}{4} - m + 1$. Let $X = x_1, x_2, \cdots, x_s$ be a binary string of length s, where not all x_i's are zero. A graph H_X is constructed from H using X as follows: take two copies of H, say H_1 and H_2 with the bi-partitions U_1, V_1 and U_2, V_2, respectively. For all i, $1 \leq i \leq s$, if $x_i = 1$, then delete the edges e_i from both H_1 and H_2 and cross two copies of e_i between the corresponding vertices of H_1 and H_2. More precisely, let $e_i = (u, v)$ be an edge of H with port numbers p at u and port number q at v. Let u_1, v_1 and u_2, v_2 be the nodes corresponding to u, v in H_1 and H_2, respectively. Delete (u_1, v_1) from H_1 and (u_2, v_2) from H_2, and connect two edges (u_1, v_2) and (u_2, v_1), (See Figs. 1 and 2). The port numbers of the newly added edges are p at both u_1, u_2 and q at both v_1 and v_2.

According to the result from [4], for each edge $e_i = (u, v) \in S$, there exists some sequence $X_i \in \{0, 1\}^s \setminus \{0\}^s$ such that if any exploration algorithm explores H starting from a node $v_0 \in H$ using a sequence of port numbers Q which traverse the edge e_i less than s times, then at least one of the edges (u_1, v_2) or (u_2, v_1) in H_{X_i} will remains unexplored while exploring H_{X_i} by Q. Let $\mathscr{H} = \{H_{X_i} : 1 \leq i \leq s\}$.

We use the above class of graphs $\mathscr{H}$ as building blocks to construct a graph for our lower bound. Our constructed graph will have the property that, if every

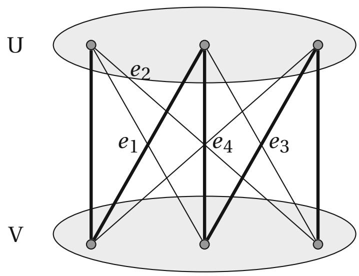

Fig. 1. Graph H with spanning tree edges shown in bold and non-tree edges labelled.

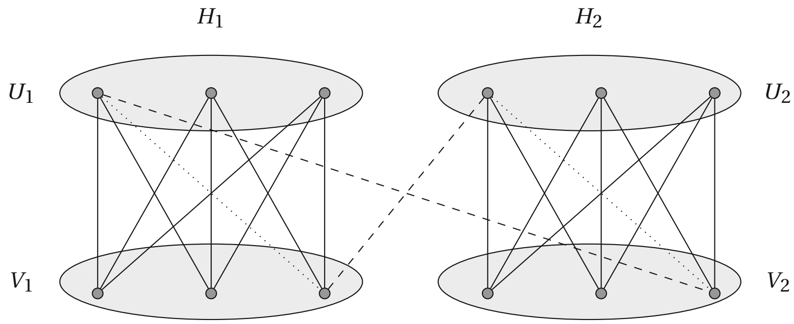

Fig. 2. The graph H_X, for $X = 0100$. The deleted edges are shown with dots and newly added edges with dash.

edge of the graph is not visited at least a fixed number of times, then some edge of the graph remains unexplored.

With the above discussions, we are ready to construct our final graph $\widehat{G}$. The high level idea of the construction is as follows. We will construct $\widehat{G}$ consisting of all the graphs from $\mathscr{H} = \{H_{X_i} : i \in [1, s]\}$ and systematically add some extra edges between every pair of H_{X_i} and H_{X_j}.

Vertices of $\widehat{G}$: The vertex set V of $\widehat{G}$ consists of all the vertices of all the graphs in $\mathscr{H}$. Hence, $|V| = (\frac{m^2}{4} - m + 1) * 2m = O(m^3)$. For the rest of the construction, we denote the independent sets of H_{X_i} as $H_{X_i}(U_1), H_{X_i}(V_1), H_{X_i}(U_2), H_{X_i}(V_2)$ (as shown in Fig. 3).

Edges of $\widehat{G}$: For every H_{X_i} with $i \in [1, s]$, we add the edges of H_{X_i} among the vertices of $H_{X_i}(U_1), H_{X_i}(V_1), H_{X_i}(U_2), H_{X_i}(V_2)$. Let these set of edges be denoted by $E_{i,i}$. For every ordered pair (i, j) such that $1 \leq i \neq j \leq s$, we add a set of edges to $\widehat{G}$, defined as $E_{i,j}$. For every edge (u, v) with $v \in H_{X_i}(V_k)$ we add the edge $(u, v') \in E_{i,j}$ with $v' \in H_{X_j}(V_k)$ for some $k \in \{1, 2\}$. The edge set E of $\widehat{G}$ is defined as $E = \bigcup_{i,j \in [1,s]} E_{i,j}$. Note that the subgraph of $\widehat{G}$ formed using the edge set $E_{i,j}$ and only the vertex sets $H_{X_i}(U_1), H_{X_i}(U_2), H_{X_j}(V_1), H_{X_j}(V_2)$

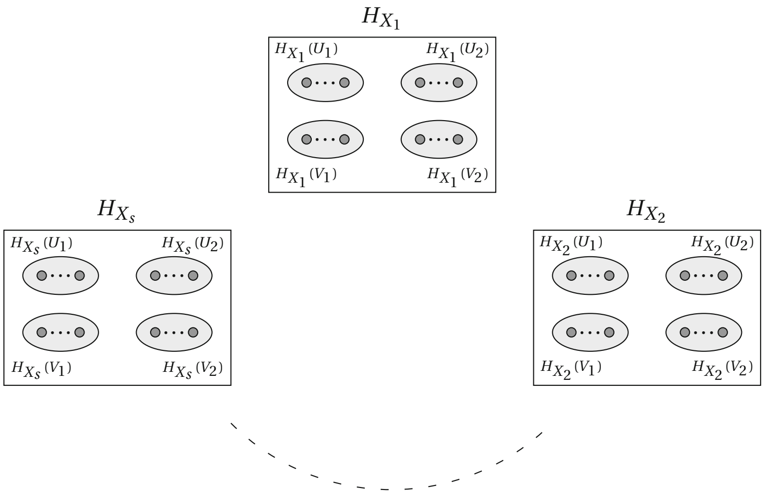

Fig. 3. Set of vertices in $\widehat{G}$

is isomorphic to H_{X_i}. In figure Fig. 4, we show the edges corresponding to $\bigcup_{j \in [1,s]} E_{1,j}$ that are added to $\widehat{G}$.

Port Numbers of $\widehat{G}$: For all ordered pair (i, j), $1 \leq i, j \leq s$, for each edge $e = (u, v) \in H_{X_i}$ with port numbers (p, q), the port number of the corresponding edge in $E_{i,j}$ is $(k\frac{m}{2} + p, k\frac{m}{2} + q)$, where $k = j - i \mod s$.

Let n be the number of nodes of $\widehat{G}$. Then $n = 2sm \geq \frac{m^3}{5}$.

Note that by construction of H, H_{X_i} and $\widehat{G}$ the degree of all the vertices in $\widehat{G}$ is exactly same. Suppose that the agent starts exploration from the node v_1 in H_{X_1}. Consider the following exploration sequence of the edges by the agent: visit all the ports from $0, 1, \cdots, \frac{m}{2} - 1$ attached at v_1 one by one, i.e, for every port i, visit the edge with port i, $0 \leq i \leq \frac{m}{2} - 1$ to reach a new vertex; come back to v_1 using the last visited edge. The above exploration sequence of edges will visit all the edges of attached with v_1 which are corresponding to the edges attached with v_1 in H_{X_i}. Now, if we change the staring node as the node v_1 in H_{X_2}, the same exploration sequence visits all the edges of attached with v_1 which are corresponding to the edges attached with v_1 in H_{X_2}. The construction of the graph $\widehat{G}$ guarantees that the agent cannot distinguish between this change in the starting node. In other words, any valid exploration algorithm for a particular graph $\widehat{G}$ gives a sequence of ports to be visited, that will remain same irrespective of the starting point of the algorithm as the agent cannot distinguish between different starting points. Therefore, any such exploration algorithm can be uniquely coded as a sequence of outgoing port numbers and the agent follows the ports according to this sequence in consecutive steps of exploration.

Let $\mathscr{B}$ be an exploration algorithm using which the agent explores all the edges of the graph starting from $v_0 \in H_{X_j}$, for any j, $1 \leq j \leq s$. Let U be the exploration sequence of outgoing port numbers corresponding to $\mathscr{B}$. Note that irrespective of the starting node, the sequence of port numbers U must visit all the edges of $\widehat{G}$, i.e., for every j, $1 \leq j \leq s$, if the agent starts from the node $v_0 \in H(X_j)$, it explores all the edges of $\widehat{G}$ following U. Let $U = q_1, q_2, \cdots, q_w$. Following lemma will be useful to prove our main lower bound result.

Lemma 1. *For any j, $1 \leq j \leq w$, if the port q_j visits an edge of $E_{x,y}$ when the starting node is $v_0 \in H_{X_i}$, for some i, $1 \leq i \leq s$, then q_j visits an edge of $E_{x+t-i \mod s, y+t-i \mod s}$ when the starting node is $v_0 \in H_{X_t}$.*

Proof. We will prove this lemma using induction. Suppose that the agent starts from the node $v_0 \in H_{X_i}$. According to the construction of $\widehat{G}$, the port q_1 must visit an edge of some $E_{i,y}$, where $1 \leq y \leq s$. According to the port number assignment of the edges of $\widehat{G}$, $(y-i)\frac{m}{2} \leq q_1 \leq (y-i+1)\frac{m}{2} - 1$. The ports from the node v_0 in $H(X_t)$ between $(y-i)\frac{m}{2}$ and $(y-i+1)\frac{m}{2} - 1$ are the part of edge set $E_{t,t+y-i \mod s}$. Therefore, if the agent starts from v_0 in H_{X_t}, then the port q_1 visits an edge of $E_{t,t+y-i \mod s}$, i.e., the edges of $E_{t+i-i \mod s, t+y-i \mod s}$. Hence the lemma is true for $j = 1$.

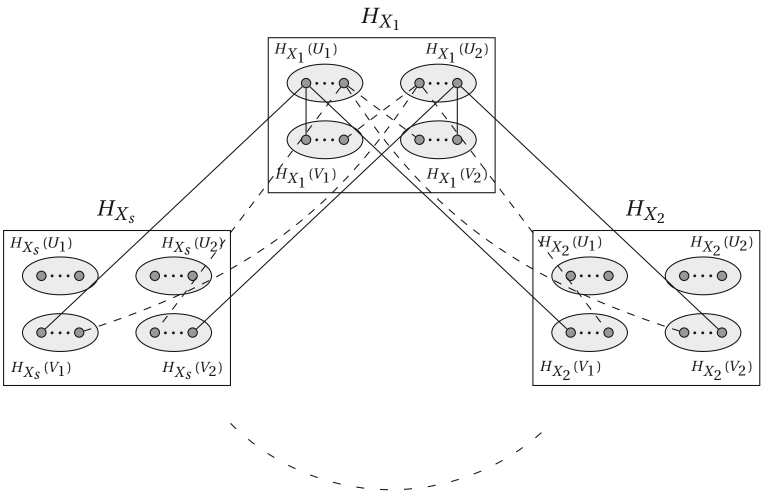

Fig. 4. A subset of edges of $\widehat{G}$ $(\bigcup_{j\in[1,s]} E_{1,j})$

Suppose that the lemma is true for any integer $j \leq f$. Let q_f be a port between $d\frac{m}{2}$ and $(d+1)\frac{m}{2} - 1$, for some d and q_f visits an edge of $E_{x,y}$ when the starting node is $v_0 \in H_{X_i}$. This implies that before taking the port q_f, then agent is in a vertex of H_{X_x} and after taking the port q_f, the agent reaches a vertex of H_{X_z}, where $z = x + d \mod s$. Therefore, using the induction hypothesis, when the starting node in $v_0 \in H_{X_t}$, before taking the port q_f, the agent is in a vertex of $H_{X_{x+t-i \mod s}}$ and after taking the port q_f, the agent reaches a vertex of $H_{X_{z'}}$, where $z' = x + t - i + d \mod s$.

Now, lets consider the port q_{f+1} when the agent starts from $v_0 \in H_{X_i}$. Let $d'\frac{m}{2} \leq q_{f+1} \leq (d'+1)\frac{m}{2}-1$, for some d'. Then q_{f+1} visits an edge of $E_{x',y'}$, where $x' = x + d \mod s$, $y' = x + d + d' \mod s$. Suppose that, when the starting node is $v_0 \in H_{X_t}$, the port q_{f+1} visits an edge of $E_{x'',y''}$. As, in this case, the agent is in a vertex of $H_{X_{z'}}$, where $z' = x + t - i + d \mod s$ and $d'\frac{m}{2} \leq q_{f+1} \leq (d'+1)\frac{m}{2}-1$, according to the port assignments of the edges of $\widehat{G}$, $x'' = x+t-i+d \mod s$ and $y'' = x + t - i + d + d' \mod s$, i.e., $x'' = x' + t - i \mod s$ and $y'' = y' + t - i \mod s$. This proves that the lemma is true for $j = f + 1$ and hence the lemma is proved by induction. □

With the above discussion, we are ready to prove our lower bound result.

Theorem 1. *Any exploration algorithm using any advice given by a Oracle must take time $\Omega(n^{\frac{8}{3}})$ time to explore all the edges of $\widehat{G}$.*

Proof. It is enough to prove the theorem for sufficiently large values of n, assuming that the advice given by the Oracle is $\widehat{G}$. Let $\mathscr{B}$ be an exploration algorithm using which the agent explores all the edges of the graph starting from $v_0 \in H_{X_j}$, for any i, $1 \leq j \leq s$. Let U be the exploration sequence of port numbers corresponding to $\mathscr{B}$. Consider the execution of the movement of the agent along the edges of $\widehat{G}$ when the starting node in $v_0 \in H(X_i)$, for some i. The sequence of port numbers U can be written as $U = B_1.(p_1).B_2.(p_2).\cdots B_k.(p_k).B_{k+1}$, where each B_ℓ is a sequence of port numbers corresponding to continuous movements of the mobile agent moving according to $\mathscr{B}$ (B_ℓ is a sequence involving zero or more port numbers, for each ℓ) and $p_1, p_2, \cdots, p_k$, are the ports that the agent takes to visit only the edges from $E_{x,y}$. Note that the sequence of port numbers $W = p_1, p_2, \cdots, p_k$ explores all the edges of $E_{x,y}$ in a scattered manner. We can convert W to a continuous sequence of port numbers W' such that W' explores all the edges in $E_{x,y}$ continuously, starting from $v_0 \in H_{X_x}$ as follows. Suppose that $u_1, u_2, \cdots, u_k$ be the vertices from where the agent takes the ports $p_1, p_2, \cdots, p_k$, respectively. Construct $W' = C_1p_1C_2p_2\cdots C_kp_k$, where C_1 is the sequence of port numbers corresponding to a shortest path from $v_0 \in H_{X_x}$ to u_1 and C_ℓ is the sequence of port numbers corresponding to a shortest path from the node the agent reached after taking the port p_ℓ (say, v), to $u_{\ell+1}$. Let W'' be the sequence of port number which is constructed from W' such that for each i, the value of the i−th port of W'' is assigned as the value of the i−th port of $W' \mod \frac{m}{2}$. Then W'' is an exploration sequence for H_{X_x}. Since degree of each node in the subgraph H_{X_x} induced by the edge set $E_{X,Y}$ is $\frac{m}{2}$, the length of each of the C_ℓ's are constant.

Hence $|W''| \in O(k)$. Also, since W'' is an exploration sequence of H_{X_x}, W'' must be an exploration sequence of H as well.

Claim: $k \in O(s^2)$.

We prove the above claim by showing that the exploration sequence W'' working on the graph H must visit every edge $e_1, \cdots e_s$ at least s times. Suppose that W'' visits some edge $e_{\ell'}$ at most $s-1$ times in H. Choose the starting node of the agent as the node v_0 of H_{X_t} where $\ell' = t + x - i \mod s$.

According to Lemma 1, using the sequence of port numbers of W'', the agent visits all the edges of $E_{\ell',\ell''}$, where $\ell'' = t + y - i \mod s$. Thus, W'' is an exploration sequence for $H_{X_{\ell'}}$ which visits the edge $e_{\ell'}$ in H at most $s-1$ times. Therefore by the result of [4], at least one of the edge in $H_{X_{\ell'}}$ is not visited by W''. This contradicts the fact that $\mathscr{B}$ is an algorithm that visits all the edges of $\widehat{G}$.

Therefore $|W''| \in O(s^2)$ and hence the claim is true.

Similarly, considering all $E_{x,y}$, for $1 \leq x, y \leq s$, it can be proved that all the edges of $\widehat{G}$ must be visited at least s times. Since, the total number of edges of the graph $\widehat{G}$ is $s^2 \frac{m^2}{4} \geq \frac{m^6}{20}$, therefore, $|U| \geq s\frac{m^6}{20} \geq \frac{m^8}{100} \in \Omega(n^{\frac{8}{3}})$. □

4 Exploration in $O(n^3)$ Time

In this section, we propose upper bound and lower bound results on size of advice in order to explore all the edges of the graph in time $O(n^3)$.

4.1 The Algorithm

Here we propose an algorithm using which the agent explores all the edges of an n node graph in time $O(n^3)$ with advice of size $O(n \log n)$.

Let G be any n node graph. The advice provided to the agent is a port numbered spanning tree of G. The spanning tree can be coded as a binary string of size $O(n \log n)$ [21]. The agent, after receiving the advice, will decode the spanning tree and explores all the edges of the graph as described below.

Algorithm: EdgeExploration

Step 1: After receiving the tree T as advice, the agent locally labels all the nodes of the spanning tree with unique labels from $\{1, 2 \cdots, n\}$. Then it computes n eulerian tours $\mathscr{E}_1, \cdots, \mathscr{E}_n$ from the spanning tree, where $\mathscr{E}_i$ is an eulerian tour that starts and ends at the vertex i. These tours are basically sequences of outgoing port numbers starting and ending at the same vertex i, for different values of i.

Step 2: For each $1 \leq i \leq n$, the agent start exploring according to the sequence of port numbers corresponding to $\mathscr{E}_i$ as follows. Let $p_1, p_2, \cdots, p_k$, be the sequence of port numbers corresponding to the tour $\mathscr{E}_i$. The agent visits all the port incident to the starting vertex and come back to it using the reverse ports from the adjacent node. Then it takes port p_1 and stores the port number of the other side of the edge in a stack. Next, it visits all the edges incident to the current node same as before, and come back to it using the reverse ports. Then it takes port p_2 and stores the port number of the other side of the edge in a stack. The agent continue visiting the edges in this way until it get stuck (this might happen in a case, when the agent is supposed to take the port p_i but the degree of the current node is less than $p_i + 1$) at some node or the tour

$\mathscr{E}_i$ is completed. At this point, it uses the port numbers which are stored in the stack to backtrack to the starting point.

Since the initial position of the agent is one of the nodes $1 \leq i \leq n$, it will succeed visiting all the edges for at least one $\mathscr{E}_i$. For each of such euler tour, the agent visit all the edges at each vertex, hence will take $O(n^2)$ time. Since there are n such euler tour, the time of exploration would be $O(n^3)$. Hence we will have the following theorem.

Theorem 2. *Algorithm* E*dgeExploration explores all the edges of the graph G in time $O(n^3)$ with advice of size $O(n \log n)$.*

4.2 Lower Bound

In this section, we prove that the size $\Omega(n^\delta)$ of advice is necessary, for any $\delta < \frac{1}{3}$ in order to perform edge exploration in $O(n^3)$ time. To prove this lower bound result, we construct a class of graphs $\mathscr{G}$ for which if the size of the advice given by the Oracle is $o(n^\delta)$, then there exist a graph in $\mathscr{G}$ for which the time of edge exploration is $\omega(n^3)$. The graphs in $\mathscr{G}$ are constructed in similar fashion as the graph $\widehat{G}$ in Sect. 3.

Let $\delta < \frac{1}{3}$ is a positive real constant. Then there exists a real constant ϵ, $0 < \epsilon < \frac{1}{2}$, such that $\delta > \frac{\epsilon}{1+\epsilon}$. Also, for $\epsilon < \frac{1}{2}$, there exists a real constant $c < \frac{1}{2}$ such that $\epsilon < \frac{(1-c)}{2}$.

Let $S = \{e_1, e_2, \cdots, e_s\}$ be the set of non spanning tree edges in H, where H is the complete bipartite graph of m (even) nodes and where each node has degree $\frac{m}{2}$. Let Z be any subset of S of size m^ϵ. Construct the graph G_Z as follows. Let $Z = \{e_{i_1}, e_{i_2}, \cdots, e_{i_p}\}$, where $p = m^\epsilon$. Construct G_Z in the same way as we have constructed $\widehat{G}$ (in Sect. 3) by replacing s with p and S with Z. In other words, take one copy of each $H(X_{i_j})$, for $1 \leq j \leq p$, and connect additional edges similarly as explained in Sect. 3 to construct sets of edges $E_{a,b}$, where $a, b \in \{i_1, \cdots i_p\}$. Note that each subset Z of S corresponds to a graph G_Z. There are $\binom{s}{p}$ different subsets of S and hence there are $\binom{s}{p}$ different graphs like G_Z can be constructed.

Let $\mathscr{G} = \{G_Z | Z \subset S\}$. Then $|\mathscr{G}| = \binom{\frac{m^2}{4}-m+1}{m^\epsilon} \geq \binom{\frac{m^2}{5}}{m^\epsilon} \geq (\frac{m^{2-\epsilon}}{5})^{m^\epsilon} \geq m^{(2-\frac{c}{2}-\epsilon)m^\epsilon}$, for large values of m. Let n be the number of nodes in each graph of $\mathscr{G}$. Then $n = 2m^{1+\epsilon}$. With this class of graphs, we are ready to prove our lower bound result.

Theorem 3. *For any $\delta < \frac{1}{3}$, any exploration algorithm using advice of size $o(n^\delta \log n)$ must take $\omega(n^3)$ time on some n node graph of the class $\mathscr{G}$ for arbitrarily large n.*

Proof. Suppose that there exists an algorithm $\mathscr{A}$, using which the agent explores all the edges of any graph in $\mathscr{G}$ using advice of size at most $\frac{c}{2}m^\epsilon \log m - 1$. There are at most $m^{\frac{c}{2}m^\epsilon}$ many different binary strings possible with length at most $\frac{c}{2}m^\epsilon \log m - 1$. Since $|\mathscr{G}| \geq m^{(2-\epsilon-\frac{c}{2})m^\epsilon}$, by Pigeon hole principle, for at least

$m^{(2-c-\epsilon)m^\epsilon}$ many graphs the agent must receive same advice. Suppose $\mathscr{G}' \subset \mathscr{G}$ be the set of graphs with same advice.

Let $F(\mathscr{G}') = \{\cup\{e_{i_1}, e_{i_2}, \cdots, e_{i_p}\}| \ Z = \{e_{i_1}, e_{i_2}, \cdots, e_{i_p}\} \ and \ G_Z \in \mathscr{G}'\}$. Intuitively, $F(\mathscr{G}')$ is the collection of all such edge e_{i_k} of S for which H_{i_k} is used in the construction of at least one graph in $\mathscr{G}'$.

Next, we claim that $|F(\mathscr{G}')| \geq |\mathscr{G}'|^{\frac{1}{m^\epsilon}}$. To prove this claim, suppose otherwise. That is, $|F(\mathscr{G}')| < |\mathscr{G}'|^{\frac{1}{m^\epsilon}}$. Note that each graph in $\mathscr{G}'$ is constructed using m^ϵ different $H_{X_{i_j}}$. Therefore, at most $\binom{|F(\mathscr{G}')|}{m^\epsilon}$ different graphs are possible in $\mathscr{G}'$. Hence, $|\mathscr{G}'| \leq \binom{|F(\mathscr{G}')|}{m^\epsilon} \leq |F(\mathscr{G}')|^{m^\epsilon} < |\mathscr{G}'|$, which is a contradiction. Therefore, $|F(\mathscr{G}')| \geq m^{(2-c-\epsilon)}$.

Let G be any graph in $\mathscr{G}'$. We consider the execution of the algorithm $\mathscr{A}$ where the starting node of the agent is v_0 of some $H_{X_{i_j}}$ in $\mathscr{G}'$.

Let U be the sequence of port numbers corresponding to $\mathscr{A}$. The sequence of port numbers U can be written as $U = B_1.(p_1).B_2.(p_2).\cdots.B_k.(p_k)$, where each B_i is a sequence of port numbers corresponding to continuous movements of the mobile agent moving according to $\mathscr{A}$ (B_i is a sequence involving zero or more port numbers, for each i) and $p_1, p_2, \cdots, p_k$, are the ports that the agent takes to visit only the edges from E_{i_x,i_y}. Consider the sequence of port numbers $W = p_1, p_2, \cdots, p_k$ (represents a discontinuous movement of the agent) and construct $W' = C_1p_1C_2p_2\cdots, C_kp_k$, where C_1 is the sequence of port numbers corresponding to a shortest path from $v_0 \in H_{i_x}$ to the end vertex of the port p_1 and C_i is the sequence of port numbers corresponding to a shortest path from the node the agent reached after taking the port p_i to the end vertex of the port p_{i+1}. Let W'' be the sequence of port number which is constructed from W' such that for each i, the value of the $i-$th port of W'' is assigned as the value of the $i-$th port of $W' \mod \frac{m}{2}$. We claim that the sequence of port numbers W'', when applied on H, starting from v_0 in H, must visit all the edges in $F(\mathscr{G}')$ at least s times. Otherwise, suppose W'' visits the edge $e_{i_\ell} \in F(\mathscr{G}')$ at most s times in H, starting from v_0 in H. Note that, since $e_{i_\ell} \in F(\mathscr{G}')$, there exists a graph $G_Z \in \mathscr{G}$ such that the edge $e_{i_\ell} \in Z$ and $Z \subset S$. Consider the exploration of the mobile agent in G_Z. Since the agent received same advice for all the graphs in $\mathscr{G}'$ and the graphs are indistinguishable for the agent, it explores all the graphs in $\mathscr{G}'$ using the same sequence of port numbers.

Therefore, in the exploration of G_Z, if the agent starts from v_0 in $H_{X_{i_t}}$, such that $i_\ell = i_x + i_t - i_j$, the sequence of port numbers W'' visits all the edges of $E_{i_\ell,i_{\ell'}}$, where $i_\ell = i_x + i_t - i_j$ (using Lemma 1). Since W'' visits the edge e_{i_ℓ} at most $s-1$ times, by the property of $E_{i_\ell,i_{\ell'}}$, the agent can not explore at least one edge in $H_{X_{i_\ell}}$, which is a contradiction that the algorithm $\mathscr{A}$ explores all the edges. This proves that $|W''| \geq s|F(\mathscr{G}')|$. Since m^ϵ copies of every graph $H(X_{i_t})$ is constructed in G_Z, for every edge $e_{i_t} \in Z$, by similar arguments we can prove that $|U| \geq s|F(\mathscr{G}')| \cdot m^\epsilon \cdot m^\epsilon \geq \frac{m^2}{5}m^{(2-\epsilon-c)} \cdot m^{2\epsilon} = \frac{m^{4+\epsilon-c}}{5}$. Since $\epsilon < \frac{1-c}{2}$, therefore $|U| \in \omega(m^{3+3\epsilon})$. Also, since $n = 2m^{1+\epsilon}$, therefore, $|U| \in \omega(n^3)$. Note that the size of the advice provided is $o(m^\epsilon \log m)$. Since $n = 2m^{1+\epsilon}$, therefore the size of advice is $o(n^{\frac{\epsilon}{1+\epsilon}} \log n)$, i.e, $o(n^{\delta} \log n)$. Therefore, with advice of size $o(n^{\delta} \log n)$, the time of exploration must be $\omega(n^3)$. □

5 Conclusion

The first lower bound results of $\Omega(n^{\frac{8}{3}})$ time for edge exploration and the proposed algorithm of $O(n^3)$ leaves a small gap of $O(n^{\frac{1}{3}})$ on exploration time. On the other hand, the second lower bound result on the size of advice, compared with the proposed algorithm also leaves a gap less than $O(n^{\frac{2}{3}+\epsilon})$, for any $\epsilon > 0$. Closing up these gaps between upper and lower bounds are natural open problems which can be addressed in the future. Another interesting problem is to study the edge exploration problem where the advice is provided by an instance Oracle.

References

1. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. Inf. Comput. **152**, 155–172 (1999)
2. Bender, M.A., Fernández, A., Ron, D., Sahai, A., Vadhan, S.P.: The power of a pebble: exploring and mapping directed graphs. Inf. Comput. **176**, 1–21 (2002)
3. Bender, M.A., Slonim, D.: The power of team exploration: two robots can learn unlabeled directed graphs. In: Proceedings of 35th Annual Symposium on Foundations of Computer Science (FOCS 1994), pp. 75–85 (1994)
4. Borodin, A., Ruzzo, W., Tompa, M.: Lower bounds on the length of universal traversal sequences. J. Comput. Syst. Sci. **45**, 180–203 (1992)
5. Boyar, J., Favrholdt, L.M., Kudahl, C., Larsen, K.S., Mikkelsen, J.W.: Online algorithms with advice: a survey. ACM Comput. Surv. **50**(2), 19:1–19:34 (2017)
6. Chalopin, J., Das, S., Kosowski, A.: Constructing a map of an anonymous graph: applications of universal sequences. In: Proceedings of 14th International Conference on Principles of Distributed Systems (OPODIS 2010), pp. 119–134 (2010)
7. Diks, K., Fraigniaud, P., Kranakis, E., Pelc, A.: Tree exploration with little memory. J. Algorithms **51**, 38–63 (2004)
8. Dobrev, S., Kralovic, R., Markou, E.: Online graph exploration with advice. In: Proceedings of 19th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2012), pp. 267–278 (2012)
9. Duncan, C.A., Kobourov, S.G., Anil Kumar, V.S.: Optimal constrained graph exploration. ACM Trans. Algorithms **2**, 380–402 (2006)
10. Ellen, F., Gorain, B., Miller, A., Pelc, A.: Constant-length labeling schemes for deterministic radio broadcast. In: Proceedings of 31st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 171–178 (2019)
11. Emek, Y., Fraigniaud, P., Korman, A., Rosen, A.: Online computation with advice. Theor. Comput. Sci. **412**, 2642–2656 (2011)
12. Fraigniaud, P., Gavoille, C., Ilcinkas, D., Pelc, A.: Distributed computing with advice: information sensitivity of graph coloring. Distrib. Comput. **21**, 395–403 (2009)
13. Fraigniaud, P., Ilcinkas, D., Pelc, A.: Communication algorithms with advice. J. Comput. Syst. Sci. **76**, 222–232 (2010). Oracle size: a new measure of difficulty for communication problems. In: Proceedings of 25th Annual ACM Symposium on Principles of Distributed Computing (PODC 2006), pp. 179–187

14. Fraigniaud, P., Ilcinkas, D., Pelc, A.: Tree exploration with advice. Inf. Comput. **206**, 1276–1287 (2008)
15. Fraigniaud, P., Korman, A., Lebhar, E.: Local MST computation with short advice. Theory Comput. Syst. **47**, 920–933 (2010)
16. Fraigniaud, P., Ilcinkas, D.: Directed graphs exploration with little memory. In: Proceedings of 21st Symposium on Theoretical Aspects of Computer Science (STACS 2004), pp. 246–257 (2004)
17. Fusco, E., Pelc, A.: Trade-offs between the size of advice and broadcasting time in trees. Algorithmica **60**, 719–734 (2011)
18. Fusco, E., Pelc, A., Petreschi, R.: Topology recognition with advice. Inf. Comput. **247**, 254–265 (2016)
19. Gavoille, C., Peleg, D., Pérennes, S., Raz, R.: Distance labeling in graphs. J. Algorithms **53**, 85–112 (2004)
20. Glacet, C., Miller, A., Pelc, A.: Time vs. information tradeoffs for leader election in anonymous trees. In: Proceedings of 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016), pp. 600–609 (2016)
21. Gorain, B., Pelc, A.: Deterministic graph exploration with advice. ACM Trans. Algorithms **15**, 8:1–8:17 (2018)
22. Gorain, B., Pelc, A.: Short labeling schemes for topology recognition in wireless tree networks. In: Proceedings of 24th International Colloquium Structural Information and Communication Complexity (SIROCCO), pp. 37–52 (2017)
23. Ilcinkas, D., Kowalski, D., Pelc, A.: Fast radio broadcasting with advice. Theor. Comput. Sci. **411**, 1544–1557 (2012)
24. Korman, A., Kutten, S., Peleg, D.: Proof labeling schemes. Distrib. Comput. **22**, 215–233 (2010)
25. Megow, N., Mehlhorn, K., Schweitzer, P.: Online graph exploration: new results on old and new algorithms. Theor. Comput. Sci. **463**, 62–72 (2012)
26. Nisse, N., Soguet, D.: Graph searching with advice. Theoret. Comput. Sci. **410**, 1307–1318 (2009)
27. Pelc, A., Tiane, A.: Efficient grid exploration with a stationary token. Int. J. Found. Comput. Sci. **25**, 247–262 (2014)
28. Rao, N.S.V., Kareti, S., Shi, W., Iyengar, S.S.: Robot navigation in unknown terrains: introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, July 1993
29. Reingold, O.: Undirected connectivity in log-space. J. ACM **55**, 17 (2008)