Optimal Exploration of Small Rings

Stéphane Devismes
VERIMAG (UMR 5104), Univ. Joseph Fourier, France

ABSTRACT

In [4], the authors look at *probabilistic* bounds and solutions for the exploration of anonymous unoriented rings of any size by a cohort of robots. Considering identical, oblivious, and probabilistic robots, they show that at least four of them are necessary to solve the problem. Moreover, they give a randomized protocol for four robots working in any ring of size more than eight.

Here we close the question of optimal (*w.r.t.* the cohort size) ring exploration by probabilistic robots. Indeed, we propose a protocol for four robots working in any ring of size less or equal to eight. Composing this protocol with the one in [4], we obtain a protocol for any ring-size.

Keywords

Mobile robots, ring exploration, probabilistic algorithm.

1. INTRODUCTION

We consider autonomous robots that are endowed with visibility sensors (but that are otherwise unable to communicate) and motion actuators. However, we assume they have weak capacities: they are *anonymous*, *oblivious* (that is, they cannot remember the past), and have no compass whatsoever.

Those robots must collaborate to solve a collective task, here *exploration*. In our context, the exploration task requires every possible location to be visited by at least one robot, with the additional constraint that all robots stop moving after task completion.

The vast majority of literature on robots considers that they evolve 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.*, [2, 11].

Several works investigate restricting the capabilities of both visibility sensors and motion actuators of the robots, to circumvent the many impossibility results that appear in the general continuous model. In [1, 8], robots visibility sensors are supposed to be accurate within a constant range, and sense nothing beyond this range. In [8, 3], the space allowed for the motion actuator is reduced to a one-dimentional continuous one: a ring in [8], an infinite path in [3].

A recent trend is 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 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 robots. For each location, a robot is able to sense if the location is empty or if robots are positioned on it. 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 rather than on infinite ones. However, as noted in most related papers [10, 9, 6, 7], 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 [10, 9] and exploration [6, 7]. For gathering, both breaking symmetry [10] 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, [7] 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, [6] 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 *deterministic*, *i.e.*, if a robot is presented twice the same situation, its behavior is the same in both cases.

In [4], the authors initiate research on *probabilistic* bounds and solutions for the exploration of anonymous unoriented rings of any size. By contrast with [6] while similar settings, they show that *four* identical probabilistic robots are necessary to solve the same problem, also removing the coprime constraint between the number of robots and the size of the ring. In the same paper, they present a randomized protocol for four robots in any ring of size more than eight.

In this paper, after formally defining the model we use (Section 2), we close the question of optimal (*w.r.t.* the cohort size) ring exploration by probabilistic robots. Indeed, we propose (in Section 3) a protocol for four robots working in any ring of size less or equal to eight. Composing this protocol with the one of [4], we obtain a protocol for any ring-size. In the last section (Section 4), we gives some concluding remarks. Note that, for space consideration, several technical proofs have been omitted.

WRAS'2010 July 29, 2010, Zurich, Switzerland.
ACM 978-1-59593-989-0/08/08.

2. MODEL

Distributed System.

Autonomous mobile entities called *robots* evolve in a *graph*. We assume the graph to be a *ring* of n nodes, $u_0, \ldots, u_{n-1}$, *i.e.*, u_i is connected to both u_{i-1} and u_{i+1} (computations over indices are 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 operated by each robots. 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, it then (probabilistically or deterministically) decides to move or stay idle (Compute phase). When a robot decides a move, it moves to its destination during the Move phase.

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

Computations.

We consider a *semi-synchronous* model where 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 robots are on some nodes of the ring but not on edges. That is, during a Look phase, a robot cannot see another 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*. Let $d_i(t)$ be the state of a node u_i at instant t defined as follows: given a time instant t, $d_i(t)$ is equal to either $\circ$, $\bot$, or $\top$ according to u_i contains no, one or several robots, respectively. If $d_i(t) = \circ$, then we say that u_i is *free* at instant t. Otherwise ($d_i(t) \in \{\bot, \top\}$), u_i is said *occupied* at instant t. If $d_i(t) = \top$, then we say that u_i is *overcrowded* at instant t.

For an arbitrary orientation of the ring and a node u_i, $\gamma^{+i}(t)$ (respectively, $\gamma^{-i}(t)$) denotes the sequence $\langle d_i(t)d_{i+1}(t) \ldots d_{i+n-1}(t)\rangle$ (resp., $\langle d_i(t)d_{i-1}(t) \ldots d_{i-(n-1)}(t)\rangle$). 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. The view of u_i is said to be *symmetric* if and only if $\gamma^{+i}(t) = \gamma^{-i}(t)$, *asymmetric*, otherwise.

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 $\gamma = \langle x_0 x_1 \ldots x_{n-1}\rangle$ be a configuration. The configuration $\langle x_i x_{i+1} \ldots x_{i+n-1}\rangle$ is obtained by rotating γ of $i \in [0 \ldots n-1]$. 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, 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 $\gamma_0, \ldots, \gamma_t, \gamma_{t+1}, \ldots$ 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*. A computation c *terminates* if c contains a terminal configuration.

A *scheduler* is a predicate on computations which defines the set of *admissible* computations. 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.

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 configuration with no overcrowded node satisfies: (1) c terminates in *finite time* (resp. with *expected finite time*); (2) every node is visited by at least one robot during c.

Note that the previous definition implies that every initial configuration of the system in the problem we consider contains no overcrowded node. Note also that using probabilistic solutions, termination is not certain, however the overall probability of non-terminating computations is 0.

3. EXPLORATION PROTOCOL

In this section, we propose a probabilistic protocol for $\boldsymbol{k = 4}$ robots in a ring of $\boldsymbol{n}$ nodes with $\boldsymbol{4 < n \leq 8}$.

3.1 Definitions

We now give definitions to 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

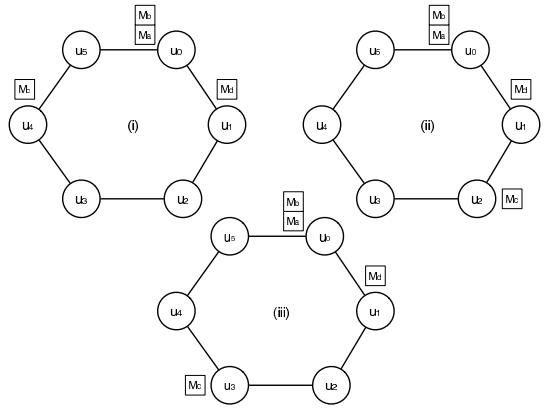

Figure 1: Arrows.

one robot, (ii) $\forall j \in [i+1 \ldots k-2]$, u_j is free, and (iii) u_{k-1} is *overcrowded*, the latter meaning occupied by at least two robots. 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 $n-3$.

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.

3.2 Overview

Our protocol is presented in Algorithm 1. Except for two special cases, it works in three main steps:

1. From an initial (with no overcrowded node) configuration, the robots move along the ring in such a way that (i) they never form an arrow and (2) they form a unique (4-)segment in finite expected time. (Lines 2-6;20-32)
2. 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. (Lines 8-12)
3. 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. (Lines 14-18)

Note that the protocol is probabilistic. As a matter of fact, as long as possible the robots move deterministically. Randomization is used to break the symmetry in some cases: When the system is in a symmetric configuration, the scheduler may choose to synchronously activate some robots in such a way that the system stays in a symmetric configuration. To break the symmetry despite the choice of the scheduler, the robots proceed as follows: The activated robots decide either to move or to stay idle (with a uniform probability) during their Compute phase. We say that the robots **try to move**. Conversely, when a robot deterministically decides to move in its Compute phase, we simply say that the robot **moves**.

3.3 Detailed Description

We first give some results holding for all rings of size 5 to 8 (Subsection 3.4) and then study the behavior of the protocol for each specific ring-size (Subsection 3.5 to 3.8). Finally, we conclude (Subsection 3.9).

3.4 Some Results

Let us first consider the case where there is no symmetry in the initial configuration: Assume a configuration where there is either (1) one 3-segment or (2) a unique 2-segment. Lines 2 to 6 in Algorithm 1 manage these two cases. In the first case, there is one isolated robot and it deterministically moves through its smallest neighboring hole until a 4-segment is formed.[1] In the second case, there are two isolated robots: the isolated robots (deterministically) moves through their neighboring hole having an extremity of the 2-segment as neighbor. Hence, eventually a 4-segment is formed, leading to the following lemma:

LEMMA 1. *Starting from any initial (with no overcrowded node) configuration containing either a 3-segment or a unique 2-segment, the system reaches in finite time a configuration containing a 4-segment without creating any overcrowded node during the process.*

Consider now a configuration (initial or not) containing a 4-segment on nodes $u_i, u_{i+1}, u_{i+2}, u_{i+3}$. In this case, Lines 8-12 in Algorithm 1 are executed. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be the robots located 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 a primary arrow is formed on nodes $u_i, u_{i+1}, u_{i+2}, u_{i+3}$, and we obtain the two lemmas below:

LEMMA 2. *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 3. *From a configuration containing a 4-segment, the system reaches a configuration containing a primary arrow in finite expected time.*

Once the system reaches a configuration containing a primary arrow, robots executes Lines 14-18 in Algorithm 1. From such a configuration, the protocol is fully deterministic: Let $\mathcal{H}$ be the hole between the tail and the head of the primary arrow. We know that all nodes forming the primary arrow are already visited. So, the unvisited nodes can only be on $\mathcal{H}$ and the process 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 lemma:

LEMMA 4. *From any configuration containing a 4-segment, the system reaches a terminal configuration containing a final arrow in finite expected time and when it is done, all nodes have been visited.*

[1]Note that the first time a robot moves, its two neighboring holes may have the same length and the adversary decides which edge to traverse.

Algorithm 1 The protocol.

```
1: if the four robots do not form a final arrow and
   the configuration is distinguishable from Configurations (b) and (d) in Figure 2 then
2:     if the largest segment has a size strictly less than four and is unique then
3:         begin
4:         if I am an isolated robot then
5:             Move toward the unique largest segment by the smallest hole having me and an extremity of the segment as neighbors;
6:         end
7:     else
8:         if the configuration contains a 4-segment then
9:             begin
10:            if I am not located at 4-segment extremity then
11:                Try to move toward my neighboring node that is not an extremity of the 4-segment;
12:            end
13:        else
14:            if the configuration contains an arrow then
15:                begin
16:                if I am the arrow tail then
17:                    Move toward the arrow head through the hole having me and the arrow head as neighbor;
18:                end
19:            else
20:                if the ring-size is 6 then
21:                    See Figure 2, Configuration (a);
22:                else
23:                    if the ring-size is 7 then /* there are two 2-segments */
24:                        begin
25:                        if I am neighbor of the largest hole then
26:                            Move through my neighboring hole;
27:                        end
28:                    else /* the ring-size is 8 */
29:                        if there are two 2-segments then
30:                            See Figure 4, Configurations (a) and (e);
31:                        else /* there are four isolated robots */
32:                            See Figure 5, Configuration (a);
```

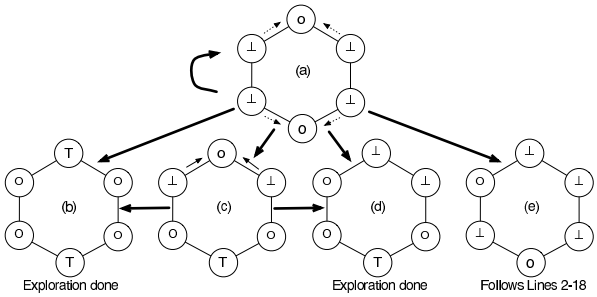

Figure 2: Symmetry breaking in a 6-size ring.

3.5 Size 5

Any initial configuration of a ring of size 5 contains a 4-segment. So, by Lemma 4, we can conclude:

THEOREM 1. *Algorithm 1 is a probabilistic exploration protocol for* **4** *robots in a ring of* **5** *nodes.*

Any initial configuration of a ring of size 6,7, or 8 matches one of these cases: (1) the configuration contains a 4-segment; (2) the configuration contains a 3-segment and one isolated node; (3) The configuration contains a 2-segment and two isolated node; (4) The configuration contains two 2-segments.

In the three first cases, Lines 2-18 in Algorithm 1 are executed and the correctness is obtained by Lemmas 1 and 4. So, in the following we only consider the last case.

3.6 Size 6

Consider a configuration containing two 2-segments in a ring of size 6. This configuration is indistinguishable with Configuration (a) in Figure 2. (In the figure, the symbols $\circ$, $\perp$, or $\top$ into a node means that the node contains no, one or several robots, respectively.) In Configuration (a), any robot tries to move toward its neighboring hole (dashed arrow). So, either the system stayes in the same configuration or the system reaches Configuration (b), (c), (d), or (e). However, after a finite expected number of attempts we have the guarantee that the system leaves Configuration (a) to Configuration (b), (c), (d), or (e). In Configuration (e) we retrieve a previous case, the robots executes Lines 2-18 in Algorithm 1. In case (b) and (d), we have the guarantee that all nodes are visited and as configuration (b) and (d) cannot be obtained anywhere else, there is no ambiguity and the process can stop. In Configuration (c), the two isolated nodes move as shown by the bold arrow and the system reaches either Configuration (b) or Configuration (d). Once again, we have the guarantee that all nodes are visited and as configuration (b) and (d) cannot be obtained anywhere else, there is no ambiguity and the process can stop. So, we can conclude with the following theorem:

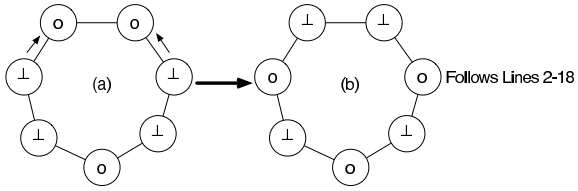

Figure 3: Symmetry breaking in a 7-size ring.

THEOREM 2. *Algorithm 1 is a probabilistic exploration protocol for* **4** *robots in a ring of* **6** *nodes.*

3.7 Size 7

Any configuration that contains two 2-segments in a ring of

4.(a)	$\mapsto$	4.(d)				
4.(b) = 5.(a)	$\mapsto$	5.(c)				
4.(c)						
4.(d)						
4.(e)	$\mapsto$	4.(g)				
4.(f) = 5.(a)	$\mapsto$	5.(c)				
4.(g)						
5.(a)	$\mapsto$	5.(c)				
5.(b)	$\mapsto$	5.(c)				
5.(c)						
5.(d)	$\mapsto$	5.(c)				
5.(e)	$\mapsto$	5.(f)	$\mapsto$	5.(i) = 4.(a)	$\mapsto$	4.(d)
5.(f)	$\mapsto$	5.(i) = 4.(a)	$\mapsto$	4.(d)		
5.(g)	$\mapsto$	5.(p) = 4.(e)	$\mapsto$	4.(g)		
5.(h)	$\mapsto$	5.(f)	$\mapsto$	5.(i) = 4.(a)	$\mapsto$	4.(d)
5.(i) = 4.(a)	$\mapsto$	4.(d)				
5.(j)	$\mapsto$	5.(g)	$\mapsto$	5.(p) = 4.(e)	$\mapsto$	4.(g)
5.(k)	$\mapsto$	5.(l)				
5.(l)						
5.(m)	$\mapsto$	5.(k)	$\mapsto$	5.(l)		
5.(n)						
5.(o)						
5.(p) = 4.(e)	$\mapsto$	4.(g)				

Table 1: Probabilistic Convergence.

size 7 is indistinguishable with Configuration (a) in Figure 3. In this case, robots executes Lines 23-27 in Algorithm 1 and the system reaches a configuration indistinguishable with configuration (b) in Figure 3, *i.e.*, the configuration contains one 2-segment and two isolated nodes. From that point, robots execute Lines 2-18 in Algorithm 1 and by Lemmas 1 and 4, we have:

THEOREM 3. *Algorithm 1 is a probabilistic exploration protocol for* **4** *robots in a ring of* **7** *nodes.*

3.8 Size 8

Figures 4 and 5 describe the behavior of our protocol starting from a configuration that contains two 2-segments. These figures can be seen as an automaton:

- Configurations are the states of the automaton.
- Bold arrows between configurations represent possible transitions. (More precisely, the transition $\gamma \mapsto \gamma'$ means that any configuration indistinguishable with γ' can be reached from any configuration indistinguishable with γ.)
- Configurations (a), (e) in Figure 4, and Configuration (a) in Figure 5 are initial states of the automaton. Any configuration that contains two 2-segments in a ring of size 8 is indistinguishable with one of those configurations.
- Below any configuration having no outgoing transition, we explain what robots have to do.

In any configuration, we show how robots must behave using arrows: dashed arrows represents **Try to move** actions. When there are two possible directions for a robot, this means that if the robot is activated, the edge it will traverse is chosen by the adversary.

First, we can observe that there is no ambiguity between the process described in Figures 4 and 5 and the rest of the protocol.

We can then remark that starting from Configurations (a), (e) in Figure 4, or Configuration (a), the system leaves configurations of Figures 4 and 5 only when the system reaches a configuration containing either a 3-segment and one isolated node or a 2-segment and two isolated nodes: Configurations (c), (d), and (g) in Figure 4 as well as Configurations (c), (l), (n), (o) in Figure 5. Let $\mathcal{C}_{good}$ the set of all these configurations.

From any configurations in $\mathcal{C}_{good}$, robots execute Lines 2-18 in Algorithm 1 and by Lemmas 1 and 4, the exploration is achieved a finite expected time.

Consider now a configuration γ in Figures 4 or 5 that is not in $\mathcal{C}_{good}$. In any configuration γ, there is at least one robot that executes a **Try to move** if activated and every robot either stays idle or executes **Try to move** if activated. So, in any configuration, there is a strictly positive probability that only one robot moves despite the choice of the scheduler. We can then remark (see Table 1) that from γ, there is path that leads to a configuration of $\mathcal{C}_{good}$ and any transition in this path has a strictly positive probability to occur: these transitions correspond to steps where exactly one robot moves. So, as the set of configurations in Figures 4 or 5 is finite, the expected time to reach a configuration of $\mathcal{C}_{good}$ is finite and we can conclude:

THEOREM 4. *Algorithm 1 is a probabilistic exploration protocol for* **4** *robots in a ring of* **8** *nodes.*

3.9 General Result

By Theorems 1 to 4, follows:

THEOREM 5. *Algorithm 1 is a probabilistic exploration protocol for* **4** *robots in a ring of* $\boldsymbol{n}$ *nodes with* $\mathbf{4 < n \leq 8}$.

4. CONCLUSION

The problem of exploring a discrete environment is one of the main problems in the field of mobile computing. One of the main challenges is to overcome the weakness of the model by itself, namely (*i*) the fact that the robots cannot remember past actions or positions and (*ii*) the lack of means to particularize robots or vertices, nor a mean of orientation. For instance, the fact that robots need to stop after exploring all locations requires robots to find an implicit way 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. This paper closes the question of the optimal number of probabilistic robots to explore a ring. Indeed, our protocol completes the work in [4], as it allows four robots to explore any ring of size less or equal to eight. Overall, it is possible to explore any ring using only four probabilistic robots.

5. REFERENCES

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 1999.

[2] Y. Asahiro, S. Fujita, I. Suzuki, and M. Yamashita. A self-stabilizing marching algorithm for a group of oblivious robots. In *OPODIS*, pages 125–144, 2008.

[3] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Byzantine-resilient convergence in oblivious robot networks. In *International Conference on Distributed Systems and Networks (ICDCN 2009)*, January 2009.

[4] S. Devismes, F. Petit, and S. Tixeuil. Optimal probabilistic ring exploration by asynchronous oblivious robots. In *Proceedings of Sirocco 2009*, Lecture Notes in Computer Science, Piran, Slovenia, May 2009. Springer-Verlag Berlin Heidelberg.

[5] Y. Dieudonné, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. *TAAS*, 3(4), 2008.

[6] P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. In *OPODIS*, pages 105–118, 2007.

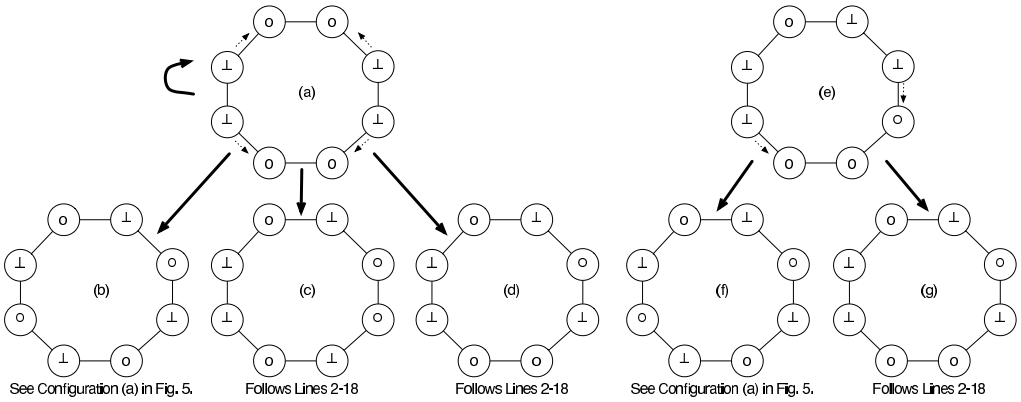

Figure 4: 2-segment symmetries in a 8-size ring.

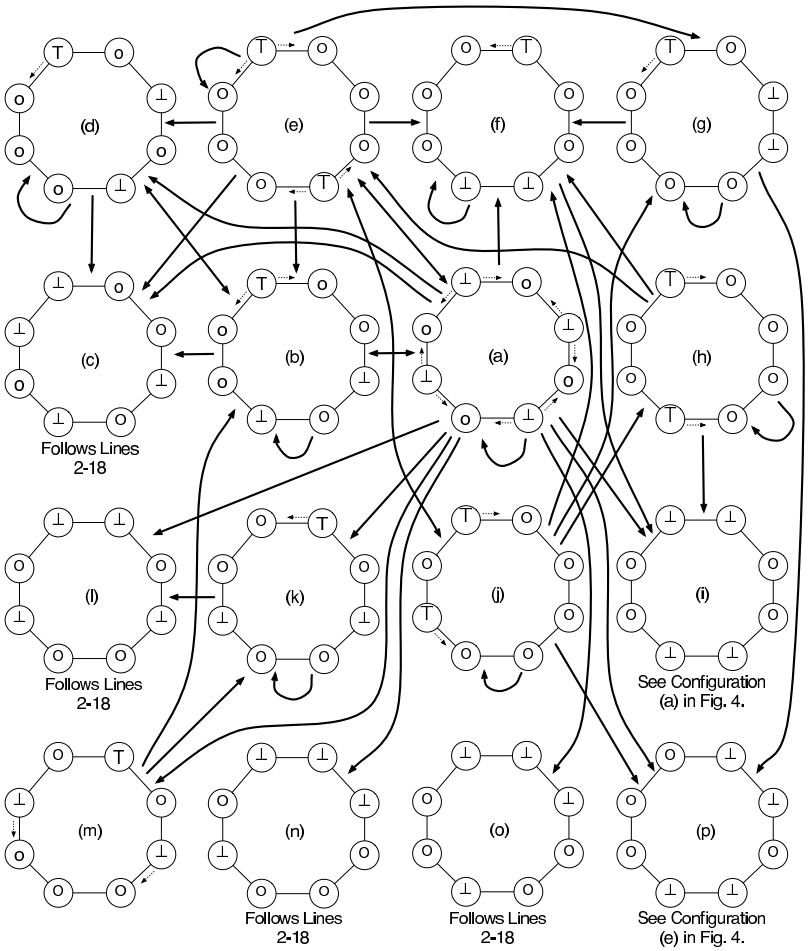

Figure 5: Isolated nodes symmetry in a 8-size ring.

[7] P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots. In A. A. Shvartsman and P. Felber, editors, *SIROCCO*, volume 5058 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.

[8] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theor. Comput. Sci.*, 337(1-3):147–168, 2005.

[9] R. Klasing, A. Kosowski, and A. Navarra. Taking advantage of symmetries: Gathering of asynchronous oblivious robots on a ring. In *OPODIS*, pages 446–462, 2008.

[10] R. Klasing, E. Markou, and A. Pelc. Gathering asynchronous oblivious mobile robots in a ring. *Theor. Comput. Sci.*, 390(1):27–39, 2008.

[11] S. Souissi, X. Défago, and M. Yamashita. Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. *TAAS*, 4(1), 2009.