Brief Announcement: Multi-Agent Online Graph Exploration on Cycles and Tadpole Graphs

Erik van den Akker[1] [0000-0003-4956-255X], Kevin Buchin[1] [0000-0002-3022-7877], and Klaus-Tycho Foerster[1] [0000-0003-4635-4480]

TU Dortmund, Germany

Abstract. We study the problem of multi-agent online graph exploration, in which a team of k agents has to explore a given graph, starting and ending on the same node. The graph is initially unknown. Whenever a node is visited by an agent, its neighborhood and adjacent edges are revealed. The agents share a global view of the explored parts of the graph. The cost of the exploration has to be minimized, where cost either describes the time needed for the entire exploration (time model), or the length of the longest path traversed by any agent (energy model). We investigate graph exploration on cycles and tadpole graphs for 2-4 agents, providing optimal results on the competitive ratio in the energy model (1-competitive with two agents on cycles and three agents on tadpole graphs), and for tadpole graphs in the time model (1.5-competitive with four agents). We also show competitive upper bounds of 2 for the exploration of tadpole graphs with three agents, and 2.5 for the exploration of tadpole graphs with two agents in the time model.

Keywords: Graph Exploration · Cycles · Tadpole Graphs · Agents.

1 Introduction

In the Online Graph Exploration problem, all nodes of a given weighted graph $G = (V, E, l)$ have to be visited by an agent a, starting and ending on a starting node $s \in V$. The graph is initially unknown, except for s and its neighborhood $N(s)$. Whenever the agent reaches a yet unvisited node v, all adjacent edges and $N(v)$ are revealed. Nodes can be distinguished, but the labels do not provide any information about the graph to the agent. We consider graph exploration with $k \in \mathbb{N}$ agents $a_1, ..., a_k$, for which we assume unlimited computational power and shared knowledge, meaning that as soon as any agent learns something about the neighborhood of a node, all other agents instantly receive the same information. A graph is considered explored, when each node in V has been visited by any agent, and all agents have returned to s. All agents move at identical speed, taking time $l(e)$ for traversing some edge e. An agent can decide to wait at a node, while other agents are traversing edges. The goal of a graph exploration strategy is to minimize the cost of the exploration.

This problem is widely used to describe the exploration of unknown terrain by multiple autonomous robots. Some variations focus on the general exploration

	Time Model		Energy Model	
Graph Class (# of Agents)	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Cycles (2 Agents)	1.5 [10]	1.5 [10]	1	1 (Thm. 2)
Tadpole Graphs (2 Agents)	1.5 (Thm. 3)	2.5 (Thm. 5)	1.5 (Thm. 4)	2.5 (Thm. 5)
Tadpole Graphs (3 Agents)	1.5 (Thm. 3)	2 (Thm. 6)	1	1 (Thm. 6)
Tadpole Graphs (4+ Agents)	1.5 (Thm. 3)	1.5 (Thm. 7)	1	1 (Thm. 6)

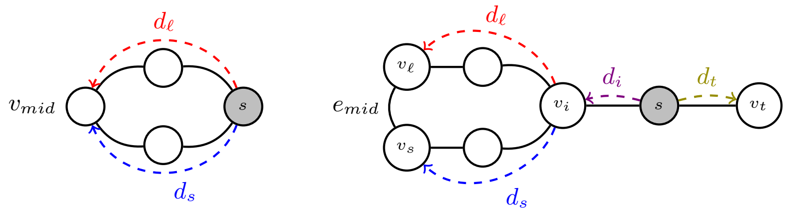
Table 1: The new bounds for the exploration of cycles and tadpole graphs.

time [10,3,8] (here called the **Time Model**), while others focus on the maximum distance traversed by any individual agent, modeling the maximum energy consumption of a single robot [5,6] (here called the **Energy Model**). We consider both models for the cost of an exploration.
In the single-agent case, many graph classes like directed graphs, tadpole graphs or cactus graphs have been investigated [7,2,9], while the main focus of the literature in the case of multi-agent exploration lies on general graphs and trees [5,4,3,8], with the exception of cycles (in the time model) and $n \times n$-grid graphs [10,11]. We consider cycles (in the energy model) and tadpole graphs, which consist of a cycle and a path attached to one of its nodes, called the tail. An overview of our results is given in Table 1.
Online algorithms are analyzed in terms of their competitive ratios, which describe the relationship between the costs of online and optimal solutions. Note that a c-competitive exploration strategy for the time model is also at most c-competitive for the energy model, since in an exploration that takes time $\mathcal{T}$, no agent can traverse a distance greater than $\mathcal{T}$. In the optimal case the maximum distance traversed by any agent matches the general exploration time.
In the following Sections we provide an overview of our results on Cycles in §2 and Tadpole Graphs in §3. A detailed version of this BA is provided on arXiv[1].

2 Cycles

Preliminaries We call the point on a cycle with exactly distance $L_c/2$ in both directions to the starting point s, the midpoint m of the cycle. If m falls onto an edge, this edge is called e_{mid}. If m falls onto a node, the node is called v_{mid}. In this case we consider $l(e_{mid}) = 0$. We consider $e_{mid} = (v_l, v_s)$. If $e_{mid} = (v_l, v_s)$ does not exist, we consider $v_l = v_s = v_{mid}$. We denote p_l and p_s with length d_l and d_s as edge-disjoint paths from s to v_l and v_s respectively, where p_l and p_s combined contain all edges of the cycle except (v_l, v_s). We always assume w.l.o.g that $d_s \leq d_l$. An example is given in Figure 1a.

Higashikawa et al. presented the *ALE (avoid longest edge) Algorithm*, which is a greedy 1.5-competitive online algorithm for the exploration of cycles in the time model [10], that sends two agents in different directions and always chooses the shorter edge. We first show that *ALE* is 1.5-competitive in the energy model and then present a modification we call the *AMP (avoid midpoint) Algorithm*,



(a) Cycle example. p_ℓ and p_s with lengths d_ℓ, d_s and v_{mid} are marked.

(b) Tadpole Graph example. d_ℓ, d_s, d_i, d_t and e_{mid} are marked.

Fig. 1: Examples of a cycle (a) and a tadpole graph (b).

reaching a competitive ratio of 1 in the energy model, while still being 1.5-competitive in the time model. For proving this we make use of an observation made by Higashikawa et al. [10]: an optimal strategy for exploring cycles with 2 Agents takes time $2d_\ell$.

Theorem 1 (The *ALE* algorithm and the energy model). *The ALE algorithm has a competitive ratio of* 1.5 *in the energy model on cycles.*

Proof. Consider the graph shown in Figure 2a. Let $0.5 > \varepsilon > 0$. Since (s, x_2) is the longest edge in the graph, one agent explores the graph clockwise from s to x_2. After the agent reaches x_2 the shortest path to s is the edge (x_2, s), leading to a cost of 3. An optimal strategy would have sent both agents from s to x_1 and x_2 respectively, leading to an exploration cost of $2(1 + \varepsilon)$. The overhead of ALE in this case is $\frac{3}{2(1+\varepsilon)}$, leading to a lower bound of 1.5. Since ALE is 1.5-competitive in the time model, the upper bound of 1.5 for the energy model follows.

Theorem 2 (AMP on Cycles). *For the energy model, the AMP (Avoid Midpoint) Algorithm 1 explores a cycle with a competitive ratio of* 1. *For the time model the algorithm explores a cycle with a competitive ratio of* 1.5.

Proof. We can prove the 1-competitiveness of the *AMP algorithm* for the energy model, by showing that the agents only traverse the cycle to v_s and v_ℓ (or v_{mid}) before returning, and never traverse the edge e_{mid}. We then use this result to prove the 1.5-competitive ratio for the time model.

Energy Model If both agents reach v_{mid} at the same time, all nodes have been visited and both agents have traversed a distance of $L/2$ before and L after backtracking, which matches the result of the optimal offline strategy.
If an agent a_1 reaches v_{mid} first, having traversed a distance $d_1 = L/2$, the other agent a_2 must have traversed a distance $d_2 < d_1$. Thus, a_2 moves to v_{mid} without stopping. Then both agents have traversed a distance of $L/2$ when all nodes have been visited and L after backtracking.

Algorithm 1 AMP (Avoid Midpoint)

Require: A unknown cycle graph $G = (V, E)$, Two agents a_1, a_2, a starting node $s \in V$

```
d(a1) ← 0; d(a2) ← 0                    ▷ Agents track their already traversed distance
Exp ← {s}                               ▷ Keep track of already explored vertices
Let n(a1) and n(a2) describe the next nodes seen by the agents
Assign the two neighbors of s randomly to n(a1) and n(a2)
while n(a1) ∉ Exp ∨ n(a2) ∉ Exp do      ▷ While graph is not explored
    if d(a1) + l(n(a1)) < d(a2) + l(n(a2)) then
        a1 traverses edge to n(a1)
        Exp ← Exp ∪ {n(a1)}
        d(a1) ← d(a1) + l(n(a1))
        n(a1) ← next revealed node
    else
        a2 traverses edge to n(a2)
        Exp ← Exp ∪ {n(a2)}
        d(a2) ← d(a2) + l(n(a2))
        n(a2) ← next revealed node
    end if
end while
a1 and a2 return to s using shortest paths.
```

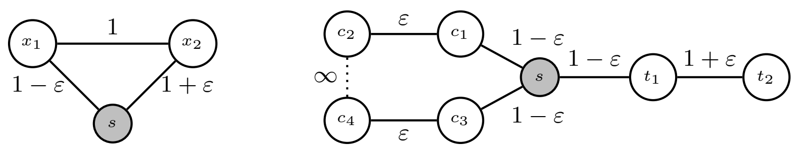
Assume w.l.o.g an agent a_1 is currently located on v_s after having traversed distance d_s and has not yet traversed e_{mid}, while the other agent a_2 has not reached v_ℓ yet. Since $d_\ell < L/2 < d_s + l(e_{mid})$ the agent a_2 traverses to v_ℓ without stopping. At this point all nodes have been visited and the agents backtrack, having taken the same paths as they would have in an optimal offline exploration.

Time Model Recall that an optimal strategy takes time $2d_\ell$. In the AMP algorithm, the agents traverse the distances d_s and d_ℓ. Since only one agent is traversing an edge at a time until all nodes have been visited, the maximum time needed to visit all nodes is $d_s + d_\ell \leq 2d_\ell$. After backtracking to s the full exploration time is at most $3d_\ell$, leading to a competitive ratio of 1.5.

3 Tadpole Graphs

Preliminaries A tadpole graph contains exactly one node with degree 1, which we call the end of the tail v_t and exactly one node with degree 3, we call the intersection v_i. All other nodes of the graph have degree 2. We define v_s, v_ℓ, v_{mid} similar to cycles, but starting at v_i instead of s, when s is located on the tail of the tadpole graph. An Example is given in Figure 1b. Note that any optimal exploration strategy on tadpole graphs takes at least time $2\max(d(s, v_\ell), d(s, v_t))$.

We first prove that the lower bound of 1.5 holds on tadpole graphs for at least two agents in the time model, and for two agents in the energy model in Thm.3 and



(a) Lower bound construction for ALE (energy model).

(b) Lower bound construction for Tadpole Graphs (energy model).

Fig. 2: Lower Bound constructions for: ALE and Tadpole Graphs (energy model).

Thm.4. For the following Thm.5-7, we roughly sketch the exploration strategies on tadpole graphs, and how they achieve the stated upper bounds for the given number of agents. A detailed descripton and analysis is provided on arXiv[1].

Theorem 3 (Time Model: Lower bound for exploring tadpole graphs). *For any number of agents $k \geq 2$, any online exploration strategy for tadpole graphs has a competitive ratio of at least 1.5 in the time model.*

Proof. We can adapt the cycle from the proof of Thm.2.2 by Higashikawa et al.[10], by adding a tail of length ε to the starting node. This does not influence the offline exploration time, since the path can always be explored by some agent in parallel and thus keeps the lower bound of 1.5 on tadpole graphs with any number of agents ≥ 2.

Theorem 4 (Energy Model: Lower bound for exploring tadpole graphs). *For $k = 2$ agents, no online exploration strategy on tadpole graphs can have a competitive ratio better than 1.5 in the energy model.*

Proof. We can adapt the proof idea from Lemma 7 by Dynia et al.[5], using a tree with three branches and connecting two branches with an arbitrarily long edge. The resulting tadpole graph is shown in Figure 2b. Let $\varepsilon > 0$ be an arbitrary small and ∞ be an arbitrary large number. The maximum distance traversed by any agent online here is at least $6 - 2\varepsilon$, while an optimal exploration has length 4. This leads to a lower bound of 1.5

Theorem 5 (Two-agent randomized tadpole graph exploration). *Using the AMP strategy with $k = 2$ agents for the exploration of tadpole graphs, choosing random paths as soon as the intersection is found leads to a competitive upper bound of 2.5.*

Using two agents, we randomly choose a direction as soon as the intersection is found and apply the AMP strategy on the chosen paths. As soon as the tail or the cycle are explored, one agent returns to s and explores the remaining path of the graph. This leads to an exploration time of at most $5 \max(d(s, v_\ell), d(s, v_t))$ and a competitive ratio of 2.5 in the time model, which also applies to the energy model.

Theorem 6 (Three agent tadpole graph exploration). *For $k = 3$ agents there exists an exploration strategy for tadpole graphs with a competitive ratio of* 2 *for the time-, and* 1 *for the energy model.*

Using three agents, when starting at the intersection one agent is sent in each direction. When starting on a node with degree 2, one agent waits at s until the intersection is found. Then each direction is assigned to an agent. At each step, only the agent with the shortest distance to s after the next edge, traverses the next edge. This leads to an exploration time of at most $4 \max(d(s, v_\ell), d(s, v_t))$ and a competitive ratio of 2 in the time model. Since any agent only traverses one path and returns, this leads to a maximum distance of $2 \max(d(s, v_\ell), d(s, v_t))$ and a competitive ratio of 1 in the energy model.

Theorem 7 (Four agent tadpole graph exploration). *Using $k = 4$ agents, a competitive ratio of* 1.5 *can be achieved on tadpole graphs in the time model.*

Using four agents, we send three teams containing one agent in each direction when starting on the intersection, and two teams of two agents in each direction when starting on a different node. When the intersection is first discovered, the team on the intersection splits into two single-agent teams. Only one team (the one that sees the largest distance to s) is waiting at any point in time. This leads to an exploration time of at most $3 \max(d(s, v_\ell), d(s, v_t))$ and a competitive ratio of 1.5 in the time model.

References

1. van den Akker, E., Buchin, K., Foerster, K.: Multi-agent online graph exploration on cycles and tadpole graphs. CoRR **abs/2402.13845** (2024)
2. Brandt, S., Foerster, K., Maurer, J., Wattenhofer, R.: Online graph exploration on a restricted graph class: Optimal solutions for tadpole graphs. Theor. Comput. Sci. **839**, 176–185 (2020)
3. Dereniowski, D., Disser, Y., Kosowski, A., Pajak, D., Uznanski, P.: Fast collaborative graph exploration. Inf. Comput. **243**, 37–49 (2015)
4. Disser, Y., Mousset, F., Noever, A., Skoric, N., Steger, A.: A general lower bound for collaborative tree exploration. Theor. Comput. Sci. **811**, 70–78 (2020)
5. Dynia, M., Korzeniowski, M., Schindelhauer, C.: Power-aware collective tree exploration. In: ARCS 2006. LNCS, vol. 3894, pp. 341–351. Springer (2006)
6. Dynia, M., Lopuszanski, J., Schindelhauer, C.: Why robots need maps. In: SIROCCO 2007. LNCS, vol. 4474, pp. 41–50. Springer (2007)
7. Foerster, K., Wattenhofer, R.: Lower and upper competitive bounds for online directed graph exploration. Theor. Comput. Sci. **655**, 15–29 (2016)
8. Fraigniaud, P., Gasieniec, L., Kowalski, D.R., Pelc, A.: Collective tree exploration. Networks **48**(3), 166–177 (2006)
9. Fritsch, R.: Online graph exploration on trees, unicyclic graphs and cactus graphs. Inf. Process. Lett. **168**, 106096 (2021)
10. Higashikawa, Y., Katoh, N., Langerman, S., Tanigawa, S.: Online graph exploration algorithms for cycles and trees by multiple searchers. J. Comb. Optim. **28**(2), 480–495 (2014)
11. Ortolf, C., Schindelhauer, C.: Online multi-robot exploration of grid graphs with rectangular obstacles. In: SPAA 2012. pp. 27–36. ACM (2012)