Brief Announcement: Crash-Tolerant Exploration of Trees by Energy Sharing Mobile Agents

Quentin Bramas
University of Strasbourg, ICUBE, CNRS, Strasbourg, France

Toshimitsu Masuzawa
Graduate School of Information Science and Technology, Osaka University, Japan

Sébastien Tixeuil
Sorbonne University, CNRS, LIP6, Institut Universitaire de France, Paris, France

Abstract

We consider the problem of graph exploration by energy sharing mobile agents that are subject to crash faults. More precisely, we consider a team of two agents where at most one of them may fail unpredictably, and the considered topology is that of acyclic graphs (*i.e.* trees). We consider both the asynchronous and the synchronous settings, and we provide necessary and sufficient conditions about the energy in two settings: line-shaped graphs, and general trees.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Computing methodologies → Mobile agents

Keywords and phrases Mobile Agents, Distributed Algorithms, Energy sharing

Digital Object Identifier 10.4230/LIPIcs.SAND.2024.25

Funding This research was supported in part by Agence Nationale de la Recherche under Grant Agreement SAPPORO (Ref. 2019-CE25-0005-1) and by JSPS KAKENHI 20KK0232.

1 Context

This paper investigates the collective exploration of a known edge-weighted graph by mobile agents originating from arbitrary nodes. The objective is to traverse every edge at least once. Each agent possesses a battery with an initial energy level (that may differ among agents). An agent's battery is depleted by x when it travels a distance of x. Also, when two agents meet, they may freely exchange remaining energy. Finally, the possibility for one of the two agents to crash, or cease functioning indefinitely and unpredictably, exists.

Energy transfer by mobile agents was previously considered by Czyzowicz et al. [3]. Agents travel and spend energy proportional to distance traversed. Some nodes have information acquired by visiting agents. Meeting agents may exchange information and energy. They consider communication problems where information held by some nodes must be communicated to other nodes or agents. They deal with data delivery and convergecast problems for a centralized scheduler with full knowledge of the instance. With energy exchange, both problems have linear-time solutions on trees. For general undirected and directed graphs, these problems are NP-complete. Then, Czyzowicz et al. [2] consider the gossiping problem in tree networks. In an edge-weighted tree network, agents spend energy while traveling and collect copies of data packets from visited nodes. They deposit copies of possessed data packets and collect copies of data packets present at the node. Czyzowicz et al. [2] prove that gossiping can be solved in $O(k^2n^2)$ time for an n-node tree with k agents.

Most related to our paper are the works by Czyzowicz et al. [4], Sun et al. [5], and Bramas et al. [1]. On the one hand, Czyzowicz et al. [4] study the collective exploration of a known n-node edge-weighted graph by k mobile agents with limited energy and energy transfer capability. The goal is for every edge to be traversed by at least one agent. For an n-node path, they give an $O(n + k)$ time algorithm to find an exploration strategy or report that none exists. For an n-node tree with ℓ leaves, they provide an $O(n + \ell k^2)$ algorithm to find an exploration strategy if one exists. For general graphs, deciding if exploration is possible by energy-sharing agents is NP-hard, even for 3-regular graphs. However, it's always possible to find an exploration strategy if the total energy of agents is at least twice the total weight of edges; this is asymptotically optimal. Next, Sun et al. [5] examines circulating graph exploration by energy-sharing agents on an arbitrary graph. They present the necessary and sufficient energy condition for exploration and an algorithm to find an exploration strategy if one exists. The exploration requires each node to have the same number of agents before and after. Finally, Bramas et al. [1] considered the problem of exploring every weighted edge of a given ring-shaped graph using a team of two mobile energy-sharing agents. They introduce the possibility for one of the two agents to fail unpredictably and cease functioning permanently (i.e., crashing). In this context, Bramas et al. [1] considered two scenarios: asynchronous (where no limit on the relative speed of the agents is known, so one agent cannot wait at a meeting point for another agent for a bounded amount of time and infer that the other agent has crashed, as it may simply be arbitrarily slow), and synchronous (where the two agents have synchronized clocks and move at precisely the same speed).

2 Model

Our model is similar to that proposed by Bramas et al. [1].

We are given a weighted graph $G = (V, E)$ where V is a set of n nodes, E is a set of m edges, and each edge $e_i \in E$ is assigned a positive integer $w_i \in \mathbb{N}^+$, denoting its weight (or length). We have k mobile agents (or agents for short) $r_0, r_1, \ldots, r_{k-1}$ respectively placed at some of the nodes $s_0, s_1, \ldots s_{k-1}$ of the graph. We allow more than one agent to be located in the same place. Each agent r_i initially possesses a specific amount en_i of energy for its moves. An agent has the ability to travel along the edges of graph G in any direction. It can pause its movement if necessary and can change its direction either at a node or while traveling along an edge. The energy consumed by a moving agent is equal to the distance x it moved. An agent can move only if its energy is greater than zero. Now, the distance between two agents (that is, the minimum sum of the weights for all the paths connecting them) is the smallest amount of energy needed for them to meet at some point.

In our setting, agents can share energy with each other. When two agents, r_i and r_j, meet at a vertex or edge, r_i can take some energy from r_j. If their energy levels at meeting time meeting are en'_i and en'_j, then r_i can take an amount of energy $0 < en \leq en'_j$ from r_j. After the transfer, their energy levels are $en'_i + en$ and $en'_j - en$, respectively.

Each agent adheres to a pre-established trajectory until encountering another agent. At this point, the agent determines if it acquires energy, and calculates its ensuing trajectory. The definition of a trajectory depends on the synchrony model:

- **In the synchronous model**, a trajectory is a sequence of pairs $((u_0, t_0), (u_1, t_1), \ldots)$, where u_i is a node, and t_i denotes the time at which the agent should reach u_i. For each $i \geq 0$, $t_i < t_{i+1}$, and u_{i+1} is either equal to u_i (i.e., the agent waits at u_i between t_i and t_{i+1}), or is adjacent to u_i (i.e., the agent leaves u_i at time t_i and arrives at u_{i+1} at time t_{i+1}). For simplicity, we assume in our algorithm that the moving speed is always one (it takes time d to travel distance d, so if $u_i \neq u_{i+1}$ and the weight of edge (u_i, u_{i+1}) is w, then $t_{i+1} - t_i = w$).
- **In the asynchronous model**, a trajectory is just a sequence of nodes $(u_0, u_1, u_2, \ldots)$, u_{i+1} being adjacent to u_i for each $i \geq 0$, and the times at which it reaches the nodes are determined by an adversary.

In other words, in the synchronous model, the agent controls its speed and its waiting time at nodes, while an adversary decides them in the asynchronous model.

The computation of the trajectory and the decision to exchange energy is based on a *localized algorithm* (that is, an algorithm executed by the agent). In a given execution, the configuration at time t is denoted by C_t.

Localized algorithm. A *localized algorithm* f_i executed by an agent r_i at time t takes as input the *pasts* of r_i and its collocated agents, and returns *(i)* its ensuing trajectory $traj_i$ and *(ii)* the amount of energy $take_{i,j}$ taken from each collocated agent r_j. The past $Past_i(t)$ of r_i at time t corresponds to the path already traversed by r_i union the past of all the previously met agents. More formally:

$$Past_i(t) = \{path_i(t)\} \cup \{Past_j(t') \mid r_i \text{ met } r_j \text{ at time } t' \leq t\}$$

A set of localized algorithms is *valid* for a given initial configuration c if, for any execution starting from c, agents that are ordered to move have enough energy to do so and when an agent r_i takes energy from an agent r_j at time t, then r_j does not take energy from r_i at t.

In this paper, we consider the possibility of agent crashes. At any point in the execution, an agent r_i may crash and stop operating forever. However, if r_i has remaining energy $en'_i > 0$, then other agents meeting r_i may take energy from r_i. Now, a set of localized algorithms is t-crash-tolerant if it is valid even in executions where at most t agents crash.

We are interested in solving the problem of t-crash-tolerant collaborative exploration:

***t*-crash-tolerant collaborative exploration.** Given a weighted graph $G = (V, E)$ and k mobile agents $r_0, r_1, \ldots, r_{k-1}$ together with their respective initial energies $en_0, en_1, \ldots, en_{k-1}$ and positions $s_0, s_1, \ldots, s_{k-1}$ in the graph, find a valid set of localized algorithms that explore (or cover) all edges of the graph despite the unexpected crashes of at most $t < k$ agents.

This paper focuses on the 1-crash-tolerant collaborative exploration of *trees* by *two* agents.

3 Our Results

We consider the problem of graph exploration by energy-sharing mobile agents that are subject to crash faults. More precisely, we consider a team of two agents where at most one of them may fail unpredictably, and the considered topology is that of acyclic graphs (*i.e.* trees). Similarly to Bramas et al. [1] who studied the case of ring-shaped networks, we consider both the asynchronous and the synchronous settings, and we provide necessary and sufficient conditions for the initial amounts of energy in two settings: lines and trees. In the following, en_0 and en_1 denote the initial energy of the first and second agents, respectively.

Lines. In the case of the line, x (resp. y) denotes the distance of the first (resp. second) agent to the left border of the line, assuming $x \leq y$ and $x \leq \ell - y$, while ℓ denotes the weight of the line. In the asynchronous case, a necessary and sufficient condition is:

$$\begin{aligned} & (en_0 \geq x + y) \wedge (en_1 \geq y) \wedge (en_0 + en_1 \geq 2\ell + x + y) \\ \vee \quad & (en_0 \geq \ell - x) \wedge (en_1 \geq 2\ell - (x + y)) \wedge (en_0 + en_1 \geq 4\ell - (x + y)) \\ \vee \quad & (en_0 \geq \ell + x) \wedge (en_1 \geq 2\ell - y) \\ \vee \quad & (en_0 \geq y - x) \wedge (en_1 \geq y - x) \wedge (en_0 + en_1 \geq min(3\ell + y - x, 2\ell - x + 3y)) \end{aligned}$$

In the synchronous case, a necessary and sufficient condition is:

$$\begin{aligned} & (en_0 \geq x + y) \wedge (en_1 \geq y) \wedge (en_0 + en_1 \geq max(\ell + x + y, 2\ell + x - y)) \\ \vee \quad & (en_0 \geq \ell - x) \wedge (en_1 \geq 2\ell - (x + y)) \wedge (en_0 + en_1 \geq 3\ell - x - y) \\ \vee \quad & (en_0 \geq \ell + x) \wedge (en_1 \geq 2\ell - y) \\ \vee \quad & (en_0 \geq y - x) \wedge (en_1 \geq y - x) \wedge (en_0 + en_1 \geq 2\ell - x + y) \end{aligned}$$

Trees. In the case of a weighted tree T, d denotes the diameter of the tree, x the initial distance between the two agents, and W its total weight. In the asynchronous case, a sufficient condition is:

$$(en_0 \geq x) \wedge (en_1 \geq x) \wedge (en_0 + en_1 \geq 2W + 2d\lceil \log_{3/2} W \rceil + x + 2) \tag{1}$$

We provide a lower bound on the total energy for unweighted star graphs: $en_0 + en_1$ cannot be in $2W + 2\log(o(W))$ (notice $W = |E|$).

The main ingredients for our positive result are as follows.

First, we construct a family of k connected non-empty subtrees of T named $T_1, T_2, \ldots, T_k$, where $T_i = (V_i, E_i)$ and $(E_i)_{1\leq i\leq k}$ forms a partition of E. At the beginning, the agents meet to share energy. Then the agents repeat a procedure *explore*(T_i) for all $i \in \{1, \ldots, k\}$ from 1 to k. The procedure assumes that the agents are initially at the same location (possibly on an edge), and ensure that after the execution the agents are at the same location (not necessarily the same as the initial one) if $i < k$ (when $i = k$ the agents can terminate anywhere on completion of the exploration).

Agent r_0 (resp. r_1) executing *explore*(T_i) first moves to the closest node v_i of T_i, executes *EulerianExplore*(T_i) (resp. *ReverseEulerianExplore*(T_i)), and moves back to its initial location, until it meets the other agent, and T_i is explored. If the agents meet before ending this sequence of moves and E_i is explored, then the procedure terminates. This occurs during the exploration of the Eulerian tour from v_i, or when one of the agents r comes back from v_i to its initial location after completing its Eulerian tour while the other has not started it (it is still moving towards v_i from the location where it started executing *explore*(T_i)).

Since the length of the Eulerian tour is $2w(T_i)$ (where $w(T_i)$ denotes the weight of T_i) and the distance to v_i from their initial location is d in the worst case, each agent must have, at the beginning of the procedure, the energy of at least $2d + 2w(T_i)$ if $i < k$ (to terminate even when the other agent remains at the initial location), at least $d + 2w(T_i)$ if $i = k$. When the procedure terminates, the total energy consumed during the procedure is at most $2d + 2w(T_i)$ (because every edge traversed in the procedure is traversed exactly twice if $i < k$, and at most twice if $i = k$).

Consequently, to complete all *explore*(T_i), for every i, sequentially, our algorithm requires that the total remaining energy EN_i at the beginning of the procedure *explore*(T_i) is as follows, where x is the initial distance between the agents:

- $EN_k \geq 2d + 4w(T_k)$
- $EN_i \geq \max(2d + 2w(E_i) + EN_{i+1}, 4d + 4w(T_i))$ $(2 \leq i \leq k-1)$
- $EN_1 \geq x + \max(2d + 2w(T_1) + EN_2, 4d + 4w(T_1))$ where x is the initial weighted distance between the agents.

Moreover, the total energy consumption for exploring T is at most $x + \sum_{i=1..k}(2d + 2w(T_i)) = 2W + 2kd + x$.

We now have to construct the partition $T_1, T_2, \ldots, T_k$ of T so that k should be small to reduce the number of calls to *explore*(), but each $w(T_i)$ should not be too large to avoid increasing the energy required at the beginning of *explore*(T_i). A good partition could be to have $w(T_k) = 1$ and $w(T_i) = 2w(T_{i+1})$, which results in $k = \lceil \log W \rceil$. In general trees, such a partition does not exist, but we can obtain a similar result using the centroid-based partition recursively, which guarantees $W_i/3 \leq w(T_i) \leq W_i/2$ (W_i is the total weight of the remaining part of the tree).

Let $T = (V, E)$ be a weighted tree with total weight W. The centroid of T is defined as follows. In the following, for a tree T and a node u of T, T can be regarded as a rooted tree, denoted by T^u, rooted at u. For the root u and its neighbor v, let T^u_v be the subtree of T^u rooted at v.

1. When there exists an edge $(u, v) \in E$ satisfying $w(T_v^u) < W/2$ and $w(T_u^v) < W/2$, the centroid of T is the point p on edge (u, v) such that $w(T_v^u) + w(v, p) = w(T_u^v) + w(u, p) = W/2$. We call p the *edge centroid*.
2. When there exists a node $u \in V$ satisfying $w(T_v^u) + w(u, v) \leq W/2$ for each neighbor v of u, the centroid of T is node u. We call u the *node centroid*.

By spliting the tree recursively at the centroid point, we can construct a partition $T_1, \ldots, T_k$ with $k = \lceil \log_{3/2} W \rceil$ to obtain the sufficient condition (1).

In the synchronous case, a sufficient condition is:

$$(en_0 \geq x) \wedge (en_1 \geq x) \wedge (en_0 + en_1 \geq 2W + d + x)$$

On the other hand we show that there exists an infinite family of trees such that the required total energy is at least $2W + \frac{d}{2} - 3$.

4 Conclusion

We characterized the solvability of exploration with two crash-prone energy-sharing mobile agents in the case of tree topologies, both in the synchronous and in the asynchronous settings. Obvious open questions include further closing the gap between necessary and sufficient conditions for the initial amounts of energies in the case of trees, solving the problem with more than two agents, and considering general graphs.

Also, our model for energy transfer is very simple (all energy can be transferred instantaneously between two agents, at no cost). It would be interesting to study non-linear battery models (where the capacity decreases faster if more instantaneous current is drawn, and the capacity increases less if faster charge is executed) in this context.

References

1 Quentin Bramas, Toshimitsu Masuzawa, and Sébastien Tixeuil. Brief announcement: Crash-tolerant exploration by energy sharing mobile agents. In Shlomi Dolev and Baruch Schieber, editors, *Stabilization, Safety, and Security of Distributed Systems - 25th International Symposium, SSS 2023, Jersey City, NJ, USA, October 2-4, 2023, Proceedings*, volume 14310 of *Lecture Notes in Computer Science*, pages 380–384. Springer, 2023. doi:10.1007/978-3-031-44274-2_28.

2 Jurek Czyzowicz, Dariusz Dereniowski, Robert Ostrowski, and Wojciech Rytter. Gossiping by energy-constrained mobile agents in tree networks. *Theor. Comput. Sci.*, 861:45–65, 2021. doi:10.1016/j.tcs.2021.02.009.

3 Jurek Czyzowicz, Krzysztof Diks, Jean Moussi, and Wojciech Rytter. Communication problems for mobile agents exchanging energy. In Jukka Suomela, editor, *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, volume 9988 of *Lecture Notes in Computer Science*, pages 275–288, 2016. doi:10.1007/978-3-319-48314-6_18.

4 Jurek Czyzowicz, Stefan Dobrev, Ryan Killick, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, Denis Pankratov, and Sunil M. Shende. Graph exploration by energy-sharing mobile agents. In Tomasz Jurdzinski and Stefan Schmid, editors, *Structural Information and Communication Complexity - 28th International Colloquium, SIROCCO 2021, Wrocław, Poland, June 28 - July 1, 2021, Proceedings*, volume 12810 of *Lecture Notes in Computer Science*, pages 185–203. Springer, 2021. doi:10.1007/978-3-030-79527-6_11.

5 X. Sun, N. Kitamura, T. Izumi, and T. Masuzawa. Circulating exploration of an arbitrary graph by energy-sharing agents. In *Proc. of the 2023 IEICE General Conference*, pages 1–2, 2023.