Fast Collaborative Graph Exploration*

Dariusz Dereniowski[1], Yann Disser[2], Adrian Kosowski[3], Dominik Pająk[3], and Przemysław Uznański[3]

[1] Gdańsk University of Technology, Poland
[2] TU Berlin, Germany
[3] CEPAGE project, Inria Bordeaux Sud-Ouest, France

Abstract. We study the following scenario of online graph exploration. A team of k agents is initially located at a distinguished vertex r of an undirected graph. At every time step, each agent can traverse an edge of the graph. All vertices have unique identifiers, and upon entering a vertex, an agent obtains the list of identifiers of all its neighbors. We ask how many time steps are required to complete exploration, i.e., to make sure that every vertex has been visited by some agent.

We consider two communication models: one in which all agents have global knowledge of the state of the exploration, and one in which agents may only exchange information when simultaneously located at the same vertex. As our main result, we provide the first strategy which performs exploration of a graph with n vertices at a distance of at most D from r in time $O(D)$, using a team of agents of polynomial size $k = Dn^{1+\epsilon} < n^{2+\epsilon}$, for any $\epsilon > 0$. Our strategy works in the local communication model, without knowledge of global parameters such as n or D.

We also obtain almost-tight bounds on the asymptotic relation between exploration time and team size, for large k. For any constant $c > 1$, we show that in the global communication model, a team of $k = Dn^c$ agents can always complete exploration in $D(1+\frac{1}{c-1}+o(1))$ time steps, whereas at least $D(1+\frac{1}{c}-o(1))$ steps are sometimes required. In the local communication model, $D(1+\frac{2}{c-1}+o(1))$ steps always suffice to complete exploration, and at least $D(1+\frac{2}{c}-o(1))$ steps are sometimes required. This shows a clear separation between the global and local communication models.

1 Introduction

Exploring an undirected graph-like environment is relatively straight-forward for a single agent. Assuming the agent is able to distinguish which neighboring vertices it has previously visited, there is no better systematic traversal strategy than a simple depth-first search of the graph, which takes $2(n-1)$ moves in total for a graph with n vertices. The situation becomes more interesting if multiple agents want to collectively explore the graph starting from a common location. If arbitrarily many agents may be used, then

* This work was initiated while A. Kosowski was visiting Y. Disser at ETH Zurich. Supported by ANR project DISPLEXITY and by NCN under contract DEC-2011/02/A/ST6/00201. The authors are grateful to Shantanu Das for valuable discussions and comments on the manuscript. The full version of this paper is available online at: `http://hal.inria.fr/hal-00802308`.

F.V. Fomin et al. (Eds.): ICALP 2013, Part II, LNCS 7966, pp. 520–532, 2013.

we can generously send n^D agents through the graph, where D is the distance from the starting vertex to the most distant vertex of the graph. At each step, we spread out the agents located at each node (almost) evenly among all the neighbors of the current vertex, and thus explore the graph in D steps.

While the cases with one agent and arbitrarily many agents are both easy to understand, it is much harder to analyze the spectrum in between these two extremes. Of course, we would like to explore graphs in as few steps as possible (i.e., close to D), while using a team of as few agents as possible. In this paper we study this trade-off between exploration time and team size. A trivial lower bound on the number of steps required for exploration with k agents is $\Omega(D + n/k)$: for example, in a tree, some agent has to reach the most distant node from r, and each edge of the tree has to be traversed by some agent. We look at the case of larger groups of agents, for which D is the dominant factor in this lower bound. This complements previous research on the topic for trees [6,8] and grids [17], which usually focused on the case of small groups of agents (when n/k is dominant).

Another important issue when considering collaborating agents concerns the model that is assumed for the communication between agents. We need to allow communication to a certain degree, as otherwise there is no benefit to using multiple agents for exploration [8]. We may, for example, allow agents to freely communicate with each other, independent of their whereabouts, or we may restrict the exchange of information to agents located at the same location. This paper also studies this tradeoff between global and local communication.

The Collaborative Online Graph Exploration Problem. We are given a graph $G = (V, E)$ rooted at some vertex r. The number of vertices of the graph is bounded by n. Initially, a set $\mathcal{A}$ of k agents is located at r. We assume that vertices have unique identifiers that admit a total ordering. In each step, an agent visiting vertex v receives a complete list of the identifiers of the nodes in $N(v)$, where $N(v)$ is the neighborhood of v. Time is discretized into steps, and in each step, an agent can either stay at its current vertex or slide along an edge to a neighboring vertex. Agents have unique identifiers, which allows agents located at the same node and having the same exploration history to differentiate their actions. We do not explicitly bound the memory resources of agents, enabling them in particular to construct a map of the previously visited subgraph, and to remember this information between time steps. An *exploration strategy* for G is a sequence of moves performed independently by the agents. A strategy explores the graph G in t time steps if for all $v \in V$ there exists time step $s \leq t$ and an agent $g \in \mathcal{A}$, such that g is located at v in step s. Our goal is to find an exploration strategy which minimizes the time it takes the explore a graph in the worst case, with respect to the shortest path distance D from r to the vertex furthest from r in the graph.

We distinguish between two communication models. In exploration *with global communication* we assume that, at the end of each step s, all agents have complete knowledge of the explored subgraph. In particular, in step s all agents know the number of edges incident to each vertex of the explored subgraph which lead to unexplored vertices, but they have no information on any subgraph consisting of unexplored vertices. In exploration *with local communication* two agents can exchange information only if they occupy the same vertex. Thus, each agent g has its own view on which vertices

Table 1. Our bounds for the time required to explore general graphs with using Dn^c agents. The same upper and lower bounds hold for trees. The lower bounds use graphs with $D = n^{o(1)}$.

Communication Model	*Upper bound*	*Lower bound*
Global communication:	$D \cdot (1 + \frac{1}{c-1} + o(1))$ Thm. 3	$D \cdot (1 + \frac{1}{c} - o(1))$ Thm. 5
Local communication :	$D \cdot (1 + \frac{2}{c-1} + o(1))$ Thm. 3	$D \cdot (1 + \frac{2}{c} - o(1))$ Thm. 5

were explored so far, constructed based only the knowledge that originates from the agent's own observations and from other agents that it has met.

Our results. Our main contribution is an exploration strategy for a team of polynomial size to explore graphs in an asymptotically optimal number of steps. More precisely, for any $\epsilon > 0$, the strategy can operate with $Dn^{1+\epsilon} < n^{2+\epsilon}$ agents and takes time $O(D)$. It works even under the local communication model and without prior knowledge of n or D.

We first restrict ourselves to the exploration of trees (Section 2). We show that with global communication trees can be explored in time $D \cdot (1 + 1/(c-1) + o(1))$ for any $c > 1$, using a team of Dn^c agents. Our approach can be adapted to show that with local communication trees can be explored in time $D \cdot (1 + 2/(c-1) + o(1))$ for any $c > 1$, using the same number of agents. We then carry the results for trees over to the exploration of general graphs (Section 3). We obtain precisely the same asymptotic bounds for the number of time steps needed to explore graphs with Dn^c agents as for the case of trees, under both communication models.

Finally, we provide lower bounds for collaborative graph exploration that almost match our positive results (Section 4). More precisely, we show that, in the worst case and for any $c > 1$, exploring a graph with Dn^c agents takes at least $D \cdot (1+1/c-o(1))$ time steps in the global communication model, and at least $D \cdot (1+2/c-o(1))$ time steps in the local communication model. Table 1 summarizes our upper and corresponding lower bounds.

Related Work. Collaborative online graph exploration has been intensively studied for the special case of trees. In [8], a strategy is given which explores any tree with a team of k agents in $O(D+n/\log k)$ time steps, using a communication model with whiteboards at each vertex that can be used to exchange information. This corresponds to a competitive ratio of $O(k/\log k)$ with respect to the optimum exploration time of $\Theta(D + n/k)$ when the graph is known. In [13] authors show that the competitive ratio of the strategy presented in [8] is precisely $k/\log k$. Another DFS-based algorithm, given in [2], has an exploration time of $O(n/k + D^{k-1})$ time steps, which provides an improvement only for graphs of small diameter and small teams of agents, $k = O(\log_D n)$. For a special subclass of trees called sparse trees, [6] introduces online strategies with a competitive ratio of $O(D^{1-1/p})$, where p is the density of the tree as defined in that work. The best currently known lower bound is much lower: in [7], it is shown that any deterministic exploration strategy with $k < \sqrt{n}$ has a competitive ratio of $\Omega(\log k/\log\log k)$, even in the global communication model. A stronger lower bound of $\Omega(k/\log k)$ holds for so-called greedy algorithms [13]. Both for deterministic and randomized strategies, the competitive ratio is known to be at least $2 - 1/k$, when $k < \sqrt{n}$ [8]. None of these lower bounds concern larger teams of agents. In [16] a lower bound of $\Omega(D^{1/(2c+1)})$ on competitive ratio is shown to hold for a team of $k = n^c$ agents, but this lower bound only concerns so-called rebalancing algorithms which keep all agents at the same height in the tree throughout the exploration process.

The same model for online exploration is studied in [17], where a strategy is proposed for exploring graphs which can be represented as a $D \times D$ grid with a certain number of disjoint rectangular holes. The authors show that such graphs can be explored with a team of k agents in time $O(D \log^2 D + n \log D/k)$, i.e., with a competitive ratio of $O(\log^2 D)$. By adapting the approach for trees from [7], they also show lower bounds on the competitive ratio in this class of graphs of $\Omega(\log k/\log\log k)$ for deterministic strategies and $\Omega(\sqrt{\log k}/\log\log k)$ for randomized strategies. These lower bounds also hold in the global communication model.

Collaborative exploration has also been studied with different optimization objectives. An exploration strategy for trees with global communication is given in [7], achieving a competitive ratio of $(4 - 2/k)$ for the objective of minimizing the maximum number of edges traversed by an agent. In [5] a corresponding lower bound of $3/2$ is provided.

Our problem can be seen as an online version of the k Traveling Salesmen Problem (k-TSP) [9]. Online variants of TSP (for a single agent) have been studied in various contexts. For example, the geometric setting of exploring grid graphs with and without holes is considered by [10,11,14,15,17], where a variety of competitive algorithms with constant competitive ratios is provided. A related setting is studied in [4], where an agent has to explore a graph while being attached to the starting point by a rope of restricted length. A similar setting is considered in [1], in which each agent has to return regularly to the starting point, for example for refueling. Online exploration of polygons is considered in [3,12].

2 Tree Exploration

We start our considerations by designing exploration strategies for the special case when the explored graph is a tree T rooted at a vertex r. For any exploration strategy, the set of all encountered vertices (i.e., all visited vertices and their neighbors) at the beginning of step $s = 1, 2, 3, \ldots$ forms a connected subtree of T, rooted at r and denoted by $T^{(s)}$. In particular, $T^{(1)}$ is the vertex r together with its children, which have not yet been visited. For $v \in V(T)$ we write $T^{(s)}(v)$ to denote the subtree of $T^{(s)}$ rooted at v. We denote by $L(T^{(s)}, v)$ the number of leaves of the tree $T^{(s)}(v)$. Note that $L(T^{(s)}, v) \leq L(T^{(s+1)}, v)$ because each leaf in $T^{(s)}(v)$ is either a leaf of the tree $T^{(s+1)}$ or the root of a subtree containing at least one vertex. If v is an unencountered vertex at the beginning of step s, i.e., its parent was not yet visited, we define $L(T^{(s)}, v) = 1$.

2.1 Tree Exploration with Global Communication

We are ready to give the procedure TEG (*Tree Exploration with Global Communication*). The pseudocode uses the command "move$^{(s)}$", describing the move to be performed by each agent, specifying the destination at which the agent appears at the start of time step $s+1$. Since the agents can communicate globally, the procedure can centrally coordinate the movements of each agent. For simplicity we assume that x agents spawn in r in each time step, for some given value of x. Then, the total number of agents used after l steps is simply lx.

Procedure TEG (tree T with root r, integer x) **at time step** s:
  Place x new agents at r.
  for each $v \in V(T^{(s)})$ which is not a leaf **do**: { determine moves of the agents located at v }
    Let $\mathcal{A}_v^{(s)}$ be the set of agents currently located at v.
    Denote by $v_1, v_2, \ldots, v_d$ the set of children of v.
    Let $i^* := \arg\max_i\{L(T^{(s)}, v_i)\}$. { v_{i^*} is the child of v with the largest value of L }
    Partition $\mathcal{A}_v^{(s)}$ into disjoint sets $\mathcal{A}_{v_1}, \mathcal{A}_{v_2}, \ldots, \mathcal{A}_{v_d}$, such that:
    (i) $|\mathcal{A}_{v_i}| = \left\lfloor \frac{|\mathcal{A}_v^{(s)}|\cdot L(T^{(s)},v_i)}{L(T^{(s)},v)} \right\rfloor$, for $i \in \{1,2,\ldots,d\} \setminus \{i^*\}$,
    (ii) $|\mathcal{A}_{v_{i^*}}| = |\mathcal{A}_v^{(s)}| - \sum_{i\in\{1,2,\ldots,d\}\setminus\{i^*\}} |\mathcal{A}_{v_i}|$.

    for each $i \in \{1,2,\ldots,d\}$ **do for each** agent $g \in \mathcal{A}_{v_i}$ **do move**$^{(s)}$ g to vertex v_i.
  end for
end procedure TEG.

The following lemma provides a characterization of the tradeoff between exploration time and the number of agents x released at every round in procedure TEG. In the following, all logarithms are with base 2 unless a different base is explicitly given.

Lemma 1. *In the global communication model, procedure* TEG *with parameter x explores any rooted tree T in at most $D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2\log x)})$ time steps, for $x > 6(n \log n + 1)$.*

Proof. Fix any leaf f of the tree T. We want to prove that procedure TEG visits the leaf f after at most $D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2\log x)})$ time steps. Take the path $\mathcal{F} = (f_0, f_1, f_2, \ldots, f_{D_f})$ from r to f in T, where $r = f_0$,$f = f_{D_f}$, and $D_f \le D$. We define the *wave* of agents w_s starting from r at time s and traversing the path $\mathcal{F}$ as the maximum sequence of the non-empty sets of agents which leave the root in step s and traverse edges of $\mathcal{F}$ in successive time steps, i.e., $w_s = (\mathcal{A}_{f_0}^{(s)}, \mathcal{A}_{f_1}^{(s+1)}, \ldots)$, where we use the notation from procedure TEG. The size of wave w_s in step $s+t$ is defined to be $|\mathcal{A}_{f_t}^{(s)}|$, i.e., the number of exploring agents located at vertex f_t at the beginning of time step $s+t$; initially, every wave has size $|\mathcal{A}_{f_0}^{(s)}| = x$. Note that each agent in $\mathcal{A}_{f_i}^{(s+i)}$, $0 \le i < D_f$, is located at r at the start of time step s. We denote the number of leaves in the subtree of $T^{(i)}$ rooted at f_j by $\lambda_j^{(i)} = L(T^{(i)}, f_j)$. Recall that if f_j is not yet discovered in step i, by definition of the function L, we have $\lambda_j^{(i)} = 1$. In general, $1 \le \lambda_j^{(i)} \le n$. We define

$$\alpha_i = \frac{x}{2} \frac{\lambda_1^{(i)}}{\lambda_0^{(i)}} \frac{\lambda_2^{(i+1)}}{\lambda_1^{(i+1)}} \cdots \frac{\lambda_{D_f}^{(i+D_f-1)}}{\lambda_{D_f-1}^{(i+D_f-1)}},$$

and define α_i^* as the number of agents of the i-th wave that reach the leaf f, i.e., the size of the i-th wave in step $i + D_f$. If $\alpha_1^* = \alpha_2^* = \cdots = \alpha_{i-1}^* = 0$ and $\alpha_i^* \geq 1$ for some time step i, then we say that leaf f is explored by the i-th wave. Before we proceed with the analysis, we show the following auxiliary claim.

Claim (*). Let i be a time step for which $\alpha_i \geq \log x$. Then, $\alpha_i^* \geq \alpha_i$, and thus α_i is a lower bound on the number of agents reaching f in step $i + D_f$.

Proof (of the claim). We define $c_j = \lambda_{j+1}^{(i+j)}/\lambda_j^{(i+j)}$ for $j = 0, \ldots, D_f - 1$. For $i \geq 1$ we have $\alpha_i = x/2 \prod_{j=0}^{D_f-1} c_j$. Since $c_j \leq 1$ for all j and since $\alpha_i \geq \log x$, there exist at most $\log x$ different j such that $c_j \leq 1/2$. Denote the set of all such j by $\mathcal{J}$, with $|\mathcal{J}| \leq \log x$. Also, denote the size of wave w_i in step $i + s$ by a_s (for $s = 0, 1, 2, \ldots$), in particular $a_0 = x$.

Consider some index s for which $c_s > 1/2$. We have $\lambda_{s+1}^{(i+s)}/\lambda_s^{(i+s)} > 1/2$, thus more than half of all leaves of the tree $T^{(i+s)}(f_s)$ also belong to the tree $T^{(i+s)}(f_{s+1})$. But then, in time step $i + s + 1$, agents are sent from f_s to f_{s+1} according to the definition in expression (ii) in procedure TEG. Thus, we can lower-bound the size of wave w_i in step $i + s + 1$ by $a_{s+1} \geq a_s c_s$. Otherwise, if $c_s \leq 1/2$ (i.e., if $s \in \mathcal{J}$), then agents are sent according the definition in expression (i) in procedure TEG, and hence $a_{s+1} \geq \lfloor a_s c_s \rfloor$. Note that these bounds also hold if there are no agents left in the wave, i.e., $a_s = a_{s+1} = 0$. Thus, we have:

$$a_{s+1} \geq a_s c_s - \delta_s, \quad \text{where } \delta_s = \begin{cases} 1, & \text{if } s \in \mathcal{J}, \\ 0, & \text{otherwise.} \end{cases}$$

In this way we expand the expression for $\alpha_i^* = a_{D_f}$:

$$\alpha_i^* = a_{D_f} \geq a_{D_f-1}c_{D_f-1} - \delta_{D_f-1} \geq \ldots \geq (\ldots((a_0c_0 - \delta_0)c_1 - \delta_1)c_2 - \ldots)c_{D_f-1} - \delta_{D_f-1} =$$
$$= x \prod_{j=0}^{D_f-1} c_j - \sum_{j=0}^{D_f-1} \left(\delta_j \prod_{p=j+1}^{D_f-1} c_j \right) \geq 2\alpha_i - \sum_{j=0}^{D_f-1} \delta_j \geq 2\alpha_i - |\mathcal{J}| \geq 2\alpha_i - \log x.$$

Since by assumption $\alpha_i \geq \log x$, we obtain $\alpha_i^* \geq 2\alpha_i - \log x \geq \alpha_i$, which completes the proof of the claim.

We now show that if the number of waves a in the execution of the procedure is sufficiently large, then there exists an index $i \leq a$, such that $\alpha_i \geq \log x$. Thus, taking into account Claim (*), leaf f is explored at the latest by the a-th wave.

Take a waves and consider the product $\prod_{i=1}^{a} \alpha_i$. Note that $\lambda_{D_f}^{(s)} = 1$ for every s. Thus, simplifying the product of all α_i by shortening repeating terms in numerators and denominators, and using $1 \leq \lambda_j^{(i)} \leq n$, we get

$$\prod_{i=1}^{a} \alpha_i = (\tfrac{x}{2})^a \prod_{i=1}^{a} \prod_{j=0}^{D_f-1} \frac{\lambda_{j+1}^{(i+j)}}{\lambda_j^{(i+j)}} = (\tfrac{x}{2})^a \frac{\prod_{i=1}^{a} \prod_{j=0}^{D_f-1} \lambda_{j+1}^{(i+j)}}{\prod_{i=1}^{a} \prod_{j=0}^{D_f-1} \lambda_j^{(i+j)}} = (\tfrac{x}{2})^a \frac{\prod_{i'=0}^{a-1} \prod_{j'=1}^{D_f} \lambda_{j'}^{(i'+j')}}{\prod_{i=1}^{a} \prod_{j=0}^{D_f-1} \lambda_j^{(i+j)}} =$$
$$= (\tfrac{x}{2})^a \frac{\left(\prod_{j'=1}^{D_f} \lambda_{j'}^{(j')}\right) \left(\prod_{i'=1}^{a-1} \prod_{j'=1}^{D_f-1} \lambda_{j'}^{(i'+j')}\right) \left(\prod_{i'=1}^{a-1} \lambda_{D_f}^{(i'+D_f)}\right)}{\left(\prod_{i=1}^{a} \lambda_0^{(i)}\right) \left(\prod_{i=1}^{a-1} \prod_{j=1}^{D_f-1} \lambda_j^{(i+j)}\right) \left(\prod_{j=1}^{D_f-1} \lambda_j^{(a+j)}\right)} \geq \frac{(x/2)^a}{n^a n^{D_f-1}} \geq \frac{(x/2)^a}{n^{a+D}}. \tag{1}$$

We want to find a, such that $\prod_{i=1}^{a} \alpha_i \geq (\log x)^a$. Taking into account (1), it is sufficient to find a satisfying

$$\frac{(x/2)^a}{n^{a+D}} \geq (\log x)^a,$$

which for sufficiently large x (we take $x > 6(n \log n + 1)$) can be equivalently transformed by taking logarithms and arithmetic to the form:

$$a \geq \frac{D}{\log_n x - 1 - \log_n(2 \log x)}.$$

Hence, for $a = \lceil \frac{D}{\log_n x - 1 - \log_n(2 \log x)} \rceil$, we have that there exists some i such that $\alpha_i \geq \log x$. For the same i we have $\alpha_i^* \geq \log x$, by Claim (*). Thus, a waves are sufficient to explore the path $\mathcal{F}$. This analysis can be done for any leaf f, thus it is enough to send a waves in order to explore the graph G. Considering that a wave w_i is completed by the end of step $D+i-1$, the exploration takes at most $D+a-1$ time steps in total. Thus, the exploration takes at most $D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2 \log x)})$ time steps. □

We remark that in the above Lemma, the total number of agents used throughout all steps of procedure TEG is $x \cdot D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2 \log x)})$. For any $c > 1$, by appropriately setting $x = \Theta(n^c)$, we directly obtain the following theorem.

Theorem 1. *For any fixed $c > 1$ and known n, the online tree exploration problem with global communication can be solved in at most $D \cdot \left(1 + \frac{1}{c-1} + o(1)\right)$ time steps using a team of $k \geq Dn^c$ agents.* □

2.2 Tree Exploration with Local Communication

In this section we propose a strategy for tree exploration under the local communication model. In the implementation of the algorithm we assume that whenever two agents meet, they exchange all information they possess about the tree. Thus, after the meeting, the knowledge about the explored vertices and their neighborhoods, is a union of the knowledge of the two agents before the meeting. Since agents exchange information only if they occupy the same vertex, at any time s, the explored tree $T^{(s)}$ may only partially be known to each agent, with different agents possibly knowing different subtrees of $T^{(s)}$.

In order to obtain a procedure for the local communication model, we modify procedure TEG from the previous section. Observe that in procedure TEG, agents never move towards the root of the tree, hence, in the local communication model, agents cannot exchange information with other agents located closer to the root. The new strategy is given by the procedure TEL (*Tree Exploration with Local Communication*).

In procedure TEL, all agents are associated with a state flag which may be set either to the value "exploring" or "notifying". Agents in the "exploring" state act similarly as in global exploration, with the requirement that they always move to a vertex in groups of 2 or more agents. Every time a group of "exploring" agents visits a new vertex, it detaches two of its agents, changes their state to "notifying", and sends them back along the path leading back to the root. These agents notify every agent they encounter on their way about the discovery of the new vertices. Although information about the discovery may be delayed, in every step s, all agents at vertex v know the entire subtree $T^{(s')}(v)$ which was explored until some previous time step $s' \leq s$. The state flag also has a third state, "discarded", which is assigned to agents no longer used in the exploration process.

The formulation of procedure TEL is not given from the perspective of individual agents, however, based on its description, the decision on what move to make in the current step can be made by each individual agent. The correctness of the definition of the procedure relies on the subsequent lemma, which guarantees that for a certain value s' the tree $T^{(s')}(v)$ is known to all agents at v.

Procedure TEL (tree T with root r, integer x) **at time step** s:
Place x new agents at r in state "exploring".
for each $v \in V(T^{(s)})$ which is not a leaf **do**: { determine moves of the agents located at v }
if $v \neq r$ **then for each** agent g at v in state "notifying" **do move**$^{(s)}$ g to the parent of v.
if v contains at least two agents in state "exploring" **and** agents at v do not have information of any agent which visited v before step s **then**:
{ send two new notifying agents back to the root from newly explored vertex v }
Select two agents g^*, g^{**} at v in state "exploring".
Change state to "notifying" for agents g^* and g^{**}.
move$^{(s)}$ g^* to the parent of v. { g^{**} will move to the parent one step later }
end if
Let $\mathcal{A}_v^{(s)}$ be the set of all remaining agents in state "exploring" located at v.
Denote by $v_1, v_2, \ldots, v_d$ all children of v, and by δ the distance from r to v.
$s' := \left\lfloor \frac{\delta+s}{2} \right\rfloor$. { s' is a time in the past such that $T^{(s')}(v)$ is known to the agents at v }
Let $i^* := \arg\max_i \{L(T^{(s')}, v_i)\}$. { v_{i^*} is the child of v with the largest value of L }
Partition $\mathcal{A}_v^{(s)}$ into disjoint sets $\mathcal{A}_{v_1}, \mathcal{A}_{v_2}, \ldots, \mathcal{A}_{v_d}$, such that:
(i) $|\mathcal{A}_{v_i}| = \left\lfloor \frac{|\mathcal{A}_v^{(s)}| \cdot L(T^{(s')}, v_i)}{L(T^{(s')}, v)} \right\rfloor$, for $i \in \{1, 2, \ldots, d\} \setminus \{i^*\}$,
(ii) $|\mathcal{A}_{v_{i^*}}| = |\mathcal{A}_v^{(s)}| - \sum_{i \in \{1,2,\ldots,d\} \setminus \{i^*\}} |\mathcal{A}_{v_i}|$.
for each $i \in \{1, \ldots, d\}$ **do if** $|\mathcal{A}_{v_i}| \geq 2$ **then for each** agent $g \in \mathcal{A}_{v_i}$ **do move**$^{(s)}$ g to v_i.
for each $i \in \{1, \ldots, d\}$ **do if** $|\mathcal{A}_{v_i}| = 1$ **then** change state to "discarded" for agent in $\mathcal{A}_{v_i}$.
end for
for each $v \in V(T^{(s)})$ which is a leaf **do move**$^{(s)}$ all agents located at v to the parent of v.
end procedure TEL.

Lemma 2. *Let T be a tree rooted at some vertex r and let v be a vertex with distance δ to r. After running procedure* TEL *until time step s, all agents which are located at vertex v at the start of time step s know the tree $T^{(s')}(v)$, for $s' = \left\lfloor \frac{\delta+s}{2} \right\rfloor$.*

(Some proofs are omitted from this extended abstract.)

Lemma 3. *In the local communication model, procedure* TEL *with parameter x explores any rooted tree T in at most $D \cdot (1 + \frac{2+1/\log n}{\log_n x - 1 - \log_n(4 \log x)})$ time steps, for $x > 17(n \log n + 1)$.*

Proof (sketch). As in the proof of Lemma 1, we consider any leaf f and the path $\mathcal{F} = (f_0, f_1, \ldots, f_{D_f})$ from r to f. As before, we denote the number of leaves in the subtree of $T^{(i)}$ rooted at f_j by $\lambda_j^{(i)} = L(T^{(i)}, f_j)$. Recall that if f_j is not yet discovered in step i, we have $L(T^{(i)}, f_j) = 1$. We adopt the definition of a wave from Lemma 1. We define the values α_i differently, however, to take into account the fact that the procedure relies on a delayed exploration tree, and that some waves lose agents as a result of deploying notifying agents: $\alpha_i = \frac{x}{4} \frac{\lambda_1^{(\lfloor \frac{i}{2} \rfloor)}}{\lambda_0^{(\lfloor \frac{i}{2} \rfloor)}} \frac{\lambda_2^{(\lfloor \frac{i}{2} \rfloor + 1)}}{\lambda_1^{(\lfloor \frac{i}{2} \rfloor + 1)}} \cdots \frac{\lambda_{D_f}^{(\lfloor \frac{i}{2} \rfloor + D_f - 1)}}{\lambda_{D_f - 1}^{(\lfloor \frac{i}{2} \rfloor + D_f - 1)}}$.

We call a wave that discovered at least $\lceil \log x \rceil$ new nodes (or equivalently, a wave whose agents were the first to visit at least $\lceil \log x \rceil$ nodes of the tree) a *discovery wave*. Thus, there are at most $\lfloor \frac{D_f}{\lceil \log x \rceil} \rfloor \leq \lfloor \frac{D}{\log x} \rfloor$ discovery waves along the considered path. Observe that if a wave is not a discovery wave, then the number of notifying agents it sends out is at most $2 \log x$.

We define by α_i^* the number of agents of the i-th wave that reach leaf f. We first prove that the following analogue of Claim (*) from the proof of Lemma 1 holds for non-discovery waves (we leave out the details from this extended abstract).

Claim (**). Let i be a time step for which w_i is not a discovery wave and $\alpha_i \geq \log x$. Then, $\alpha_i^* \geq \alpha_i$, and thus α_i is a lower bound on the number of agents reaching f in step $i + D_f$.

Finally, we prove that if the number of waves a in the execution of the procedure is sufficiently large, i.e. $a \geq D \cdot (\frac{2 + 1/\log n}{\log_n x - 1 - \log_n(4 \log x)})$, there exists an index $i \leq a$, such that wave w_i is not a discovery wave and $\alpha_i \geq \log x$. Exploration is then completed when the last wave reaches leaves, i.e. in $D + a - 1$ steps, which completes the proof. □

Acting as in the previous Subsection, from Lemma 3 we obtain a strategy for online exploration of trees in the model with local communication.

Theorem 2. *For any fixed $c > 1$, the online tree exploration problem can be solved in the model with local communication and knowledge of n using a team of $k \geq Dn^c$ agents in at most $D\left(1 + \frac{2}{c-1} + o(1)\right)$ time steps.* □

3 General Graph Exploration

In this section we develop strategies for exploration of general graphs, both with global communication and with local communication. These algorithms are obtained by modifying the tree-exploration procedures given in the previous section.

Given a graph $G = (V, E)$ with root vertex r, we call $P = (v_0, v_1, v_2, \ldots, v_m)$ with $r = v_0$, $v_i \in V$, and $\{v_i, v_{i+1}\} \in E$ a *walk* of length $\ell(P) = m$. Note that a walk may contain a vertex more than once. We introduce the notation $P[j]$ to denote v_j, i.e., the j-th vertex of P after the root, and $P[0, j]$ to denote the walk $(v_0, v_1, \ldots, v_j)$, for $j \leq m$. The last vertex of path P is denoted by $end(P) = P[\ell(P)]$. The concatenation of a vertex u to path P, where $u \in N(end(P))$ is defined as the path $P' \equiv P + u$ of length $\ell(P) + 1$ with $P'[0, \ell(P)] = P$ and $end(P') = u$.

Let $\mathcal{P}$ be the set of walks P in G having length $0 \leq \ell(P) < n$. We introduce a linear order on walks in $\mathcal{P}$ such that for two walks P_1 and P_2, we say that $P_1 < P_2$ if $\ell(P_1) < \ell(P_2)$, or $\ell(P_1) = \ell(P_2)$ and there exists an index $j < \ell(P_1)$ such that $P_1([0,j]) = P_2([0,j])$ and $P_1([j+1]) < P_2([j+1])$. The comparison of vertices from V is understood as comparison of their identifiers in G.

We now define the tree T with vertex set $\mathcal{P}$, root $(r) \in \mathcal{P}$, such that vertex P' is a child of vertex P if and only if $P' = P + u$, for some $u \in N(end(P))$. We first show that agents can simulate the exploration of T while in fact moving around graph G. Intuitively, while an agent is following a path from the root to the leaves of T, its location in T corresponds to the walk taken by this agent in G.

Lemma 4. *A team of agents can simulate the virtual exploration of tree T starting from root (r), while physically moving around graph G starting from vertex r. The simulation satisfies the following conditions:*

(1) *An agent virtually occupying a vertex P of T is physically located at a vertex $end(P)$ in G.*
(2) *Upon entering a vertex P of T in the virtual exploration, the agent obtains the identifiers of all children of P in T.*
(3) *A virtual move along an edge of T can be performed in a single time step, by moving the agent to an adjacent location in G.*
(4) *Agents occupying the same virtual location P in T can communicate locally, i.e., they are physically located at the same vertex of G.*

We remark that the number of vertices of tree T is exponential in n. Hence, our goal is to perform the simulation with only a subset of the vertices of T. For a vertex $v \in V$, let $P_{\min}(v) \in \mathcal{P}$ be the minimum (with respect to the linear order on $\mathcal{P}$) walk ending at v. We observe that, by property (1) in Lemma 4, if, for all $v \in V$, the vertex $P_{\min}(v)$ of T has been visited by at least one agent in the virtual exploration of T, the physical exploration of G is completed. We define $\mathcal{P}_{\min} = \{P_{\min}(v) : v \in V\}$, and show that all vertices of $\mathcal{P}_{\min}$ are visited relatively quickly if we employ the procedure `TEG` (or `TEL`) for T, subject to a simple modification. In the original algorithm, we divided the agents descending to the children of the vertex according to the number of leaves of the discovered subtrees. We introduce an alternate definition of the function $L(T^{(s)}, v)$, so as to take into account only the number of vertices in $T^{(s)}$ corresponding to walks which are smallest among all walks in $T^{(s)}$ sharing the same end-vertex.

Lemma 5. *Let $T^{(s)} \subseteq T$ be a subtree of T rooted at (r). For $P \in V(T^{(s)})$, let $L(T^{(s)}, P)$ be the number of vertices v of G, for which the subtree of $T^{(s)}$ rooted at P contains a vertex representing the smallest walk contained in $T^{(s)}$ which ends at v:*

$$L(T^{(s)}, P) = \left| V(T^{(s)}(P)) \cap \bigcup_{v \in V} \left\{ \min\{P' \in V(T^{(s)}) : end(P') = v\} \right\} \right|,$$

and for $P \in \mathcal{P} \setminus V(T^{(s)})$, let $L(T^{(s)}, P) = 1$. Subject to this definition of L, procedure `TEG` *with parameter $x > 6(n \log n + 1)$ (procedure* `TEL` *with parameter $x > 17(n \log n + 1)$) applied to tree T starting from root (r) visits all vertices from $\mathcal{P}_{\min}$*

within $D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2\log x)})$ *(respectively,* $D \cdot (1 + \frac{2+1/\log n}{\log_n x - 1 - \log_n(4\log x)})$*) time steps.*

Proof. The set $\mathcal{P}_{\min}$ spans a subtree $T_{\min} = T[\mathcal{P}_{\min}]$ in T, rooted at (r). We can perform an analysis analogous to that used in the Proofs of Lemmas 1 and 3, evaluating sizes of waves of agents along paths in the subtree $T_{\min}$. We observe that for any $P \in \mathcal{P}_{\min}$ which is not a leaf in $T_{\min}$, we always have $L(T^{(s)}, P) \geq 1$. Moreover, we have $L(T^{(s)}, P) \leq |V(T^{(s)}(P))|$, and so $L(T^{(s)}, P) \leq n$. Since these two bounds were the only required properties of the functions L in the Proofs of Lemmas 1 and 3, the analysis from these proofs applies within the tree $T_{\min}$ without any changes. It follows that each vertex of $\mathcal{P}_{\min}$ is reached by the exploration algorithm within $D \cdot (1 + \frac{1}{\log_n x - 1 - \log_n(2\log x)})$ time steps in case of global communication, and within $D \cdot (1 + \frac{2+1/\log n}{\log_n x - 1 - \log_n(4\log x)})$ time steps in case of local communication. □

We recall that by Lemma 4, one step of exploration of tree T can be simulated by a single step of an agent running on graph G. Thus, appropriately choosing $x = \Theta(n^c)$ in Lemma 5, we obtain our main theorem for general graphs.

Theorem 3. *For any* $c > 1$*, the online graph exploration problem with knowledge of* n *can be solved using a team of* $k \geq Dn^c$ *agents:*

- *in at most* $D \cdot \left(1 + \frac{1}{c-1} + o(1)\right)$ *time steps in the global communication model.*
- *in at most* $D \cdot \left(1 + \frac{2}{c-1} + o(1)\right)$ *time steps in the local communication model.*

For the case when we do not assume knowledge of (an upper bound on) n, we provide a variant of the above theorem which also completes exploration in $O(D)$ steps, with a slightly larger multiplicative constant.

Theorem 4. *For any* $c > 1$*, there exists an algorithm for the local communication model, which explores a rooted graph of unknown order* n *and unknown diameter* D *using a team of* k *agents, such that its exploration time is* $O(D)$ *if* $k \geq Dn^c$*.*

We remark that by choosing $x = \Theta(n \log n)$ in Lemma 2, we can also explore a graph using $k = \Theta(Dn \log n)$ agents in time $\Theta(D \log n)$, with local communication. This bound is the limit of our approach in terms of the smallest allowed team of agents.

4 Lower Bounds

In this section, we show lower bounds for exploration with Dn^c agents, complementary to the positive results given by Theorem 3. The graphs that produce the lower bound are a special class of trees. The same class of trees appeared in the lower bound from [8] for the competitive ratio of tree exploration algorithms with small teams of agents. In our scenario, we obtain different lower bounds depending on whether communication is local or global.

Theorem 5. *For all* $n > 1$ *and for every increasing function* f*, such that* $\log f(n) = o(\log n)$*, and every constant* $c > 0$*, there exists a family of trees* $\mathcal{T}_{n,D}$*, each with* n *vertices and height* $D = \Theta(f(n))$*, such that*

(i) for every exploration strategy with global communication that uses Dn^c agents there exists a tree in $\mathcal{T}_{n,D}$ such that number of time steps required for its exploration is at least $D\left(1 + \frac{1}{c} - o(1)\right)$,
(ii) for every exploration strategy with local communication that uses Dn^c agents there exists a tree in $\mathcal{T}_{n,D}$ such that number of time steps required for exploration is at least $D\left(1 + \frac{2}{c} - o(1)\right)$.

When looking at the problem of minimizing the size of the team of agents, our work (Theorem 4) shows that it is possible to achieve asymptotically-optimal online exploration time of $O(D)$ using a team of $k \leq Dn^{1+\epsilon}$ agents, for any $\epsilon > 0$. For graphs of small diameter, $D = n^{o(1)}$, we can thus explore the graph in $O(D)$ time steps using $k \leq n^{1+\epsilon}$ agents. This result almost matches the lower bound on team size of $k = \Omega(n^{1-o(1)})$ for the case of graphs of small diameter, which follows from the trivial lower bound $\Omega(D + n/k)$ on exploration time (cf. e.g. [8]). The question of establishing precisely what team size k is necessary and sufficient for performing exploration in $O(D)$ steps in a graph of larger diameter remains open.

References

1. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. Information and Computation 152(2), 155–172 (1999)
2. Brass, P., Cabrera-Mora, F., Gasparri, A., Xiao, J.: Multirobot tree and graph exploration. IEEE Transactions on Robotics 27(4), 707–717 (2011)
3. Czyzowicz, J., Ilcinkas, D., Labourel, A., Pelc, A.: Worst-case optimal exploration of terrains with obstacles. Information and Computation 225, 16–28 (2013)
4. Duncan, C.A., Kobourov, S.G., Kumar, V.S.A.: Optimal constrained graph exploration. ACM Transactions on Algorithms 2(3), 380–402 (2006)
5. Dynia, M., Korzeniowski, M., Schindelhauer, C.: Power-aware collective tree exploration. In: Grass, W., Sick, B., Waldschmidt, K. (eds.) ARCS 2006. LNCS, vol. 3894, pp. 341–351. Springer, Heidelberg (2006)
6. Dynia, M., Kutyłowski, J., Meyer auf der Heide, F., Schindelhauer, C.: Smart robot teams exploring sparse trees. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 327–338. Springer, Heidelberg (2006)
7. Dynia, M., Łopuszański, J., Schindelhauer, C.: Why robots need maps. In: Prencipe, G., Zaks, S. (eds.) SIROCCO 2007. LNCS, vol. 4474, pp. 41–50. Springer, Heidelberg (2007)
8. Fraigniaud, P., Gąsieniec, L., Kowalski, D.R., Pelc, A.: Collective tree exploration. Networks 48(3), 166–177 (2006)
9. Frederickson, G.N., Hecht, M.S., Kim, C.E.: Approximation algorithms for some routing problems. SIAM Journal on Computing 7(2), 178–193 (1978)
10. Gabriely, Y., Rimon, E.: Competitive on-line coverage of grid environments by a mobile robot. Computational Geometry 24(3), 197–224 (2003)
11. Herrmann, D., Kamphans, T., Langetepe, E.: Exploring simple triangular and hexagonal grid polygons online. CoRR, abs/1012.5253 (2010)
12. Higashikawa, Y., Katoh, N.: Online exploration of all vertices in a simple polygon. In: Proc. 6th Frontiers in Algorithmics Workshop and the 8th Int. Conf. on Algorithmic Aspects of Information and Management (FAW-AAIM), pp. 315–326 (2012)
13. Higashikawa, Y., Katoh, N., Langerman, S., Tanigawa, S.-I.: Online graph exploration algorithms for cycles and trees by multiple searchers. Journal of Combinatorial Optimization (2013)

14. Icking, C., Kamphans, T., Klein, R., Langetepe, E.: Exploring an unknown cellular environment. In: Proc. 16th European Workshop on Computational Geometry (EuroCG), pp. 140–143 (2000)
15. Kolenderska, A., Kosowski, A., Małafiejski, M., Żyliński, P.: An improved strategy for exploring a grid polygon. In: Kutten, S., Žerovnik, J. (eds.) SIROCCO 2009. LNCS, vol. 5869, pp. 222–236. Springer, Heidelberg (2010)
16. Łopuszański, J.: Tree exploration. Tech-report, Institute of Computer Science, University of Wrocław, Poland (2007) (in Polish)
17. Ortolf, C., Schindelhauer, C.: Online multi-robot exploration of grid graphs with rectangular obstacles. In: Proc. 24th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA), pp. 27–36 (2012)