# Ring Exploration with Oblivious Myopic Robots

Ajoy Datta, Anissa Lamani, Lawrence L. Larmore, Franck Petit

# Ring Exploration with Oblivious Myopic Robots

**(Invited Paper)**

Ajoy K. Datta[1], Anissa Lamani[2], Lawrence L. Larmore[1], and Franck Petit[3]

[1] School of Computer Science, University of Nevada Las Vegas, USA
[2] MIS, Universit de Picardie Jules Verne Amiens, France
[3] LIP6, INRIA, CNRS, UPMC Sorbonne Universities, France

**Abstract.** Many potential applications exist for multi-robot systems in various contexts related to safety and security. Terminating exploration is a basic building block for many of these applications. It requires that starting from a configuration where no two robots occupy the same location, every location needs to be visited by at least one robot, with the additional constraint that all robots eventually stop moving. We consider weak autonomous mobile robots, *i.e.*, the are anonymous, oblivious, disoriented, and unable to (directly) communicate together. They are also myopic, *i.e.*, they are cannot see beyond a certain fixed distance $\phi$. We consider strong myopia that is, a robot can only sense robots located at its own and at its immediate neighboring nodes—$\phi = 1$. We prove that within such settings, no deterministic exploration is possible in the semi-synchronous model. We then address the synchronous model and show that no deterministic exploration protocol solves the problem with less than five robots when $n > 6$. We provide optimal (in terms of number of robots) deterministic algorithms in the fully synchronous model for both cases $3 \leq n \leq 6$ and $n > 6$.

## 1 Introduction

Many potential applications exist for multi-robot systems in various contexts related to safety and security, for instance patrolling in adversarial environments, exploration of awkward environments, environmental monitoring, intelligence activities, fighting fire in a rescue scenario, and many other risky tasks for humans. Beyond technological challenges to overcome, it is crucial to devise distributed algorithms capable of coordinating robots without central control [5].

Nevertheless, environmental or technological requirements may lead to perform these tasks using as less resources as possible. Indeed, in unsafe environment, part of the fleet may be unusable. So, it is important to know the minimum number of robots that are required to accomplish a given task. However, in a hostile environment, even using the fewest number of robots, capacities of robots may be reduced. So, it is expected to perform certain tasks with weakened robots. For instance, robots could have nothing else than vision to communicate together. Numerous realistic scenarios are easily deciphered such as faulty wireless devices or robots evolving in zones where wireless communication is scrambled or forbidden. Equipment other than the means of communication may also be faulty, unusable, or even non-existent. This may be the case, for instance, sensors, vision devices, and guidance systems like GPS or compass.

The above scenarios greatly motivate the design of coordination algorithms that use the least amount of robots, themselves using the least amount of resources. Furthermore, this approach offers other benefits, namely reducing manufacturing costs and operating expenses.

The problem of exploring a finite discrete space by autonomous mobile robots is a basic building block for many applications. Space to explore is partitioned into a finite number of locations represented by a graph, where nodes represent indivisible locations that can be sensed by the robots, and where edges represent the possibility for a robot to move from one location to the other, *e.g.*, a building, a town, a factory, or a mine. We address the *terminating exploration* problem which requires that starting from a configuration where no two robots occupy the same node, every node needs to be visited by at least one robot, with the additional constraint that all robots eventually stop moving.

We assume robots having weak capabilities: they are *uniform*—meaning that all robots follow the same algorithm—, *anonymous*—meaning that no robot can distinguish any two other robots—, *oblivious*—they have no memory of any past behavior of themselves or any other robot—, and *disoriented*—they have no labeling of direction. Furthermore, the robots have no (direct) means of communicating with each other. However, robots are endowed with visibility sensors enabling to see robots located on nodes.

Terminating exploration under such weak assumptions has been investigated so far. In [3], it is shown that, in general, $\Omega(n)$ robots are necessary to explore a tree network of $n$ nodes. In [4], it is proved that no deterministic exploration is possible on a ring when the number of robots $k$ divides the number of nodes $n$. In the same paper, the authors proposed a deterministic algorithm that solves the problem using at least 17 robots, provided that $n$ and $k$ are co-prime. In [2], it is shown that no algorithm (probabilistic or deterministic) can explore a ring with fewer than four robots. In the same paper, the authors provide a probabilistic algorithm that solves the problem on a ring of size $n > 8$ that is optimal in terms of number of robots. In [6], the authors reduce the gap in the deterministic case between a large upper bound ($k \geq 17$) and a small lower bound ($k > 3$) by showing that 5 robots are necessary and sufficient in the case that the size of the ring is even, and that 5 robots are sufficient when the size of the ring is odd.

In this paper, we consider terminating exploration algorithms for ring networks of $n$ nodes that use $k$ weak robots. We add another constraint: *myopia*. A myopic robot has limited visibility, *i.e.*, it cannot see the nodes located beyond a certain fixed distance $\phi$. The stronger the myopia is, the smaller $\phi$ is. If $\phi = 1$, then a robot can sense robots located on its own node and at its immediate neighboring nodes. To the best of our knowledge, all previous results for discrete versions of the exploration problem assume unlimited visibility ($\phi = \infty$), *i.e.*, the whole graph is seen by each robot.

We consider the deterministic solubility of exploring a ring assuming $\phi = 1$, *i.e.*, a robot can only sense robots located at its own and at neighboring nodes. Our contribution is threefold[4]. We first prove that assuming $\phi = 1$, no deterministic exploration is possible in the semi-synchronous model. The result

[4] Detailed proofs are given in [1].

is also valid for the asynchronous model and holds for any $k < n$ where $k$ is the number of robots and $n$ is the number of nodes in the ring. Next, we address the synchronous model and show that no deterministic exploration algorithm solves the problem with less than 5 robots when $n > 6$. Finally, we provide optimal (in terms of number of robots) deterministic algorithms in the synchronous model for both cases $3 \leq n \leq 6$ and $n > 6$.

## 2 Preliminaries

We assume that the graph is a *ring* of $n$ nodes, $u_0, \ldots, u_{n-1}$ — indices $0 \ldots i \ldots n-1$ are assumed to be modulo $n$ and used for notation purposes only. The nodes are *anonymous* and the ring is *unoriented*, *i.e.*, given two neighboring nodes $u$, $v$, there is no kind of explicit or implicit labeling allowing to determine whether $u$ is on the right or on the left of $v$. Zero, one, or more robots can be located on a node. The number of robots located on a node $u_i$ at instant $t$ is called *multiplicity* of $u_i$ and is denoted by $M_i(t)$ (or simply $M_i$, if $t$ is understood). If $M_i(t) > 1$ then, we say that there is an $M_i(t).tower$ (or simply a *tower*) at $u_i$ at instant $t$. Each robot is equipped with an abstract device allowing to measure node multiplicity. Since we assume no ring orientation and $\phi = 1$, the view of any robot $r$ located on a node $u_i$ is a sequence $s$ of 3 integers $x_{-1}, x_0, x_1$ such that $x_0 = M_i$ and either $x_{-1} = M_{i-1}$ and $x_1 = M_{i+1}$, or $x_{-1} = M_{i+1}$ and $x_1 = M_{i-1}$. Robots operate in three phase cycles: Look, Compute and Move (L-C-M). During the Look phase, a robot $r$ located $u_i$ takes a snapshot $s$ of its environment given by the output of its multiplicity sensors. Then, using $s$, $r$ computes a destination to move to, *i.e.*, either $u_{i-1}$, $u_i$, or $u_{i+1}$. In the last phase (move phase), $r_j$ moves to the target destination computed in the previous phase.

A configuration is the state of the ring at a given time instant $t$. A configuration is said to be *towerless* if $M_i \leq 1$ for all $i \in [0..n-1]$. A *block* $B$ is any maximal elementary sequence of nodes $u_i \ldots u_{i+l}$ $(l \geq 0)$ for which each node contains at least one robot. The *size* (respectively, *length*) of $B$, is the number of robots (resp. nodes) included in $B$. If both the length and the size are equal to 1, then the robot is said to be *isolated*.

At each step $t$, a non-empty subset of robots is *selected* by the *scheduler*, which is viewed as an abstract external entity. We assume a *fair* scheduler, *i.e.*, every robot is activated infinitely often during a computation. We consider three computational models: ($i$) The *semi-synchronous* system, ($ii$) the *synchronous* model, and ($iii$) the *asynchronous* model. In the former model, at every time instant $t$, the scheduler selects a subset of robots so that every robot that is selected executes the full cycle L-C-M instantaneously between $t$ and $t+1$. The synchronous model is similar to the semi-synchronous model, except that the scheduler selects all robots at each step. In the asynchronous model, cycles L-C-M are performed asynchronously for each robot, *i.e.*, the time between Look, Compute, and Move operations is finite but unbounded, and is decided by the scheduler for each action of each robot.

Each rule in the algorithm is presented in the following manner: $<$ *Label* $>$ $<$ *Guard* $>$ :: $<$ *Statement* $>$. The guard is a possible sequence $s = x_{-1}, x_0, x_1$ provided by the sensor of a robot $r$. The statement describes the action to be performed by $r_j$. There are only two possible actions: $(i)$ $\rightarrow$, meaning that $r$ moves towards the node $u_{i+1}$, $(ii)$ $\leftarrow$, meaning that $r$ moves towards the node $u_{i-1}$. Note that when the view of $r$ is symmetric, the scheduler chooses the action to be performed. In this case, we write the statement as follows: $\leftarrow \vee \rightarrow$.

## 3 Negative Results

*Semi-Synchronous System.* Clearly, if $n = k$, then the exploration is trivially solved with the empty algorithm $\mathcal{E}$, *i.e.*, no robot moves. So, in the following, we assume that $0 < k < n$. In this section, we assume that the robots know $k$ and $n$. Let us borrow the following result from [4]:

**Theorem 1 ([4]).** *Let $k < n$. If $k|n$, then the exploration of an $n$-node ring with $k$ robots is not possible.*

The proof of Theorem 1 is also trivially valid for any $\phi \leq n$. Therefore, in the following we consider that a deterministic semi-synchronous algorithm $\mathcal{P}$ solves the exploration problem provided that $2 \leq k < n$ and $k$ does not divide $n$. Let $\Pi$ be the class of deterministic semi-synchronous algorithms that solve the exploration problem assuming a fair scheduler with $\phi = 1$ and $2 \leq k < n$. In the following, $\gamma_0$ refers to an initial configuration. Recall that $\gamma_0$ is towerless, *i.e.*, the multiplicity of any node of the ring is less than or equal to 1. So, block size and block length are equivalent in $\gamma_0$. Since robots are able to sense only at distance 1, only the four following rules are possible in $\gamma_0$:

| | |
|---|---|
| $\mathcal{R}_{\texttt{sgl}}$: $0(1)0 :: \rightarrow \vee \leftarrow$ | $\mathcal{R}_{\texttt{out}}$: $0(1)1 :: \leftarrow$ |
| $\mathcal{R}_{\texttt{in}}$: $0(1)1 :: \rightarrow$ | $\mathcal{R}_{\texttt{swp}}$: $1(1)1 :: \rightarrow \vee \leftarrow$ |

**Lemma 1.** *For any $2 \leq k < n$, no algorithm exists in $\Pi$ such that every robot is isolated in $\gamma_0$.*

The above lemma is simply proven by contradiction, assuming that the scheduler behaves synchronously. It shows that even assuming that $2 \leq k < n$ and $k$ does not divide $n$, there are initial configurations from which the problem is not solvable. In other words, if such an algorithm exists, then it works starting from some particular configurations. In particular, they contain at least one block of length 2 and $k$ must be strictly greater than $\lfloor \frac{n}{2} \rfloor$. The next lemma is shown by contradiction, considering all possible moves in $\gamma_0$:

**Lemma 2.** *For every $\mathcal{P} \in \Pi$, $\mathcal{P}$ includes $\mathcal{R}_{\texttt{in}}$ and does not include neither $\mathcal{R}_{\texttt{swp}}$ nor $\mathcal{R}_{\texttt{out}}$.*

Let us call a *symmetric $x$.tower sequence* (an $S^x$-sequence for short), a sequence of occupied nodes such that the extremities of the sequence contain an $x$.tower. Observe that a tower containing at least 2 robots (by definition), $x$ is greater than or equal to 2. Also, since the robots can only see at distance 1, the tower is only seen by its neighboring robots.

**Lemma 3.** *For every $\mathcal{P} \in \Pi$, for every $k$ ($5 \leq k < n$), there exist executions of $\mathcal{P}$ leading to a configuration containing $S^2$-sequences, isolated robots and blocks of size strictly smaller than 5.*

Consider robots being located at the border of a block. Let $x \geq 1$ be the number of robots located on the border node and the following generic rules:

$\mathcal{T}\alpha(x)$: $0(x)1 :: \leftarrow$ $\mathcal{T}\beta(x)$: $0(x)1 :: \rightarrow$

$\mathcal{T}\gamma(x)$: $x(1)1 :: \leftarrow$ $\mathcal{T}\delta(x)$: $x(1)1 :: \rightarrow$

Remark that Rule $\mathcal{R}_{\mathtt{in}}$ corresponds to Rule $\mathcal{T}\beta(1)$. Also, note that since $x$ robots are located on the border node of a block, the local configuration for both $\mathcal{T}\gamma(x)$ and $\mathcal{T}\delta(x)$ is $0x11$. Similarly, define the generic rule $\mathcal{T}_{\mathtt{sgl}}(y)$ ($y \geq 2$) as follow: $0(y)0 ::\leftarrow \vee \rightarrow$. Again by considering every execution starting from configurations containing $S^2$-sequences:

**Lemma 4.** *For every $\mathcal{P} \in \Pi$, for every $x \geq 1$, $\mathcal{P}$ includes $\mathcal{T}\beta(x)$ only.*

It follows from Lemma 4 that $\mathcal{P}$ cannot ensure both progression and termination properties. Hence:

**Theorem 2 ($\Pi = \emptyset$).** *No deterministic exploration algorithm exists in the semi-synchronous model for $\phi = 1$, $n > 1$, and $1 \leq k < n$.*

**Corollary 1.** *No deterministic exploration algorithm exists in the asynchronous model for $\phi = 1$, $n > 1$, and $1 \leq k < n$.*

*Synchronous System.* Remark that Lemma 1 still holds in synchronous settings. So, in the initial configuration $\gamma_0$, there exists at least one block of size greater than or equal to 2. We show the following theorem by contradiction over $k$:

**Theorem 3.** *Let $\mathcal{P}$ be a synchronous exploration algorithm for $\phi = 1$ and $2 \leq k < n$. If $n \geq 7$, then, $k$ must be greater than or equal to 5.*

## 4 Synchronous Algorithms

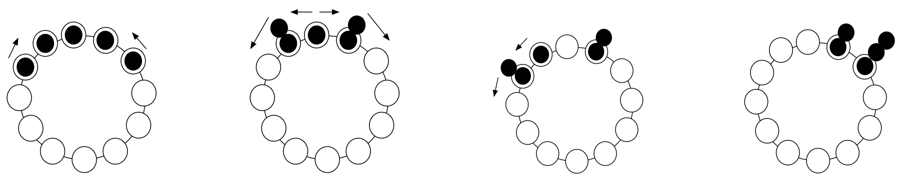
In the following, we present two optimal deterministic algorithms that solve the exploration problem in the synchronous model. The former is a general algorithm that uses 5 robots and works on any ring of size $n \geq 7$. The latter works for small rings ($3 \leq n \leq 6$) using $(n-1)$ robots, except for the case $n = 6$ that needs 4 robots only.

*General Algorithm for $k = 5$ and $n \geq 7$.* The idea of Algorithm 1 is as follow:

**Algorithm 1** Synchronous Exploration for $n \geq 7$

| | |
|---|---|
| 1A1: $0(1)1 :: \rightarrow$ | // *Towards occupied neighbor* |
| 1A2: $2(1)2 :: \rightarrow \vee \leftarrow$ | // *Towards any neighbor* |
| 1A3: $0(2)1 :: \leftarrow$ | // *Towards empty neighbor* |
| 2A4: $2(1)0 :: \leftarrow$ | // *Towards the tower* |

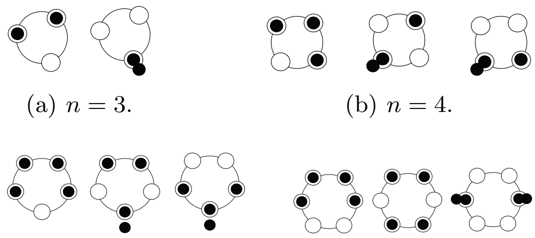
The robots that are at the border of the block are the only ones that are allowed to move in the initial configuration, $\gamma_0$. They move on their adjacent occupied node—Rule $1A1$, refer to Figure 1, $(a)$. Since the system is synchronous, the next configuration, $\gamma_1$, contains a single robot surrounded by two 2.towers. In the next step, the towers move in the opposite direction of the single robot (Rule $1A3$) and the single robot moves towards one of the two towers—Rule $1A2$, Figure 1, $(b)$. Note that the resulting configuration $\gamma_2$—Figure 1, $(c)$—provides an orientation of the ring. From there, the single 2.tower are the landmark allowing to detect termination and the three other robots explore the ring synchronously. After $n-4$ steps, $n-4$ nodes are visited by the 3 robots and the system reaches $\gamma_{n-2}$. Finally, by performing Rule $2A4$, the single robot create a 3.tower—Figure 1, $(d)$. This marks the end of the exploration in $\gamma_{n-1}$ in which each robot is awake of the termination.



*(a)* *(b)* *(c)* *(d)*

**Fig. 1.** Overview of Algorithm 1 ($n \geq 7$).

*Specific Algorithm for* $3 \leq n \leq 6$. The formal description of the algorithm is given in Algorithm 2. Instances for each value of $n$ are given in Figure 2. The robots in this case detect the end of the exploration task if they are either part of a 2.tower or neighbors of a 2.tower. The idea of the algorithm is the following: For the cases where $3 \leq n \leq 5$, $k = n-1$ robots are necessary to perform the exploration task. The robots that are at the extremities of the block are the ones allowed to move. Since $k = n-1$, once they move, a 2.tower is created. If the reached configuration contains an isolated robot (the case where $n = 4$) then, this robot is the only one allowed to move. Its destination is one of its adjacent empty nodes. Once it moves it becomes neighbor of the 2.tower. If $n = 6$, the same algorithm works with 4 robots assuming that they belong to the same block.



(a) $n = 3$. (b) $n = 4$. (c) $n = 5$. (d) $n = 6$ ($k = 4$).

**Fig. 2.** Specific Algorithm for $3 \leq n \leq 6$.

**Algorithm 2** Synchronous Exploration for $3 \leq n \leq 6$

1A'1: 0(1)1 :: $\leftarrow$ // *Towards empty neighbor*
1A'2: 0(1)0 :: $\leftarrow \vee \rightarrow$ // *Towards any neighbor*

## 5 Conclusion

We studied the terminating exploration of anonymous, unoriented rings by a team of oblivious robots. The assumptions of unlimited visibility made in previous works allow to focus on overcoming the computational weaknesses of robots. In this paper, we added one more weakness: Myopia. We consider strong myopia that is, a robot cannot sense farther than its immediate neighboring nodes. We studied the problem for both asynchronous and synchronous settings. We proved that deterministic exploration is possible if the system is synchronous only. Next, we provided deterministic algorithms for synchronous systems that are optimal in terms of number of robots.

The reader should have noticed that the algorithm proposed in Section 4 for $n \geq 7$ can easily be generalized to any odd number $k \geq 5$ of robots. Indeed, the exploration works by replacing 2 by $\lfloor \frac{k}{2} \rfloor$ in the guards of Action $1A2$ to Action $2A4$ of Algorithm 1, where the parenthesis now means "any positif integer smaller than or equal to $\lfloor \frac{k}{2} \rfloor - 1$". The guard of Action $1A1$ must be replaced by $0(1..\lfloor \frac{k}{2} \rfloor - 1)1$. The case where $k$ is even is more sophisticated and requires a specific algorithm [1].

## References


1. Ajoy K. Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. Ring exploration by oblivious agents with local vision. In *33rd International Conference on Distributed Computing (ICDCS)*, 2013. To appear.
2. Stéphane Devismes, Franck Petit, and Sébastien Tixeuil. Optimal probabilistic ring exploration by semi-synchronous oblivious robots. In *16th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 195–208, 2009.
3. Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots. *Theor. Comput. Sci.*, 411(14-15):1583–1598, 2010.
4. Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. *Algorithmica*, 65(3):562–583, 2013.
5. Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Oblivious Mobile Robots.* Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2012.
6. Anissa Lamani, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. Optimal deterministic ring exploration with oblivious asynchronous robots. In *17th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 183–196, 2010.