Optimal Probabilistic Ring Exploration by Semi-Synchronous Oblivious Robots*

Stéphane Devismes[1], Franck Petit[2], and Sébastien Tixeuil

[1] VERIMAG UMR 5104, Université Joseph Fourier, Grenoble (France),
stephane.devismes@imag.fr
[2] INRIA, LIP UMR 5668, Université de Lyon / ENS Lyon (France)
franck.petit@ens-lyon.fr
[3] Université Pierre et Marie Curie - Paris 6, INRIA Grand Large, France
sebastien.tixeuil@lip6.fr

Abstract. We consider a team of k identical, oblivious, semi-synchronous mobile robots that are able to sense (*i.e.*, view) their environment, yet are unable to communicate, and evolve on a constrained path. Previous results in this weak scenario show that initial symmetry yields high lower bounds when problems are to be solved by *deterministic* robots. In this paper, we initiate research on probabilistic bounds and solutions in this context, and focus on the *exploration* problem of anonymous unoriented rings of any size. It is known that $\Theta(\log n)$ robots are necessary and sufficient to solve the problem with k deterministic robots, provided that k and n are coprime. By contrast, we show that *four* identical probabilistic robots are necessary and sufficient to solve the same problem, also removing the coprime constraint. Our positive results are constructive.

1 Introduction

We consider autonomous robots that are endowed with visibility sensors (but that are otherwise unable to communicate) and motion actuators. Those robots must collaborate to solve a collective task, namely *exploration*, despite being limited with respect to input from the environment, asymmetry, memory, etc. In this context, the exploration tasks requires every possible location to be visited by at least one robot, with the additional constraint that all robots stop moving after task completion.

Robots operate in *cycles* that comprise *look*, *compute*, and *move* phases. The look phase consists in taking a snapshot of the other robots positions using its visibility sensors. In the compute phase a robot computes a target destination based on the previous observation. The move phase simply consists in moving toward the computed destination using motion actuators.

The robots that we consider here have weak capacities: they are *anonymous* (they execute the same protocol and have no mean to distinguish themselves

* This work has been supported in part by the ANR projets *R-Discover* (08-ANR-CONTINT) and *SHAMAN*.

from the others), *oblivious* (they have no memory that is persistent between two cycles), and have no compass whatsoever (they are unable to agree on a common direction or orientation).

Related works. The vast majority of literature on coordinated distributed robots considers that those robots are evolving in a *continuous* two-dimentional Euclidian space and use visual sensors with perfect accuracy that permit to locate other robots with infinite precision, *e.g.*, [1–6].

Several works investigate restricting the capabilities of both visibility sensors and motion actuators of the robots, in order to circumvent the many impossibility results that appear in the general continuous model. In [7, 8], robots visibility sensors are supposed to be accurate within a constant range, and sense nothing beyond this range. In [8, **?**], the space allowed for the motion actuator was reduced to a one-dimentional continuous one: a ring in [8], an infinite path in [**?**].

A recent trend was to shift from the classical continuous model to the *discrete* model. In the discrete model, space is partitioned into a *finite* number of locations. This setting is conveniently represented by a graph, where nodes represent locations that can be sensed, and where edges represent the possibility for a robot to move from one location to the other. Thus, the discrete model restricts both sensing and actuating capabilities of every robot. For each location, a robot is able to sense if the location is empty or if robots are positioned on it (instead of sensing the exact position of a robot). Also, a robot is not able to move from a position to another unless there is explicit indication to do so (*i.e.*, the two locations are connected by an edge in the representing graph). The discrete model permits to simplify many robot protocols by reasoning on finite structures (*i.e.*, graphs) rather than on infinite ones. However, as noted in most related papers [9–12], this simplicity comes with the cost of extra symmetry possibilities, especially when the authorized paths are also symmetric (indeed, techniques to break formation such as those of [5] cannot be used in the discrete model).

Assuming visibility capabilities, the two main problems that have been studied in the discrete robot model are gathering [9, 10] and exploration [11, 12]. For gathering, both breaking symmetry [9] and preserving symmetry are meaningful approaches. For exploration, the fact that robots need to stop after exploring all locations requires robots to "remember" how much of the graph was explored, *i.e.*, be able to distinguish between various stages of the exploration process since robots have no persistent memory. As configurations can be distinguished only by robot positions, the main complexity measure is then the number of robots that are needed to explore a given graph. The vast number of symmetric situations induces a large number of required robots. For tree networks, [12] shows that $\Omega(n)$ robots are necessary for most n-sized tree, and that sublinear robot complexity (actually $\Theta(\log n/ \log\log n)$) is possible only if the maximum degree of the tree is 3. In uniform rings, [11] proves that the necessary and sufficient number of robots is $\Theta(\log n)$, although it proposes an algorithm that works with an additional assumption: the number k of robots and the size n of the ring are coprime. Note that all previous approaches in the discrete model are *determinis-*

tic, *i.e.*, if a robot is presented twice the same situation, its behavior is the same in both cases.

Our contribution. In this paper, we consider the *semi-synchronous model* introduced in [13]. It is straighforward to see that the necessary conditions and bounds exposed in [11] for the deterministic exploration still hold in the semi-synchronous model. Here we propose to adopt a *probabilistic* approach to lift constraints and to obtain tigher bounds. By constrast with the deterministic approach, we show that *four* identical probabilistic robots are necessary and sufficient to solve the exploration problem in any anonymous unoriented ring of size $n > 8$, also removing the coprime constraint between the number of robots and the size of the ring. Our negative result show that for any ring of size at least four, there cannot exist any protocol with three robots in our setting, even if they are allowed to make use of probabilistic primitives. Our positive results are constructive, as we present a randomized protocol with four robots for any ring of size more than eight.

Outline. The remaining of the paper is divided as follows. Section 2 presents the system model that we use throughout the paper. Section 3 provides evidence that no three probabilistic robots can explore every ring, while Section 4 presents our protocol with four robots. Section 5 gives some concluding remarks.

For space consideration, several technical proofs are omitted, see the technical report for details ([14], `http://hal.inria.fr/inria-00360305/fr/`).

2 Model

Distributed System. We consider systems of autonomous mobile entities called *agents* or *robots* evolving into a *graph*. We assume that the graph is a *ring* of n nodes, u_0,..., u_{n-1}, *i.e.*, u_i is connected to both u_{i-1} and u_{i+1} — every computation over indices is assumed to be modulus n. The indices are 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 labelling allowing to determine whether u is on the right or on the left of v. Operating in the ring are $k \leq n$ anonymous robots.

A *protocol* is a collection of k *programs*, one operating on each robot. The program of a robot consists in executing *Look-Compute-Move cycles* infinitely many times. That is, the robot first observes its environment (Look phase). Based on its observation, a robot then (probabilistically or deterministically) decides — according to its program — to move or stay idle (Compute phase). When a robot decides a move, it moves to its destination during the Move phase.

The robots do not communicate in an explicit way; however they see the position of the other robots and can acquire knowledge from this information. We assume that the robots cannot remember any previous observation nor computation performed in any previous step. Such robots are said to be *oblivious* (or *memoryless*). The robots are also *uniform* and *anonymous*, i.e, they all have the same program using no local parameter (such that an identity) allowing to differentiate any of them.

Computations. We consider a *semi-synchronous* model similar to the one in [13]. In this model, time is represented by an infinite sequence of instants 0, 1, 2, ... At every instant $t \geq 0$, a non-empty subset of robots is activated to execute a cycle. The execution of each cycle is assumed to be *atomic*: Every robot that is activated at instant t instantaneously executes a full cycle between t and $t+1$. Atomicity guarantees that at any instant the robots are on some nodes of the ring but not on edges. Hence, during a Look phase, a robot sees no robot on edges.

We assume that during the Look phase, every robot can perceive whether several robots are located on the same node or not. This ability is called *Multiplicity Detection*. We shall indicate by $d_i(t)$ the multiplicity of robots present in node u_i at instant t. More precisely $d_i(t) = j$ indicates that there are j robots in node u_i at instant t. If $d_i(t) \geq 2$, then we say that there is a *tower* in u_i at instant t (or simply there is a *tower* in u_i when it is clear from the context). We say a node u_i is *free at instant* t (or simply *free* when it is clear from the context) if $d_i(t) = 0$. Conversely, we say that u_i is *occupied at instant* t (or simply *occupied* when it is clear from the context) if $d_i(t) \neq 0$.

Given an arbitrary orientation of the ring and a node u_i, $\gamma^{+i}(t)$ (respectively, $\gamma^{-i}(t)$) denotes the sequence $\langle d_i(t)d_{i+1}(t)\dots d_{i+n-1}(t)\rangle$ (resp., $\langle d_i(t)d_{i-1}(t)\dots d_{i-(n-1)}(t)\rangle$). The sequence $\gamma^{-i}(t)$ is called *mirror* of $\gamma^{+i}(t)$ and conversely. Since the ring is unoriented, agreement on only one of the two sequences $\gamma^{+i}(t)$ and $\gamma^{-i}(t)$) is impossible. The (unordered) pair $\{\gamma^{+i}(t), \gamma^{-i}(t)\}$ is called the *view* of node u_i at instant t (we omit "at instant t" when it clear from the context). The view of u_i is said to be *symmetric* if and only if $\gamma^{+i}(t) = \gamma^{-i}(t)$. Otherwise, the view of u_i is said to be *asymmetric*.

By convention, we state that the *configuration* of the system at instant t is $\gamma^{+0}(t)$. Any configuration from which there is a probability 0 that a robot moves is said to be *terminal*. Let $\gamma = \langle x_0x_1\dots x_{n-1}\rangle$ be a configuration. The configuration $\langle x_ix_{i+1}\dots x_{i+n-1}\rangle$ is obtained by rotating γ of $i \in [0\dots n-1]$. Two configurations γ and γ' are said to be *indistinguishable* if and only if γ' can be obtained by rotating γ or its mirror. Two configurations that are not indistinguishable are said to be *distinguishable*. We designate by *initial configurations* the configurations from which the system can start at instant 0.

During the Look phase of some cycle, it may happen that both edges incident to a node v currently occupied by the robot look identical in the snapshot, *i.e.*, v lies on a symmetric axis of the configuration. In this case, if the robot decides to move, it may traverse any of the two edges. We assume the worst case decision in such cases, *i.e.*, that the decision to traverse one of these two edges is taken by an adversary.

We call *computation* any infinite sequence of configurations $\gamma_0, \dots, \gamma_t, \gamma_{t+1}, \dots$ such that (1) γ_0 is a possible initial configuration and (2) for every instant $t \geq 0$, γ_{t+1} is obtained from γ_t after some robots (at least one) execute a cycle.

Any transition γ_t, γ_{t+1} is called a step of the computation. A computation c *terminates* if c contains a terminal configuration.

A *scheduler* is a predicate on computations, that is, a scheduler defines a set of *admissible* computations, such that every computation in this set satisfies the scheduler predicate. Here we assume a *distributed fair* scheduler. Distributed means that, at every instant, any non-empty subset of robots can be activated. Fair means that every robot is activated infinitively often during a computation. A particular case of distributed fair scheduler is the *sequential* fair scheduler: at every instant, one robot is activated and every robot is activated infinitively often during a computation. In the following, we call *sequential computation* any computation that satisfies the sequential fair scheduler predicate.

Problem to be solved. We consider the *exploration* problem, where k robots collectively explore a n-sized ring before stopping moving forever. More formally, a protocol $\mathcal{P}$ *deterministically* (resp. *probabilistically*) solves the exploration problem if and only if every computation c of $\mathcal{P}$ starting from a *towerless configuration* satisfies:

1. c terminates in *finite time* (resp. with *expected finite time*).
2. Every node is visited by at least one robot during c.

The previous definition implies that every initial configuration of the system in the problem we consider is *towerless*. Using probabilistic solutions, termination is not certain, however the overall probability of non-terminating computations is 0.

3 Negative Result

In this section, we show that the exploration problem is impossible to solve in our settings (*i.e.*, oblivious robots, anonymous ring, distributed scheduler, ...) if there is less than four robots, even in a probabilistic manner (Corollary 2). The proof is made in two steps:

– The first step is based on the fact that obliviousness constraints any exploration protocol to construct an implicit memory using the configurations. We show that if the scheduler behaves sequentially, then in any case except one, it is not possible to particularize enough configurations to memorize which nodes have been visited (Theorem 1 and Lemma 4).
– The second step consists in excluding the last case (Theorem 2).

If $n > k$, any terminal configuration should be distinguishable from any possible initial (towerless) configuration. Hence, follows:

Remark 1. If $n > k$, any terminal configuration of any exploration protocol contains at least one tower.

Lemmas 1 to 3 proven below are technical results that lead to Corollary 1. The latter exhibits the minimal size of a subset of particular configurations required to solve the exploration problem.

Definition 1 (MRP). *Let s be a sequence of configurations. The* minimal relevant prefix *of s, noted $\mathcal{MRP}(s)$, is the maximal subsequence of s where no two consecutive configurations are identical.*

Lemma 1. *Let $\mathcal{P}$ be any (probabilistic or deterministic) exploration protocol for k robots in a ring of $n > k$ nodes. For every sequential computation c of $\mathcal{P}$ that terminates, $\mathcal{MRP}(c)$ has at least $n - k + 1$ configurations containing a tower.*

Proof. Assume, by the contradiction, that there is a sequential computation c of $\mathcal{P}$ that terminates and such that $\mathcal{MRP}(c)$ has less than $n-k+1$ configurations containing a tower.

Take the last configuration α without tower which appear in computation c and all remaining configurations (all of them contains towers) that follow in c and form c'. As α could be an initial configuration and c is an admissible sequential computation that terminates, c' is also an admissible sequential computation of $\mathcal{P}$ that terminates. Notice that $\mathcal{MRP}(c')$ has at most $n - k + 1$ configurations. Since c' is sequential, going from configuration α to a configuration with towers, no new nodes are explored (the same happens when remaining at the same configuration with towers). Hence the total number of nodes explored upon the termination of c' is at most k (the ones that are initially visited) + $n - k - 1$ (the ones that are dynamically visited) = $n - 1$: c' terminates before all nodes are visited, a contradiction.

Lemma 2. *Let $\mathcal{P}$ be any (probabilistic or deterministic) exploration protocol for k robots in a ring of $n > k$ nodes. For every sequential computation c of $\mathcal{P}$ that terminates, $\mathcal{MRP}(c)$ has at least $n - k + 1$ configurations containing a tower of less than k robots.*

Proof. Assume, by the contradiction, that there is a sequential computation c of $\mathcal{P}$ that terminates and such that $\mathcal{MRP}(c)$ has less than $n-k+1$ configurations containing a tower of less than k robots.

Take the last configuration α without tower which appear in computation c and all remaining configurations (all of them contains towers) that follow in c and form c'. As α could be an initial configuration and c is an admissible sequential computation that terminates, c' is also an admissible sequential computation of $\mathcal{P}$ that terminates.

$\mathcal{MRP}(c')$ is constituted of a configuration with no tower followed by at least $n - k + 1$ configurations containing a tower by Lemma 1 and $n - k$ new nodes (remember that k nodes are already visited in the initial configuration) must be visited before c' reaches its terminal configuration.

Consider a step $\alpha\alpha'$ in c'.

- If $\alpha = \alpha'$, then no node is visited during the step.
- If $\alpha \neq \alpha'$, then there are three possible cases:

1. *α contains no towers.* In this case, α is the initial configuration and α' contains a tower. As only one robot moves in $\alpha\alpha'$ to create a tower (c' is sequential), no node is visited during this step.
2. *α contains a tower and α' contains a tower of k robots.* As c' is sequential and all robots are located at the same node in α', one robot moves to an already occupied node in $\alpha\alpha'$ and no node is visited during this step.
3. *α contains a tower and α' contains a tower of less than k robots.* In this case, at most one node is visited in $\alpha\alpha'$ because c' is sequential.

To sum up, only the steps from a configuration containing a tower to a configuration containing a tower of less than k robots allow to visit at most one node each time. Now, in $\mathcal{MRP}(c')$ there are less than $n-k+1$ configurations containing a tower of less than k robots and the first of these configurations appearing into c' is consecutive to a step starting from the initial configuration. Hence, less than $n-k$ nodes are dynamically visited during c' and, as exactly k nodes are visited in the initial configuration, less than n nodes are visited when c' terminates, a contradiction.

Lemma 3. *Let $\mathcal{P}$ be any (probabilistic or deterministic) exploration protocol for k robots in a ring of $n > k$ nodes. For every sequential computation c of $\mathcal{P}$ that terminates, $\mathcal{MRP}(c)$ has at least $n-k+1$ configurations containing a tower of less than k robots and any two of them are distinguishable.*

Proof. Consider any sequential computation c of $\mathcal{P}$ that terminates.

By Lemma 2, $\mathcal{MRP}(c)$ has x configurations containing a tower of less than k robots where $x \geq n-k+1$.

We first show that (**) *if c contains at least two different configurations having a tower of less than k robots that are indistinguishable, then there exists a sequential computation c' that terminates and such that $\mathcal{MRP}(c')$ has x' configurations containing a tower of less than k robots where $x' < x$.* Assume that there are two different indistinguishable configurations γ and γ' in c having a tower of less than k robots. Without loss of generality, assume that γ occurs at time t in c and γ' occurs at time $t' > t$ in c. Consider the two following cases:

1. **γ' can be obtained by applying a rotation of i to γ.** Let p be the prefix of c from instant 0 to instant t. Let s be the suffix of c starting at instant $t'+1$. Let s' be the sequence obtained by applying a rotation of $-i$ to the configurations of s. As the ring and the robots are anonymous, ps' is an admissible sequential computation that terminates. Moreover, by construction $\mathcal{MRP}(ps')$ has x' configurations containing a tower of less than k robots where $x' < x$. Hence (**) is verified in this case.
2. **γ' can be obtained by applying a rotation of i to the mirror of γ.** We can prove (**) in this case by slightly modifying the proof of the previous case: we have just to apply the rotation of $-i$ to the *mirrors* of the configurations of s.

By (**), if $\mathcal{MRP}(c)$ contains less than $n-k+1$ distinguishable configurations with a tower of less than k robots, it is possible to (recursively) construct an admissible computation c' of $\mathcal{P}$ that terminates such that $\mathcal{MRP}(c')$ has less than $n-k+1$ configurations containing a tower of less than k robots, a contradiction to Lemma 2. Hence, the lemma holds.

From Lemma 3, we can deduce the following corollary:

Corollary 1. *Considering any (probabilistic or deterministic) exploration protocol for k robots in a ring of $n > k$ nodes, there exists a subset $\mathcal{S}$ of at least $n-k+1$ configurations such that:*

1. *Any two different configurations in $\mathcal{S}$ are distinguishable, and*
2. *In every configuration in $\mathcal{S}$, there is a tower of less than k robots.*

Theorem 1. $\forall k, 0 \leq k < 3, \forall n > k$, *there is no exploration protocol (even probabilistic) of a n-size ring with k robots.*

Proof. First, for $k = 0$, the theorem is trivially verified. Consider then the case $k = 1$ and $k = 2$: with one robot it is impossible to construct a configuration with one tower; with two robots it is impossible to construct a configuration with one tower of less than k robots ($k = 2$). Hence, for $k = 1$ and $k = 2$, the theorem is a direct consequence of Corollary 1.

Lemma 4. $\forall n > 4$, *there is no exploration protocol (even probabilistic) of a n-size ring with three robots.*

Proof. With three robots, the size of the maximal set of distinguishable configurations containing a tower of less than three robots is $\lfloor n/2 \rfloor$. By Corollary 1, we have then the following inequality:

$$\lfloor n/2 \rfloor \geq n - k + 1$$

From this inequality, we can deduce that n must be less of equal than four and we are done.

From this point on, we know that, assuming $k < 4$, Corollary 1 prevents the existence of any exploration protocol in any case except one: $k = 3$ and $n = 4$ (Theorem 1 and Lemma 4). Actually, assuming that the scheduler is sequential is not sufficient to show the impossibility in this latter case: Indeed, there is an exploration protocol for $k = 3$ and $n = 4$ if we assume a sequential scheduler. This latter protocol can be found in the technical report ([14], `http://hal.inria.fr/inria-00360305/fr/`).

The theorem below is obtained by showing the impossibility for $k = 3$ and $n = 4$ using a (non-sequential) distributed scheduler.

Theorem 2. *There is no exploration protocol (even probabilistic) of a n-size ring with three robots for every $n > 3$.*

Proof Outline. Lemma 4 excludes the existence of any exploration protocol for three robots in a ring of $n > 4$ nodes. Hence, to show this theorem, we just have to show that there is no exploration protocol for three robots working in a ring of four nodes.

The remainder of the proof consists in a combinatorial study of all possible protocols for $k = 3$ robots and $n = 4$ nodes. In each case, we show that the protocol leads to one of the following contradiction:

- Either, the adversarial choices of the scheduler allow to construct an admissible computation that never terminates with probability 1.
- Or, for every possible terminal configuration (*i.e.*, any configuration containing a tower, see Remark 1), there is an admissible computation that reaches the terminal configuration without visiting all nodes.

□

From Theorems 1 and 2, we can deduce the following corollary:

Corollary 2. *$\forall k, 0 \leq k < 4, \forall n > k$, there is no exploration protocol (even probabilistic) of a n-size ring with k robots.*

4 Positive Result

In this section, we propose a probabilistic exploration protocol for $\boldsymbol{k = 4}$ robots in a ring of $\boldsymbol{n > 8}$ nodes. We first define some useful terms in Subsection 4.1. We then give the general principle of the protocol in Subsection 4.2. Finally, we detail and prove the protocol in Subsection 4.3.

4.1 Definitions

Below, we define some terms that characterize the configurations.

We call *segment* any maximal non-empty elementary path of occupied nodes. The *length of a segment* is the number of nodes that compose it. We call *x-segment* any segment of length x. In the segment $s = u_i, \ldots, u_k$ $(k \geq i)$ the nodes u_i and u_k are termed as the *extremities* of s. An *isolated node* is a node belonging to a 1-segment.

We call *hole* any maximal non-empty elementary path of free nodes. The *length of a hole* is the number of nodes that compose it. We call *x-hole* any hole of length x. In the hole $h = u_i, \ldots, u_k$ $(k \geq i)$ the nodes u_i and u_k are termed as the *extremities* of h. We call *neighbor* of an hole any node that does not belong to the hole but is neighbor of one of its extremities. In this case, we also say that the hole is a *neighboring hole* of the node. By extension, any robot that is located at a neighboring node of a hole is also referred to as a neighbor of the hole.

We call *arrow* a maximal elementary path $u_i, \ldots, u_k$ of length at least four such that (i) u_i and u_k are occupied by one robot, (ii) $\forall j \in [i+1 \ldots k-2]$, u_j

is free, and (*iii*) there is a tower of two robots in u_{k-1}. The node u_i is called the *arrow tail* and the node u_k is called the *arrow head*. The *size* of an arrow is the number of free nodes that compose it, *i.e.*, it is the length of the arrow path minus 3. Note that the minimal size of an arrow is 1 and the maximal size is $n-3$. Note also that when there is an arrow in a configuration, the arrow is unique. An arrow is said to be *primary* if its size is 1. An arrow is said to be *final* if its size is $n-3$.

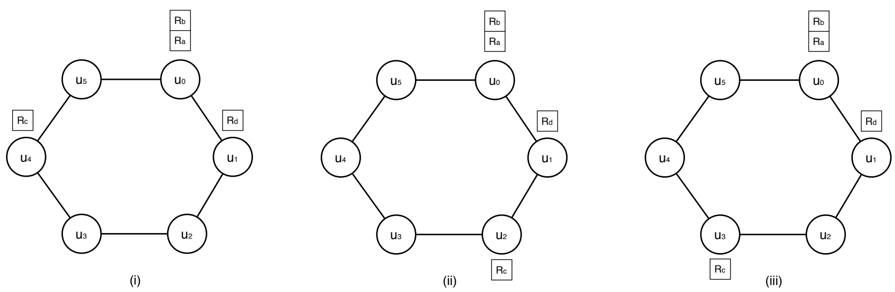

Fig. 1. Arrows

Figure 1 illustrates the notion of arrows: In Configuration (*i*) the arrow is formed by the path u_4, u_5, u_0, u_1; the arrow is primary; the node u_4 is the tail and the node u_1 is the head. In Configuration (*ii*), there is a final arrow (the path u_2, u_3, u_4, u_5, u_0, u_1). Finally, the size of the arrow in Configuration (*iii*) (the path u_3, u_4, u_5, u_0, u_1) is 2.

4.2 Overview of the solution

Our protocol (Algorithm 1) proceeds in three distinct phases:

- **Phase I:** Starting from a configuration without tower, the robots move along the ring in such a way that (*i*) they never form any tower and (2) form a unique segment (a 4-segment) in finite expected time.
- **Phase II:** Starting from a configuration with a unique segment, the four robots form a primary arrow in finite expected time. The 4-segment is maintained until the primary arrow is formed.
- **Phase III:** Starting from a configuration where the four robots form a primary arrow, the arrow tail deterministically moves toward the arrow head in such way that the length of the arrow never decreases. The protocol terminates when robots form a final arrow. At the termination, all nodes have been visited.

Note that the protocol we propose is probabilistic. As a matter of fact, as long as possible the robots move deterministically. However, we use randomization to break the symmetry in some cases: When the system is in a symmetric configuration, the scheduler may choose to synchronously activate some processes in such way that the system stays in a symmetric configuration. To break the symmetry despite the choice of the scheduler, we proceed as follows: The activated robots toss a coin (with a uniform probability) during their Compute phase. If they win the toss, they decide to move, otherwise they decide to stay idle. In this case, we say that the robots **try to move**. Conversely, when a process deterministically decides to move in its Compute phase, we simply say that the process **moves**.

Algorithm 1 The protocol.

```
if the four robots do not form a final arrow then
    if the configuration contains neither an arrow nor a 4-segment then
        Execute Procedure Phase I;
    else
        if the configuration contains a 4-segment then
            Execute Procedure Phase II;
        else    /* the configuration contains an arrow */
            Execute Procedure Phase III;
```

4.3 Detailed description of the solution

Phase I. Phase I is described in Algorithm 2. The aim of this phase is to eventually form a 4-segment without creating any tower during the process. Roughly speaking, in asymmetric configurations, robots moves determiniscally (Lines 4, 10, 27, 31). By contrast, in symmetric configurations, robots moves probabilistically using **Try to move** (Lines 16 and 22). Note that in all cases, we prevent the tower formation by applying the following constraint: a robot can move through a neighboring hole $\mathcal{H}$ only if its length is at least 2 or if the other neighboring robot cannot move through $\mathcal{H}$. Hence, we obtain the following lemma:

Lemma 5. *If the configuration at instant t contains neither a 4-segment nor a tower, then the configuration at instant $t+1$ contains no tower.*

The probabilistic convergence to a 4-segment is guaranteed by the fact that in a symmetric configuration, the moving robots move probabilistically. Thanks to that, the symmetries are eventually broken and the system reaches an asymmetric configuration from which the robots deterministically move until forming a 4-segment. Hence, we obtain the lemma below:

Lemma 6. *Starting from any initial (towerless) configuration, the system reaches in finite expected time a configuration containing a 4-segment.*

Phase II. Phase II is described in Algorithm 3: Starting from a configuration where there is a 4-segment on nodes u_i, u_{i+1}, u_{i+2}, u_{i+3}, the system eventually reaches a configuration where a primary arrow is formed on nodes u_i, u_{i+1}, u_{i+2}, u_{i+3}. To that goal, we proceed as follows: Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be the robots located

Algorithm 2 Procedure *Phase* I.

```
if the configuration contains a 3-segment then
    begin
    if I am the isolated robot then
        Move toward the 3-segment through the shortest hole;
    end
else
    if the configuration contains a unique 2-segment then        /* Two robots are isolated */
        begin
        if I am at the closest distance from the 2-segment then
            Move toward the 2-segment through the hole having me and an extremity of the 2-segment as neighbors;
        end
    else
        if the configuration contains (exactly) two 2-segments then
            begin
            if I am a neighbor of a longuest hole then
                Try to move toward the other 2-segment through my neighboring hole;
            end
        else      /* the four robots are isolated */
            begin
            Let l_max be the length of the longuest hole;
            if every robot is neighbor of a l_max-hole then
                Try to move through a neighboring l_max-hole;
            else
                if 3 robots are neighbors of a l_max-hole then
                    begin
                    if I am neighbor of only one l_max-hole then
                        Move toward the robot that is neighbor of no l_max-hole through my shortest neighboring hole;
                    end
                else      /* 2 robots are neighbors of the unique l_max-hole */
                    if I am neighbor of the unique l_max-hole then
                        Move through my shortest neighboring hole;
            end
```

at the nodes u_{i+1} and u_{i+2} of the 4-segment. $\mathcal{R}_1$ and $\mathcal{R}_2$ try to move to u_{i+2} and u_{i+1}, respectively. Eventually only one of these robots moves and we are done. Hence, we have the two lemmas below:

Lemma 7. *Let γ be a configuration containing a 4-segment $u_i, u_{i+1}, u_{i+2}, u_{i+3}$. If γ is the configuration at instant t, then the configuration at instant $t+1$ is either identical to γ or the configuration containing the primary arrow u_i, u_{i+1}, u_{i+2}, u_{i+3}.*

Lemma 8. *From a configuration containing a 4-segment, the system reaches a configuration containing a primary arrow in finite expected time.*

Algorithm 3 Procedure *Phase* II.

```
1: if I am not located at an extremity of the 4-segment then
2:     Try to move toward my neighboring node that is not an extremity of the 4-segment;
```

Phase III. Phase III is described in Algorithm 4. This phase is fully deterministic: This phase begins when there is a primary arrow. Let $\mathcal{H}$ be the hole between the tail and the head of arrow at the beginning of the phase. From the previous phase, we know that all nodes forming the primary arrow are already visited. So, the unvisited nodes can only be on $\mathcal{H}$ and the phase just consists in traversing $\mathcal{H}$. To that goal, the robot located at the arrow tail traverses $\mathcal{H}$. When it is done, the system is in a terminal configuration containing a final arrow and all nodes have been visited. Hence, we can conclude with the following theorem:

Theorem 3. *Algorithm 1 is a probabilistic exploration protocol for* $\mathbf{4}$ *robots in a ring of* $\boldsymbol{n} > \mathbf{8}$ *nodes.*

Algorithm 4 Procedure *Phase* III.

```
1: if I am the arrow tail then
2:     Move toward the arrow head through the hole having me and the arrow head as neighbor;
```

5 Conclusion

We considered a semi-synchronous model of computation. In this model, we provided evidence that for the exploration problem in uniform rings, randomization could shift complexity from $\Theta(\log n)$ to $\Theta(1)$. While applying randomization to other problem instances is an interesting topic for further research, we would like to point out immediate open questions raised by our work:

1. Though we were able to provide a general algorithm for any n (strictly) greater than eight, it seems that ad hoc solutions have to be designed when n is between five and eight (inclusive).
2. Our protocol is optimal with respect to the number of robots. However, the efficiency (in terms of exploring time) is only proved to be finite. Actually computing the convergence time from our proof argument is feasible, but it would be more interesting to study how the number of robots relates to the time complexity of exploration, as it seems natural that more robots will explore the ring faster.
3. It is worth investigating if our results can be extended to the (full) asynchronous model.

Acknowledgments. We are grateful to the anonymous referees for a very careful reading of the manuscript and a number of valuable remarks and suggestions that enabled us to improve the quality of the paper.

References

1. Asahiro, Y., Fujita, S., Suzuki, I., Yamashita, M.: A self-stabilizing marching algorithm for a group of oblivious robots. In: OPODIS. (2008) 125–144
2. Souissi, S., Défago, X., Yamashita, M.: Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. ACM Transactions on Adaptive and Autonomous Systems (TAAS) **4**(1) (2009)
3. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: Formation of geometric patterns. SIAM J. Comput. **28**(4) (1999) 1347–1363
4. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. Theor. Comput. Sci. **407**(1-3) (2008) 412–447
5. Dieudonné, Y., Labbani-Igbida, O., Petit, F.: Circle formation of weak mobile robots. ACM Transactions on Adaptive and Autonomous Systems (TAAS) **3**(4) (2008)
6. Dieudonné, Y., Petit, F.: Scatter of weak mobile robots. Parallel Processing Letters **19**(1) (2009) 175–184
7. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Transactions on Robotics and Automation (1999)
8. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. **337**(1-3) (2005) 147–168
9. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theor. Comput. Sci. **390**(1) (2008) 27–39
10. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: Gathering of asynchronous oblivious robots on a ring. In: 12th International Conference on Principles of Distributed Systems (OPODIS). (2008) 446–462
11. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Computing without communicating: Ring exploration by asynchronous oblivious robots. In: 14th International Colloquium on Structural Information and Communication Complexity (SIROCCO). (2007) 105–118

12. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Remembering without memory: Tree exploration by asynchronous oblivious robots. In: 15th International Colloquium on Structural Information and Communication Complexity (SIROCCO). (2008) 33–47
13. Efrima, A., Peleg, D.: Distributed algorithms for partitioning a swarm of autonomous mobile robots. Theor. Comput. Sci. **410**(14) (2009) 1355–1368
14. Devismes, S., Petit, F., Tixeuil, S.: Optimal probabilistic ring exploration by asynchronous oblivious robots. Technical Report inria-00360305, INRIA (February 2009)