`Exercise 1` `Collective Exploration`

FIGURE 1 – Can Astérix, Obélix, and Panoramix escape the maze faster collectively than if they were alone ?

A team of k mobile agents/robots, initially located at the root of an unknown tree, must traverse all its edges with a minimum number of movements. We consider a distributed model (agents communicate only by writing information on the nodes) and asynchronous (no guarantees on the relative speeds of the agents). More precisely, we consider the following model : DACTE (Distributed Asynchronous Collective Tree Exploration).

There are many ways to represent a tree in computer science. In this exercise, a tree is represented by a subset of $\cup_{n\in\mathbb{N}}\mathbb{N}^n$. This tree $T \subset \cup_{n\in\mathbb{N}}\mathbb{N}^n$ satisfies two properties :

— $() \in \mathbb{N}^0$ represents (by convention) the root of the tree $() \in T$,

— if a node $u = (u_1, \dots, u_d) \in T$, for $d \geq 1$, then its parent $p(u) = (u_1, \dots, u_{d-1}) \in T$.

The following terms will also be used. The node $u = (u_1, \dots, u_d) \in T$ is at *depth* d, denoted $d(u) = d$. The *depth* of the tree is defined as $D = \max_{u\in T} d(u)$. Its number of *nodes* is defined as $n = |T|$. And u_d is called the *port number* at $p(u)$ that gives access to u.

The exploration is divided into discrete rounds. Initially, at $t = 0$, all robots are located at the root of the unknown tree, and all registers are empty. At each round $t \in \mathbb{N}$, an omniscient adversary (who sees the entire situation) chooses a robot $i_t \in [k]$ to move. A robot's movement consists of the following consecutive instantaneous steps :

S1. The robot reads the whiteboard at its current position ;
S2. The robot observes the list of adjacent edges (identified by their 'port number') ;
S3. The robot writes to the whiteboard at its position and updates its internal memory ;
S4. The robot moves along an adjacent edge of its choice.

The exploration ends at the first round $t \in \mathbb{N}$ during which all nodes have been visited by at least one robot. The goal is to design algorithms that explore any tree with n nodes and depth D in at most $f(n, D)$ rounds, for some bivariate function $f(\cdot, \cdot)$.

1. Show that such a function must satisfy $f(n, D) \geq 2n - kD$.

2. For $k = 1$ and $k = 2$, present an algorithm with $f(n, D) = 2n$. Explain why this does not generalize to $k \geq 3$.

3. Show that in the model where all robots move along an edge at each round (synchronous model), an asynchronous guarantee in $f(n, D)$ implies a synchronous guarantee in $f(n, D)/k$.

Definition 1 (Locally Greedy Algorithm)**.** A locally greedy algorithm is such that each robot i maintains a target node $v_t(i) \in V$, and the moving robot follows these rules at time t :

R1. **If** adjacent to an unexplored edge, **then** the robot traverses an unexplored edge ;
R2. **Else** the robot moves towards its target ;
R3. The target is updated before the move if neither (R1.) nor (R2.) can apply (the target is said to be saturated).

4. Show that after M moves of a locally greedy algorithm, the number of edges explored by the algorithm is at least

$$\frac{1}{2}\left(M - \sum_{i\in[k]}\sum_{t<M} d(v_t(i), v_{t+1}(i))\right)$$

where $v_t(i)$ denotes the target of robot i at move t, and $d(\cdot,\cdot)$ is the distance in the underlying tree.

We now study a distributed algorithm called "Breadth First Depth Next" (BFDN). The algorithm is locally greedy and described as follows. When a robot saturates a target other than the root, its target is updated to be the root. When the robot saturates its target, which is the root, its target is updated using Algorithm 1.

5. Show that Algorithm 1 is correct, i.e., it assigns targets to robots saturating the root until the entire tree is explored. **Hint :** Show that all unexplored nodes are descendants of nodes in A.

6. Show that at any time $|A| \leq k$.

7. Show that there can be at most $\mathcal{O}(Dk\log(k))$ calls to Algorithm 1 before the exploration is complete. **Hint :** First, show that there are at most $\mathcal{O}(k^2)$ calls to Algorithm 1 before an increment of d, then refine this guarantee.

8. Show that BFDN explores any tree with n nodes and depth D in at most $2n+\mathcal{O}(k\log(k)D^2)$ iterations. Explain why it can be implemented with distributed communication.

Open question : Are there *collective* exploration algorithms for *graphs* ? More specifically, is there an algorithm for k robots that explores any graph with m edges and a diameter D in $2m + \mathcal{O}(f(k,D))$ iterations ?

Algorithm 1 `BFDN` “Breadth-First Depth-Next” (central scheduler at the root)

$d \leftarrow 0$ ▷ current depth of work
$A \leftarrow \{\mathrm{r}\}$ ▷ list of possible targets at depth d
$R \leftarrow \emptyset$ ▷ elements of A from which a robot has returned (to r)
while $A \neq \emptyset$ and a robot i saturates the root **do**
    $a \leftarrow$ last node in A visited by robot i
    $R \leftarrow R \cup \{a\}$
    `Update` $C^1(a)$, the list of children of a from which a robot has returned (to a)
    `Update` $C^2(a)$, the list of children of a from which no robot has returned (to a)
    if $A \setminus R = \emptyset$ **then**
        $d \leftarrow d+1$
        $A \leftarrow \cup_{a\in A} C^2(a)$
    end if
    `Assign to robot` i a target from $A \setminus R$ with the minimal load (the load of a target is the number of robots already assigned to it).
end while