Efficient Explorations with swarms of Mobile Robots

Shantanu Das

Aix-Marseille University, France
&
Gran Sasso Science Institute, Italy

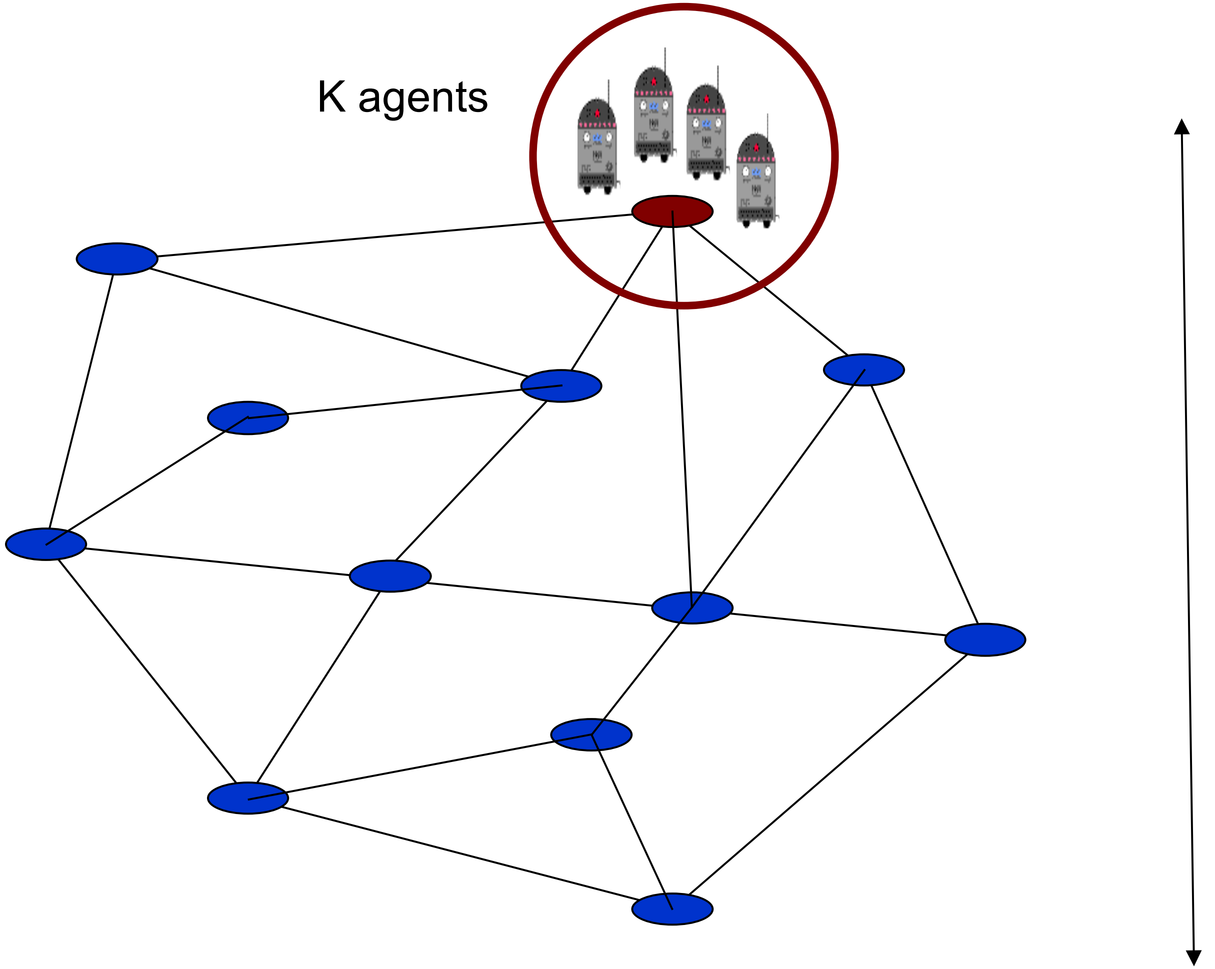
Team of Mobile Robots

Exploration by many robots

- Robustness to failures of robots
- Speedup by parallelisation
- Small robots are inexpensive (but have limited power)

Exploration Problem



The Model

- Environment: Connected undirected graph G.
- Nodes are distinctly labeled.
- The agents are numbered *1,2,3 ... k*
- Robots have internal memory.
- Local Visibility
- Communication:
  - ***Local*** : Face to face
  - ***Global*** : Wireless
- Each move costs one unit of Energy

Online vs Offline

OFFLINE

- Global Knowledge: G is known, v is known

  Find the best strategy for the k mobile agents

ONLINE

- No Prior Knowledge: G is unknown,
  - Local vs. Global Communication
  - With Return or without Return

Objective: Optimize Competitive Ratio!

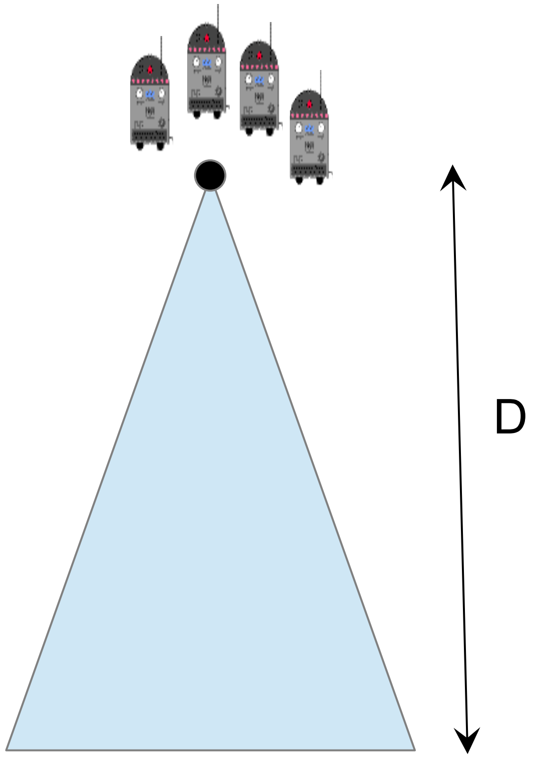
Optimization Issues

1) Exploration Time
   - Assuming full synchrony, reduce time to explore
   - Robots explore in parallel
2) Movements
   - Total moves = Total energy consumption
3) Energy Costs per Agent
   - Assume each robot has fixed energy budget
   - What should be the battery size
4) Maximize coverage
   - Assume fixed energy budget
   - How much can be explored?

Tree Exploration

- k agents at the root of T

How fast can they explore?



$$\mathbf{OPT} = \Omega(D + n / k)$$

[1] Fraigniaud, Gasieniec, Kowalski, and Pelc. *Collective tree exploration*. Networks 2006

Tree Exploration

- k agents at the root of T

<u>Algorithm</u>: [1]

- At each node u,
  - Distribute robots among incident edges (Round-robin)
- When an robot x completes a subtree,
  - It goes up to the parent node v;
  - Robot x is reassigned at node v;

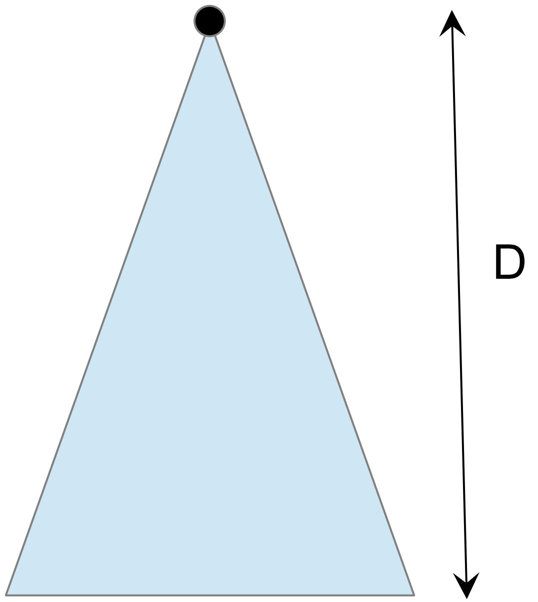
Exploration Time = O(D + n / log k) => Comp Ratio of **O(k/log k)**

[1] Fraigniaud, Gasieniec, Kowalski, and Pelc. *Collective tree exploration*. Networks 2006

When k is very large

When there are a large number of agents (k > n)

Is it possible to explore in Optimal O(D) time?



[2] D. Dereniowski, Y. Disser, A. Kosowski, D. Pajak, and P. Uznanski. Fast Collaborative Graph Exploration. Information and Computation 2015.

When **k** is very large

When there are a large number of agents ($k > n$)

Is it possible to explore in Optimal O(D) time?

Algorithm in [2]
- achieves O(D) exploration with $k = D.n^{(1+\epsilon)}$
- With local communication only
- Works for any arbitrary graph

Lower Bound: (For any $c > 0$)
Any algorithm using **$D.n^c$** agents requires at least **$D(1+1/c)$** time to explore.

[2] D. Dereniowski, Y. Disser, A. Kosowski, D. Pajak, and P. Uznanski. Fast Collaborative Graph Exploration. Information and Computation 2015.

Exploration with k agents

- k agents at the root of T

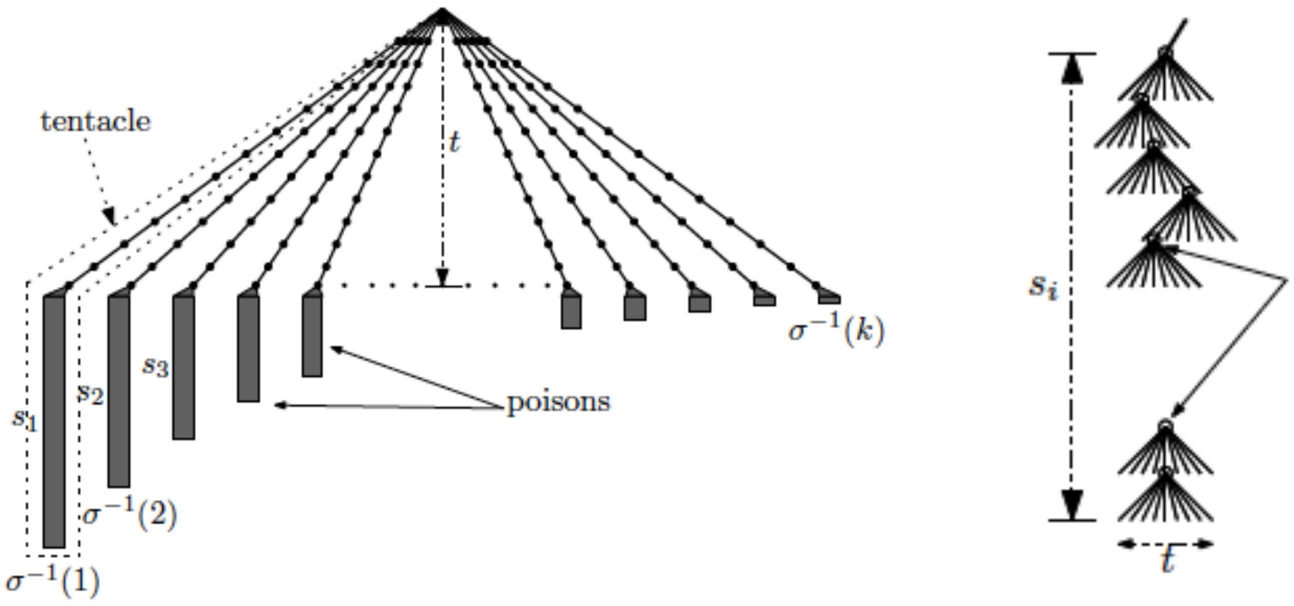
How fast can they explore?

- Fastest possible : $O(D + n/k)$
- There is an algorithm that explores in $O(D + n/\log k)$
- For large k, it is possible to explore in $O(D)$ time.
- For any given k, *what is the best possible CR?*

[3] M. Dynia, J. Lopuszanski, and C. Schindelhauer. Why robots need maps. In Proc. SIROCCO 2007

Lower Bound on Overhead

Theorem: Any online algorithm using k agents takes exploration time of $\Omega(\log k / \log \log k)$



[3] M. Dynia, J. Lopuszanski, and C. Schindelhauer. Why robots need maps. In Proc. SIROCCO 2007

Offline Exploration of Trees

Instance: An undirected tree $T = (V,E)$, $|V| = n$, a fixed node $r \in V$, an integer $k > 0$

Solution: tours $C\_1, C\_2, \ldots C\_k$, where $\cup C\_i = E$ and each tour contains the node r.

Goal: Minimize Cost = $\max\{ |C_i| : i = 1, \ldots k\}$

Optimal offline exploration is NP-hard! [1]

- Reduction from **3-PARTITION Problem**

[1] Fraigniaud, Gasieniec, Kowalski, and Pelc. *Collective tree exploration*. Networks 2006

Energy Optimization

Assumption: Each agent has energy budget **B**

(***An agent can traverse at most B edges***)

<u>Theorem</u>: Given a rooted tree T = (V,E) and a **fixed integer k**,
it is NP-hard to find the minimum budget **B**
such that k agents, each with budget B can explore T.

<u>Theorem</u>: Given a rooted tree T = (V,E) and a **fixed integer B**,
it is NP-hard to find minimum **k**
such that k agents, each with budget B can explore T.

Online Exploration

Assumption: Each agent has energy budget **B**

(***An agent can traverse at most B edges***)

For a fixed k (number of agents**),**

What is the minimum energy budget **B** ?

For a fixed B (energy budget**),**

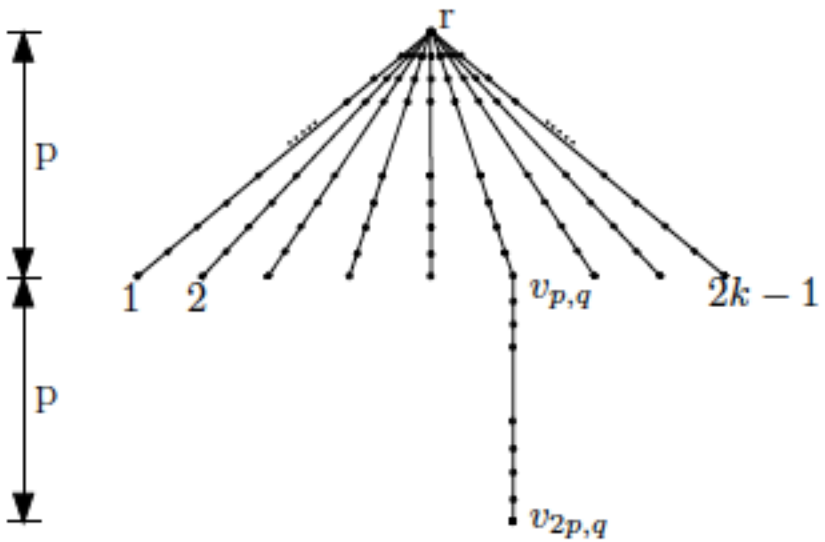
What is the minimum number of agents **k** ?

Online Exploration for fixed k

For a fixed k (number of agents),

What is the minimum energy budget **B** ?

<u>Lower Bound</u>: Any online algorithm has a competitive ratio of 3/2



Offline Algorithm:

B = 4p

Online Algorithm:

B >= 6p

Online Exploration for fixed k

For a fixed k **(**number of agents**)**,

What is the minimum energy budget **B** ?

Lower Bound: Any online algorithm has a competitive ratio of 3/2

Upper Bound: Competitive ratio = 4 - 1/k

Algorithm [3]

In each iteration,

- Send the next available agent to explore
- Perform DFS restricted to a given budget
- Increment the budget based on the depth reached.
- Agent returns to root.

[3] M. Dynia, J. Lopuszanski, and C. Schindelhauer. Why robots need maps. In Proc. SIROCCO 2007

Online Exploration for fixed B

For a fixed B (energy budget**),**
What is the minimum number of agents **k** ?

Two versions of the problem :

1. **With Return**
   - Each agent returns back to the root.
   - Equivalent to a single agent that refuel at the root
2. **Without Return**
   - Agents can terminate at any location
   - How do the agents communicate?
     - ➢ Global communication (wireless)
     - ➢ Local communication (meeting)

Piecemeal Exploration

Piecemeal Exploration

Assumption: An agent has fuel tank of size **B**
(*The agent can refuel at the root*)

<u>Objective</u>: Minimize total energy consumption
(or, # edges traversed)

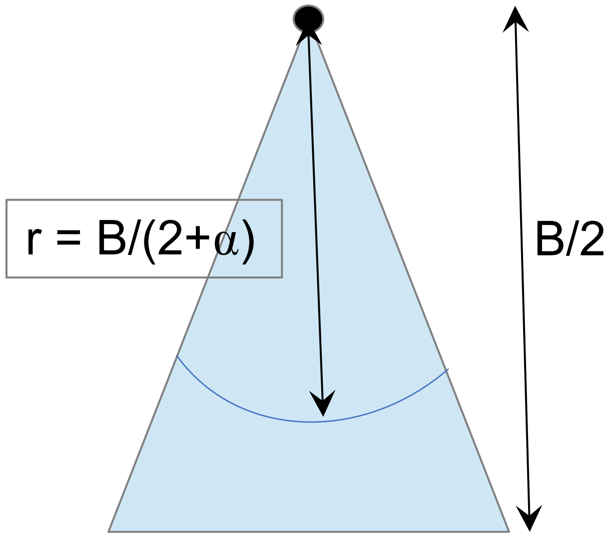
•Optimal = 2m moves

[4] : $O(m + n^{(1+\epsilon)})$

[5] : $O(m + n \log n)$

[6] : $O((m + n)/\alpha)$

All algorithms work for any arbitrary graph.



[4] Awerbuch, Betke, Rivest, and Singh. Piecemeal graph exploration by a mobile robot. Inf. Comput., 1999.

[5] Awerbuch and Kobourov. Polylogarithmic-overhead piecemeal graph exploration. Proc. COLT 1998

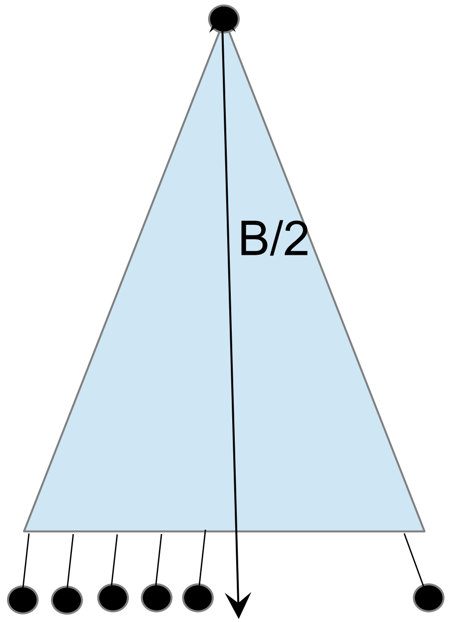
[6] Duncan, Kobourov, and Kumar. Optimal constrained graph exploration. ACM Trans. Algorithms 2006

Piecemeal Exploration

Objective: Explore all trees of height at most B/2
Assumption: Agents have energy budget B

<u>Optimization:</u>

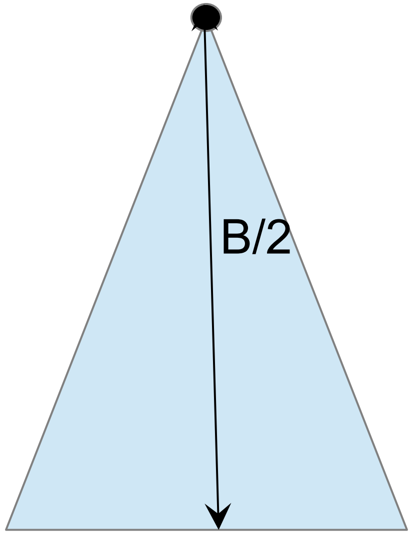
- Optimizing total energy is equivalent to optimizing number of agents (or number of refueling trips).
- Optimal offline algorithm may require
  O(B.m)
  traversals in worst case.
- Cost must be measured in terms of the competitive ratio.



Piecemeal Exploration

Assumption: An agent has energy budget **B**
(We want to explore all trees of height at most B/2)

Algorithm: Piecemeal DFS
1st Agent :
Perform DFS until remaining energy = depth;
Return to root;
2nd Agent :
Go to the next unvisited node in DFS seq.
Perform DFS until remaining energy = depth;
Return to root;
3rd Agent : …

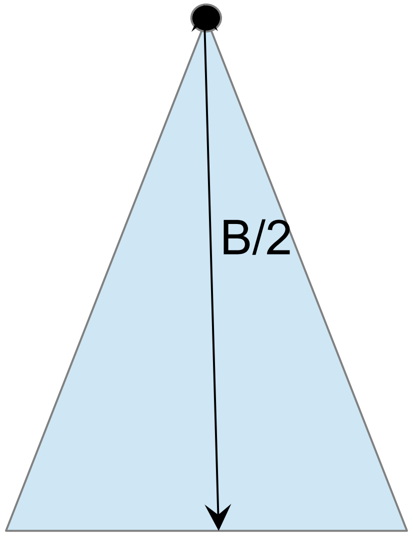


[7] Das, Dereniowski, and Uznanski. BA: Energy Constrained Depth First Search. ICALP 2018

Piecemeal Exploration

Assumption: An agent has energy budget **B**
(We want to explore all trees of height at most B/2)

<u>Theorem:</u> Algorithm Piecemeal DFS makes
10 . OPT
edge traversals (OPT is the offline optimal)

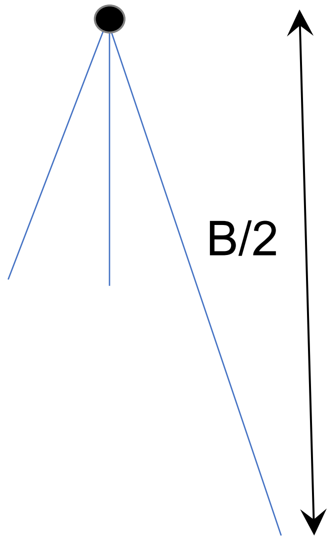


[7] Das, Dereniowski, and Uznanski. BA: Energy Constrained Depth First Search. ICALP 2018

Piecemeal Exploration

Assumption: An agent has energy budget **B**

(We want to explore all trees of height at most B/2)



Theorem: Algorithm Piecemeal DFS makes

10 . OPT

edge traversals (OPT is the offline optimal)

Lower Bound:
Any online algorithm must make 3/2 OPT edge traversals

[7] Das, Dereniowski, and Uznanski. BA: Energy Constrained Depth First Search. ICALP 2018

Exploration without Return

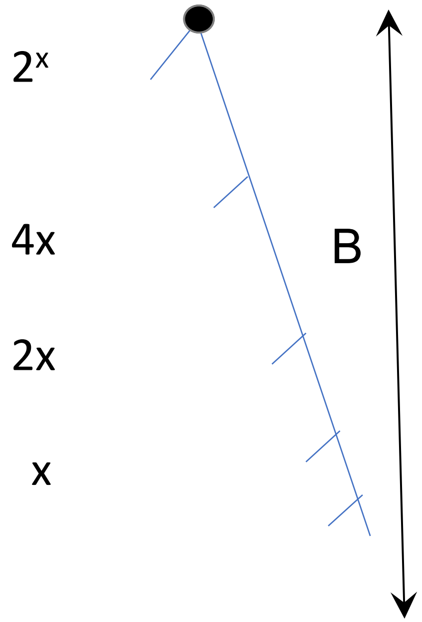
Objective: Explore all trees of height at most **B**
Assumption: Agents have energy budget B

Algorithm PDFS (with return)
may have a competitive ratio of **$\Omega(\log n)$**

$$B = n + x = 2^x.x + x$$

OPT uses one agent

PDFS uses $x = \Omega(\log n)$ agents

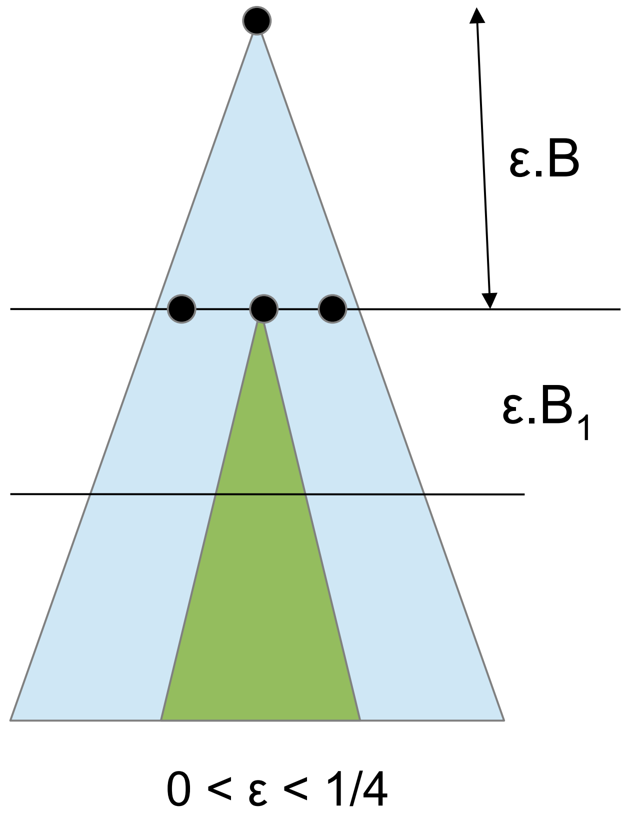


Remark: This algorithm cannot be implemented in the *local communication* model.

Level-by-Level Algorithm

<u>Algorithm</u> LCTE(ε)

- Recursive Algorithm
- Explore up to depth (ε.B)
- For each node at next level, recursively call the algorithm
- Number of levels = $\log_{(1/1-\varepsilon)} B$
- Within each level, agents perform depth restricted DFS.



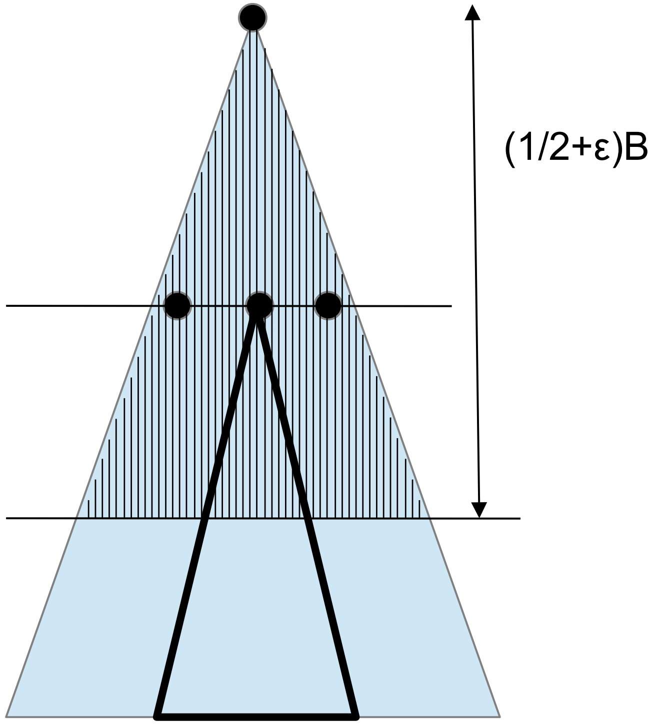
[8] Das, Dereniowski, and Karousatou. Collaborative Exploration of Trees by Energy-Constrained Mobile Robots. T. of Comp. Syst. 2018

The Look-ahead

- For each level(i), explore beyond the next level, with overlap of 1/2 B(i)
- For each node at level (i+1), the algorithm is called only if there are unexplored nodes in the sub-tree.

We can bound the number of agents used at each level as O(OPT).

Cost = **O(log B)** . OPT



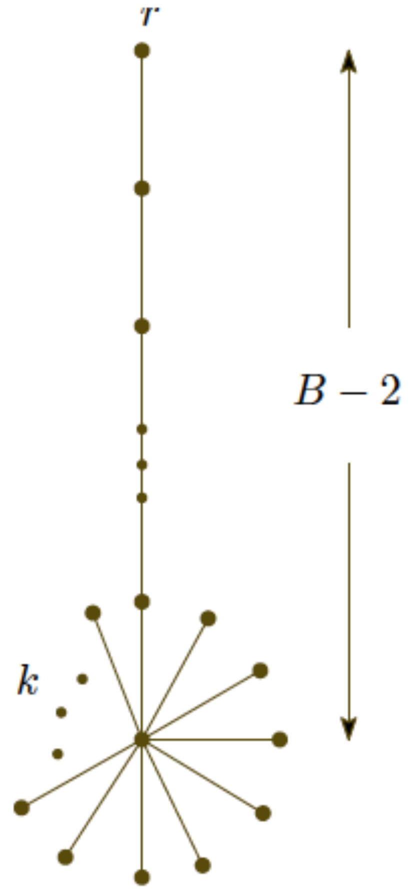
Communication is Local :
Use 4 helper agent per exploring agent

[8] Das, Dereniowski, and Karousatou. Collaborative Exploration of Trees by Energy-Constrained Mobile Robots. T. of Comp. Syst. 2018

Lower Bound on C.R.

Theorem: Any online algorithm for exploration without return for energy budget B has worst case CR of **Ω(log B)**

in the *local communication model*.



- The optimal algorithm uses exactly k agents.
- Any online algorithm requires **Ω(log B)** agents to communicate information to the root.
- By taking k=1, we get a competitive ratio of **Ω(log B)**

[8] Das, Dereniowski, and Karousatou. Collaborative Exploration of Trees by Energy-Constrained Mobile Robots. T. of Comp. Syst. 2018

Maximal Exploration

Assumptions:

1. Each agent has fixed energy budget **B**
2. There are a fixed number of agents, **k**

With limited resources how much can be explored?

Problem: (Maximal Exploration)

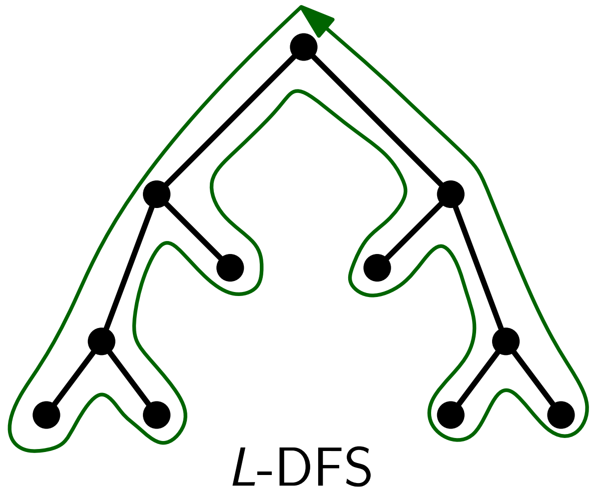
Given any rooted tree T = (V,E) and fixed **B, k**,

Maximize the number of visited nodes

$$\text{Competitive Ratio} = \text{Sup} \; \frac{\text{\#Nodes Explored by OPT}}{\text{\#Nodes Explored by A}}$$

Maximal Exploration

<u>Algorithm</u>:
1. The first agent performs a left-first DFS (LDFS)



[9] Bampas, Chalopin, Das, Hackfeld, Karousatou. Maximal Exploration of Trees with Energy-Constrained Agents, ArXiv preprint (1802.06636), 2018.

Maximal Exploration

Algorithm:

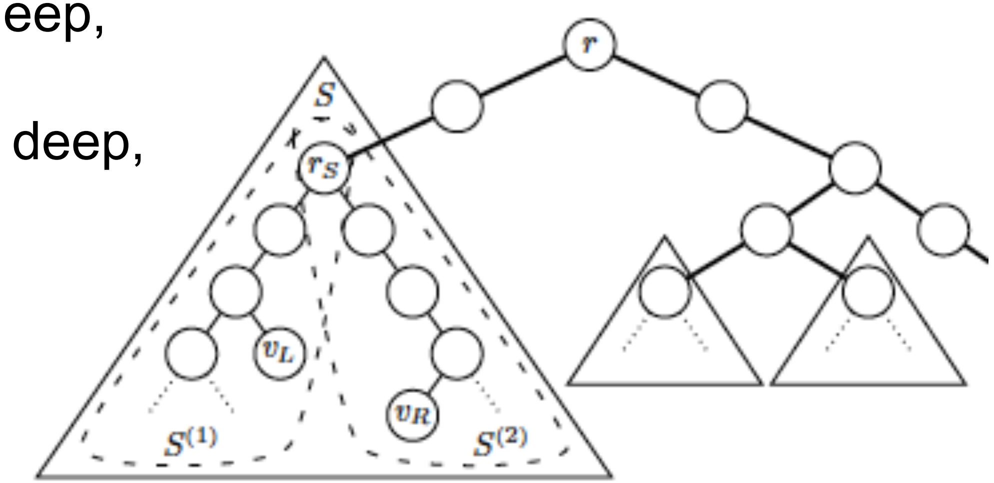
1. The first agent performs a left-first DFS (LDFS)
2. The second agent performs a right-first DFS (RDFS)

[9] Bampas, Chalopin, Das, Hackfeld, Karousatou. Maximal Exploration of Trees with Energy-Constrained Agents, ArXiv preprint (1802.06636), 2018.

Maximal Exploration

Algorithm:

1. The first agent performs a left-first DFS (LDFS)
2. The second agent performs a right-first DFS (RDFS)
3. For every other agent,
   - Pick the highest subtree S containing unexplored nodes
   - If v_L = leftmost(S) is not too deep,
   - Perform LDFS at v_L
   - If v_R = rightmost(S) is not too deep,
   - Perform RDFS at v_L
   - Else S into two subtrees.



Nodes Explored = 1/**3** . OPT

[9] Bampas, Chalopin, Das, Hackfeld, Karousatou. Maximal Exploration of Trees with Energy-Constrained Agents, ArXiv preprint (1802.06636), 2018.

Maximal Exploration

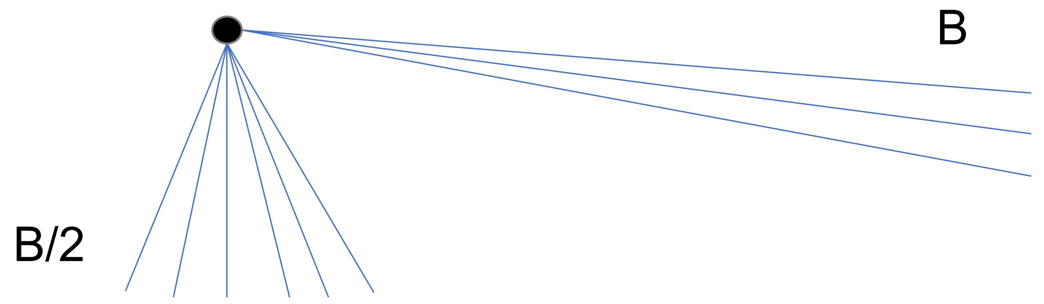
Lower Bound:
There is no online exploration algorithm with competitive ratio better than $(5+3\sqrt{17})/8 \approx 2.17$

[9] Bampas, Chalopin, Das, Hackfeld, Karousatou. Maximal Exploration of Trees with Energy-Constrained Agents, ArXiv preprint (1802.06636), 2018.

Maximal Exploration

Lower Bound:
There is no online exploration algorithm with competitive ratio better than $(5+3\sqrt{17})/8 \approx 2.17$

Simpler **Lower Bound** :
There is a tree where no online exploration algorithm that can explore more than ½ of the nodes explored by OPT.



[9] Bampas, Chalopin, Das, Hackfeld, Karousatou. Maximal Exploration of Trees with Energy-Constrained Agents, ArXiv preprint (1802.06636), 2018.

Conclusions

- Exploration by many agents offers many challenges, specially when proving efficiency under various optimization criteria.
- We consider the optimization of time, energy and movement. Other criteria could be agent memory, communication, randomization etc.
- There is a sharp difference between online exploration with or without return (O(1) vs O(log B)). Can additional capabilities reduce this?
- Many of the results are for **Trees**. How to extend to other graphs? Which other classes of graphs could facilitate efficient exploration?
- How to explore by teams of heterogeneous agents – agents with distinct speeds or energy consumptions?
- How could increase in visibility and communication help in collaborative exploration by many agents?

THANK YOU