Maximal exploration of trees with energy-constrained agents

Evangelos Bampas, Jérémie Chalopin, Shantanu Das, Jan Hackfeld, Christina Karousatou

Comment explorer un arbre inconnu avec des agents à énergie limitée ?

Evangelos Bampas[1], Jérémie Chalopin[1], Shantanu Das[1], Jan Hackfeld[2] et Christina Karousatou[1]

[1]*LIF, Aix-Marseille Université & CNRS, France*
[2]*TU Berlin, Institute of Mathematics, Germany*

On souhaite explorer un arbre inconnu avec un groupe d'agents mobiles initialement regroupés sur un sommet. Chaque agent dispose d'une énergie limitée et ne peut pas traverser plus de B arêtes. On souhaite maximiser le nombre de sommets visités (par au moins un agent) lors de l'exécution. Initialement, les agents n'ont aucune connaissance sur la structure de l'arbre, mais ils en découvrent la topologie au fur et à mesure qu'ils traversent de nouvelles arêtes. Nous supposons que les agents peuvent communiquer entre eux à distance illimitée, donc la connaissance qu'un agent obtient lors de la traversée d'une arête est instantanément transmise aux autres agents. Nous proposons un algorithme très intuitif, basé sur le parcours en profondeur, et nous étudions son efficacité par rapport à la solution optimale qu'on peut obtenir lorsqu'on connaît initialement la carte. Nous prouvons que cet algorithme a un rapport de compétitivité constant. Nous fournissons également une borne inférieure sur le rapport de compétitivité réalisable par un algorithme quelconque.

Mots-clefs : agents mobiles, exploration, arbre inconnu, énergie limitée

1 Introduction

The problem of exploration of an unknown graph by one or more robots (agents) is a well known problem with many applications ranging from searching the internet to physical exploration of unknown terrains using mobile sensor robots. Most results on exploration algorithms focus on minimizing the exploration time or sometimes the memory of the agent. For a brief survey of such results see [3].

We study the exploration problem under a very natural constraint that each agent can traverse at most a finite number of edges, denoted by an integer B (henceforth called the *energy budget* of the agent). We assume that the agents have limited energy resources, and movement consumes energy. A similar restriction was considered in the *piecemeal exploration* problem [1], where the agent could refuel by going back to its starting location. Thus, the exploration could be performed by a single agent using a sequence of tours starting and ending at the root vertex. On the other hand, [5] and [4] studied exploration without refuelling, using multiple agents with the objective of minimizing the energy budget per agent, or the number of agents needed for a fixed budget. In the above studies, the graph (or tree) to be explored was assumed to be of a restricted diameter allowing an agent with fixed budget to visit any node of the graph.

In this paper, we focus on exploration of tree networks of arbitrary size and structure. Since exploring the complete tree is not always possible, our objective is to visit as many nodes of the tree as possible. We provide and analyze an online algorithm for partial exploration of an unknown tree, using a fixed number of agents with fixed energy budgets.

Model and problem description The agents operate in an undirected tree T. The edges at every vertex v in T have locally distinct edge labels $0, \ldots, \delta_v - 1$, where δ_v is the degree of v. These edge labels are referred to as the *local port numbers* at v. Initially, a group of k agents (numbered 1 to k) are placed at a node r of T. Each agent has limited energy B and it consumes one unit of energy for every edge that it traverses.

The tree is initially unknown to the agents, but they learn the map of the tree as they traverse new edges. Each time an agent arrives at a new node, it learns the local port number of the edge through which it arrived, as well as the degree of the node. We assume that agents can communicate at arbitrary distances, so the updated map of the tree, including all agent positions, is instantaneously available to all agents (global communication). Moreover, we will assume, without loss of generality, that the local port number of the edge leading back to the root r is $\delta_v - 1$ for any vertex $v \neq r$ in T.

The goal is to design an algorithm A that maximizes the number of edges collectively discovered by the agents. If $I = \langle T, r, k, B \rangle$ is an instance of the problem, let $A(I)$ denote the number of edges explored using algorithm A on I. We measure the performance of an algorithm A by the competitive ratio $\rho_A = \sup_I \frac{\textsc{Opt}(I)}{A(I)}$, where $\textsc{Opt}(I)$ is the maximum number of edges that can be explored on instance I with full initial knowledge of the instance.

2 Exploration algorithm

Given an instance $I = \langle T, r, k, B \rangle$ of the online tree exploration problem, for $d \geq 0$ we let T_d denote the induced subtree of T containing all vertices at distance at most d from r. We propose an algorithm that works in phases. In each phase i, the agents attempt to completely explore T_{d_i}, for some $d_i \geq 0$. The increasing sequence $(d_i)_{i \geq 1}$ depends only on B and is chosen appropriately, so as to optimize the obtained competitive ratio.

In each phase i, the agents essentially perform a collaborative depth-first search within T_{d_i} (they ignore unexplored edges that lead to nodes at distance greater than d_i from r). Agents are sent off sequentially, i.e., the next agent waits until the current one has depleted its energy. Each agent always chooses the smallest local port number that leads to an unexplored edge within T_{d_i}. Formally, the algorithm is as follows: (the computation of $(d_i)_{i \geq 1}$ is explained in the proof of Theorem 1)

Instance: Tree T, root vertex r, number of agents k, energy budget B
1: **for** $i \leftarrow 1, 2, \ldots$ **do**
2: **while** there is an agent a at r and T_{d_i} is not fully explored **do**
3: $a \leftarrow$ the agent with the least positive remaining energy at r
4: **while** agent a has energy and T_{d_i} is not fully explored **do**
5: agent a follows the smallest local port number that leads to an unexplored edge in T_{d_i}
6: **while** agent a has energy and it is not at r **do**
7: agent a moves towards r

Note that, if the current agent stops and leaves some unexplored edges in the subtree rooted at its last position, then the next agent will move to the last position of the current agent and will continue the depth-first search. On the other hand, if the current agent has completely explored the subtree rooted at its last position, the next agent will take a shortcut to the next node along the depth-first search path of T_{d_i} that has unexplored edges.

Theorem 1. *The sequence $(d_i)_{i \geq 1}$ can be chosen as a function of B so that the proposed algorithm achieves a competitive ratio of $1 + 4\varphi < 7.473$, where $\varphi \approx 1.618$ is the golden ratio.*

Proof. Consider the sequence $(f_i)_{i \geq 0}$ given by $f_0 = 0$, $f_1 = 1$, $f_2 = 2$, and $f_i = f_{i-1} + f_{i-2}$ for $i \geq 3$. For $i \geq 0$, let $S_i = \sum_{j=0}^{i} f_j$ and let $\gamma \geq 1$ be the smallest index such that $S_{\gamma+1} > B$. Note that $S_i = f_{i+2} - 2$ for all $i \geq 0$ and, furthermore, $\gamma \geq 2$ if and only if $B \geq 3$. Now, for $i = 1, \ldots, \gamma$, we define $d_i = B - S_{\gamma - i}$. The sequence $(d_1, \ldots, d_\gamma)$ is strictly increasing and, setting for convenience $d_0 = 0$, it satisfies $d_{i-1} - d_{i-2} > B - d_i$ for $i \geq 2$.

We will assume that the instance is such that the algorithm fails to explore the whole tree T. We further assume for the moment that $B \geq 3$, therefore $\gamma \geq 2$, and that the last phase to be executed by the algorithm is phase $\sigma \geq 2$, i.e., $T_{d_{\sigma-1}}$ is completely explored and T_{d_σ} only partially.

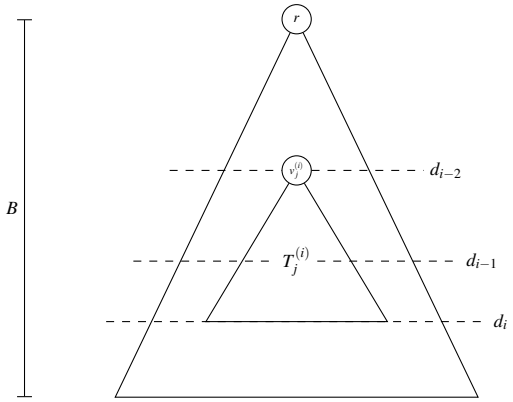

Fig. 1: Subtrees $T_j^{(i)}$ in tree T.

We first give an upper bound on the number of edges explored by an optimal offline algorithm OPT. Any agent in the offline algorithm can explore at most $B - d_{\sigma-1}$ new edges that are not contained in $T_{d_{\sigma-1}}$, because it uses up at least $d_{\sigma-1}$ of its energy to reach the bottommost node of $T_{d_{\sigma-1}}$. We can therefore bound the number of edges explored by OPT as $|\text{OPT}| \leq |T_{d_{\sigma-1}}| + k(B - d_{\sigma-1})$.

Next, we give a lower bound on the number of edges $|\text{ALG}|$ explored by our algorithm. For $i = 1, \ldots, \sigma$, let n_i be the number of newly explored edges in phase i of the algorithm and k_i be the number of fresh agents (not carried over from the previous phase) used in this phase. For convenience, let also $n_0 = 0$. Note that we have $\sum_{i=1}^{\sigma-1} n_i = |T_{d_{\sigma-1}}|$.

Consider the first phase of the algorithm. Let DFS_1 be the closed walk visiting all edges in T_{d_1} in the same order as a depth-first search, which always follows the local port with the smallest number first. The length of DFS_1 satisfies $|\text{DFS}_1| \leq 2n_1$. The first agent used by the algorithm will spend all of its energy making a progress of B steps on DFS_1. The second agent either moves to where the first agent stopped or shortcuts to a point in the path from r to where the first agent stopped, and then makes progress on the closed walk DFS_1. Note that if the previous agent stopped at distance d_1, then the next agent will always shortcut. In any case, the second agent uses at most $d_1 - 1$ of its energy before it starts making progress on DFS_1, so the progress is at least $B - d_1 + 1$. Similarly, all other agents in the first phase, except for the last agent, make a progress of at least $B - d_1 + 1$ on DFS_1, yielding a total progress of at least $B + (k_1 - 2)(B - d_1 + 1)$ for the first $k_1 - 1$ agents. If p is the progress contributed by the k_1-th agent, we have $B + (k_1 - 2)(B - d_1 + 1) + p \leq 2n_1$.

The k_1-th agent used at most $d_1 - 1$ of its energy before it made progress p, therefore at the end of phase 1 either its energy is exhausted (which implies $p \geq B - d_1 + 1$, and the previous progress inequality gives $k_1(B - d_1 + 1) \leq 2n_1$), or it is located at the root with at least $B - d_1 + 1 - p$ available energy. If its remaining energy is positive, it will be the first agent to be activated in the second phase.

Now assume that the algorithm has completed phase $i - 1$ for $i \geq 2$ and therefore completely explored $T_{d_{i-1}}$. Let $v_1^{(i)}, v_2^{(i)}, \ldots, v_{t_i}^{(i)}$ be all vertices of depth d_{i-2} in T, whose subtrees contain unexplored edges. Moreover, let $T_j^{(i)}$ be the induced subtree of T_{d_i} with root $v_j^{(i)}$ (Fig. 1) and $n_j^{(i)}$ be the number of edges of $T_j^{(i)}$. The subtrees $T_j^{(i)}$ are completely explored up to the vertices at level d_{i-1}, but unexplored below. As all vertices of the subtrees $T_j^{(i)}$ lie at a distance between d_{i-2} and d_i from the root, we have $\sum_{j=1}^{t_i} n_j^{(i)} \leq n_{i-1} + n_i$.

Let $k_j^{(i)}$ be the number of agents that start in the i-th phase and reach the vertex $v_j^{(i)}$ with energy at least $B - d_{i-2}$. As the agents in every phase only move to subtrees with unexplored edges, every agent used in phase i, will move to one of the subtrees $T_j^{(i)}$ and therefore arrive at $v_j^{(i)}$ with energy $B - d_{i-2}$. We therefore have $\sum_{j=1}^{t_i} k_j^{(i)} = k_i$. We now want to bound the number of agents $k_j^{(i)}$ that we need to explore a subtree $T_j^{(i)}$ in terms of $n_j^{(i)}$ as above. Let $\text{DFS}_j^{(i)}$ be a depth-first search tour of all vertices in $T_j^{(i)}$. Then $|\text{DFS}_j^{(i)}| \leq 2n_j^{(i)}$ and $|\text{DFS}_j^{(i)}| \geq 2(d_{i-1} - d_{i-2})$ because $T_j^{(i)}$ contains an unexplored leaf at distance at least $d_{i-1} - d_{i-2}$ from

$v_j^{(i)}$. Every agent that enters $T_j^{(i)}$ with energy at least $B-d_{i-2}$, will either move to the previous agent's stopping position or shortcut, thus making at least $B-d_i+1$ progress on $\mathrm{DFS}_j^{(i)}$. This also holds for the last agent because at the time the last agent enters $T_j^{(i)}$ there is still an unexplored leaf and thus also the last agent can make a progress of at least $B-d_i+1$ on $\mathrm{DFS}_j^{(i)}$ as the part of $\mathrm{DFS}_j^{(i)}$ returning from the last unexplored leaf to $v_j^{(i)}$ is at least $d_{i-1}-d_{i-2} > B-d_i$. We get $k_j^{(i)} \cdot (B-d_i+1) \leq 2n_j^{(i)}$ and hence assuming $i \geq 3$: $k_i \cdot (B-d_i+1) = \sum_{j=1}^{t_i} k_j^{(i)} \cdot (B-d_i+1) \leq \sum_{j=1}^{t_i} 2n_j^{(i)} \leq 2n_{i-1}+2n_i$. By considering the subtree T'_{d_σ} of T_{d_σ} containing all vertices explored by our algorithm, we obtain the above inequality also for the last phase σ of the algorithm. In particular for $i=2$, we also take into account the contribution of the last agent of phase 1 (assuming it reached r), which starts at $v_1^{(2)} = r$ with at least $B-d_1+1-p$ available energy and, since it is the first agent to contribute to $DFS_1^{(2)}$, it contributes at least $B-d_1+1-p$ progress. The total progress during the second phase is, therefore: $B-d_1+1-p+k_2 \cdot (B-d_2+1) \leq 2n_1+2n_2$. If, however, the last agent of the first phase didn't reach r, then for phase 2 we have simply $k_2 \cdot (B-d_2+1) \leq 2n_1+2n_2$.

Summing the progress inequalities from each phase and using the monotonicity of the d_i and the fact that $\sum_{i=1}^{\sigma} k_i = k$, we obtain $k(B-d_\sigma+1) \leq \sum_{i=1}^{\sigma} k_i(B-d_\sigma+1) \leq \sum_{i=1}^{\sigma} k_i(B-d_i+1) \leq \sum_{i=1}^{\sigma} 2n_{i-1}+2n_i \leq 4|\mathrm{ALG}|$. Combining the upper bound on OPT, the lower bound on ALG above, and the inquality $|T_{d_{\sigma-1}}| \leq |\mathrm{ALG}|$, we obtain $\frac{|\mathrm{OPT}|}{|\mathrm{ALG}|} \leq \frac{|T_{d_{\sigma-1}}|+k(B-d_{\sigma-1})}{|\mathrm{ALG}|} \leq 1+4\frac{B-d_{\sigma-1}}{B-d_\sigma+1} = 1+4\frac{S_{\gamma-\sigma+1}}{1+S_{\gamma-\sigma}} = 1+4\frac{-2+f_{\gamma-\sigma+3}}{-1+f_{\gamma-\sigma+2}}$. It can be verified that $\frac{-2+f_{i+3}}{-1+f_{i+2}}$ is strictly increasing in i for $i \geq 0$ and it converges to φ, therefore $\rho_A \leq 1+4\varphi$.

Finally, if $B \leq 2$, which implies $\gamma = \sigma = 1$, or if $\sigma = 1 < \gamma$, then the algorithm finishes in the first phase and it follows from the above arguments that $|\mathrm{ALG}| \geq \frac{k(B-d_1+1)}{2}$, while $|\mathrm{OPT}| \leq kB$. Therefore, $\rho_A \leq \frac{2B}{B-d_1+1} = \frac{2B}{1+S_{\gamma-1}}$. However, recall that $S_{\gamma+1} > B$, so we have $\rho_A \leq 2\frac{-1+S_{\gamma+1}}{1+S_{\gamma-1}} = 2\frac{-3+f_{\gamma+3}}{-1+f_{\gamma+1}}$. It can be verified that $\frac{-3+f_{i+3}}{-1+f_{i+1}}$ is strictly increasing in i for $i \geq 2$ and it converges to $\varphi+1$, therefore $\rho_A \leq 2+2\varphi < 1+4\varphi$. □

2.1 Lower bound on the competitive ratio

Proposition 2. *No algorithm achieves a competitive ratio of* $2-o(1)$.

Proof. Given positive integers k and B, where B is even, consider the star consisting of k rays of length B and $\frac{kB}{2}$ rays of length 1. A group of k agents, each with energy B, start on r. For every algorithm, the adversary can ensure that no agent ever enters a long ray: Whenever an agent is at the center and decides to follow an unexplored edge, the adversary directs it to a short ray. For every edge that an agent explores, it needs to go back to the center in order to explore other edges. Therefore, every agent can explore at most $\frac{B}{2}$ edges for a total of at most $\frac{kB}{2}$ edges. On the other hand, the optimal solution is to send all agents in the long rays and explore kB edges. □

References

[1] B. Awerbuch, M. Betke, and M. Singh. Piecemeal graph learning by a mobile robot. *Information and Computation*, 152:155–172, 1999.

[2] J. Anaya, J. Chalopin, J. Czyzowicz, A. Labourel, A. Pelc, Y. Vaxés. Convergecast and Broadcast by Power-Aware Mobile Agents. Algorithmica, 1–39, 2014.

[3] S. Das. Mobile Agents in Distributed Computing: Network Exploration. Bulletin of the European Association for Theoretical Computer Science (EATCS), No. 109, pages 54–69, 2013.

[4] S. Das, D. Dereniowski, and C. Karousatou, Collaborative Exploration by Energy-Constrained Mobile Robots. In Proc. 22nd Int. Colloquium on Structural Information and Communication Complexity (SIROCCO), LNCS 9439, pages 357-369, 2015.

[5] M. Dynia, M. Korzeniowski, and C. Schindelhauer. Power-Aware Collective Tree Exploration. In Proc. 19th Int. Conference on Architecture of Computing Systems (ARCS), pp. 341-351, 2006.