# Perpetual Torus Exploration by Myopic Luminous Robots

Omar Darwich[1], Ahmet-Sefa Ulucan[1], Quentin Bramas[1], Anissa Lamani[1], Anaïs Durand[2], Pascal Lafourcade[2]

[1] University of Strasbourg, CNRS UMR 7357, ICUBE, France
[2] University Clermont Auvergne, CNRS UMR 6158, LIMOS, France

**Abstract.** We study perpetual torus exploration for swarm of autonomous, anonymous, uniform, luminous robots with a common chirality. We consider robots with only few capabilities. They have a finite limited vision (myopic), they can only see robots at distance one or two. We show that the problem is impossible with only two luminous robots and also with three oblivious robots (without light). We design an optimal algorithm for three luminous robots using two colors and with visibility one. We also propose an optimal algorithm with visibility two with four oblivious robots.

## 1 Introduction

Swarm robotics has drawn a lot of attention the past decade. Inspired by natural systems, a lot of investigations focused on how to reproduce autonomous behaviors observed in nature within artificial systems. Given a collection of autonomous mobile entities called robots, the main focus is to determine the minimum hypothesis in order for the robots to solve a given task. Robots can evolve either on a continuous 2D plane on which they can freely move or on a discrete universe, generally represented by a graph, where nodes indicate possible locations of the robots and the edges the possibility for the robots to move from one node to another.

In this paper, we assume that the mobile robots are autonomous (*i.e.* there is no central authority to coordinate their move), anonymous (*i.e.* they have no identity), uniform (*i.e.* they all execute the same algorithm) and luminous (*i.e.* they are endowed with lights of different colors). Moreover, they cannot communicate directly but are endowed by visibility sensors allowing them to sense their environment within a certain distance called visibility range. We assume myopic robots that can only sense at small distances. Robots operate in the well-known LCM model. That is, they operate in cycles which comprise three phases: Look, Compute, and Move. During the first phase (Look), robots take a snapshot of their environment using their visibility sensors. In the second phase (Compute), based on the taken snapshot, they first decide whether to move or remain idle and then whether they change their color. If they decide to move, they compute a neighboring destination. Similarly, they compute a new color if they decide to change it. Finally, in the last phase (Move), they move to the computed destination (if any) and they change their color (if they decided to). We consider the fully synchronous model (FSYNC) in which all robots execute the LCM cycle synchronously and atomically.

In the following, we investigate the case in which the robots have to solve the perpetual exploration problem. In this problem, robots evolve in a discrete universe and have to ensure that each location (node) is visited by at least one robot infinitely often. We are interested in torus shaped networks and focus on optimal exclusive solutions with respect to both the visibility range and the number of robots. Exclusiveness add an additional constraint on robots behavior as they can neither occupy the same node simultaneously or traverse the same edge at the same time.

## 2 Related work

The exploration problem is considered as one of the benchmarking tasks when it comes to robots evolving on graphs. Various topologies have been considered: lines [**?**], rings [**?**,**?**,**?**,**?**,**?**], tori [**?**], grids [**?**,**?**,**?**,**?**], cuboids [**?**], and trees [**?**]. Two variants of the problem has been investigated: (i) the perpetual exploration problem [**?**,**?**,**?**,**?**], considered in this paper, which requires the robots to visit each node of the graph infinitely often and (ii) the terminating exploration problem [**?**,**?**,**?**,**?**,**?**,**?**,**?**] which requires the robots to visit each node of the graph at least once and then stop moving.

Most of the investigations consider robots with unlimited visibility range allowing them to observe every node of the system [**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**]. Robots are in this case oblivious (*i.e.* they cannot remember past actions) and have to solve the terminating exploration problem. Myopic robots have also been considered in both variants of the problem [**?**,**?**,**?**,**?**,**?**]. When it comes to the perpetual exploration problem, an additional assumption has an impact on the feasibility of the task and the optimality of the proposed solutions. This assumption endow the robots with a common chirality. In fact, chirality is usually assumed when robots evolve in the continuous 2D Euclidean plan but some investigations have also considered it recently in the discrete universe. On finite grids, it has been shown that two (resp. three) synchronous robots with three colors (resp. one color) are sufficient to solve the problem when robots have visibility one and share a common chirality [**?**]. The case in which robots have no common chirality was investigated in [**?**]. It was proven that the problem is not solvable with only two robots having any finite number of colors and a finite visibility range. An optimal solution is also presented using only three robots having visibility range one, using only three colors. The case in which robots are oblivious and visibility range 2 was solved using five robots. In the case of infinite grids, assuming robots with visibility range one and few colors (O(1)), five (resp. six) synchronous robots are necessary and sufficient to solve the problem with (resp. without) the common chirality assumption [**?**,**?**]. Finally, in the case of cuboids, it has been shown in [**?**] that three synchronous robots with a common chirality endowed with five colors are necessary and sufficient to solve the perpetual exploration problem.

*Contribution:* We first present two impossibility results: we start by showing that the perpetual torus exploration problem is not solvable with only two robots if the number of colors is finite and their visibility range is limited. We then show that three oblivious robots are not sufficient to solve the PTE problem. Next, we propose two optimal solutions $\mathcal{A}_3^2$ and $\mathcal{A}_4^1$ with respect to both the number of robots and the number of colors for the case of visibility one and two respectively. Table **??** summarizes our contribution:

| Visibility | # Robots | # Colors | Algorithm |
|---|---|---|---|
| finite | 2 | finite | Impossible (Thm. **??**) |
| finite | 3 | 1 | Impossible (Thm. **??**) |
| 1 | 3 | 2 | $\mathcal{A}_3^2$ |
| 2 | 4 | 1 | $\mathcal{A}_4^1$ |

Table 1: Summary of our results.

## 3 Model

We consider a set $\mathcal{R}$ of $n > 0$ robots located on a *torus*. A graph $G = (V, E)$ is a $(l, L)$-torus (or torus for short) if $|V| = l \times L$ and for any $v_{(i,j)} \in V$; $i \in [0, l-1]$, $j \in [0, L-1]$:

- $\{v_{(i,j)}, v_{((i+1) \bmod l, j)}\} \in E$, and
- $\{v_{(i,j)}, v_{(i,(j+1) \bmod L)}\} \in E$.

The order on the nodes of $G$ forms a coordinate system. For example node $v_{(i,j)}$ is at coordinate $(i, j)$, or, the node is at column $i$ and row $j$. For simplicity we note node $(i, j)$ instead of $v_{(i,j)}$. This order/coordinate is used for the analysis only, *i.e.*, robots cannot access it.

We assume a discrete time where at each *round*, the robots synchronously perform a *Look-Compute-Move* cycle. In the *Look* phase, a robot gets a snapshot of the subgraph induced by the nodes within distance $\Phi \in \mathbb{N}^*$ from its position. $\Phi$ is called the *visibility range* of the robots. The snapshot is not oriented in any way as the robots do not agree on a common North. However, it is implicitly ego-centered since the robot that performs a Look phase is located at the center of the subgraph in the obtained snapshot. Robots agree on a common chirality. Then, each robot *computes* a destination (either Up, Left, Down, Right or Idle) based only on the snapshot it received. Finally, it *moves* towards its computed destination. We also assume that robots are *opaque*, *i.e.*, they obstruct the visibility in such way that if three robots are aligned, the two extremities cannot see each other. We forbid any two robots to occupy the same node simultaneously. A node is *occupied* when a robot is located at this node, otherwise it is *empty*.

Robots may have *lights* with different colors that can be seen by robots within distance $\Phi$ from them. We denote by $Cl$ the set of all possible colors. For simplicity, we assume that all tore has dimensions $l \times L$ where $l, L \geq n\Phi + 1$.

The *state* of a node is either the color of the light of the robot located at this node, if it is occupied, or $\perp$ otherwise. In the Look phase, the snapshot includes the state of the nodes (within distance $\Phi$, including its current node). During the compute phase, a robot may decide to change the color of its light.

In all our algorithms, we also prevent any two robots from traversing the same edge simultaneously. Since we already forbid them to occupy the same position simultaneously, this means that we additionally prevent robots from swapping their position. Algorithms verifying this property are said to be *exclusive*. However, to be as general as possible, we do not make this additional assumption in our impossibility results.

In the following, we borrow some of the definitions already presented in [**?**].

**Configurations** A *configuration* $C$ in a torus $G(V, E)$ is a set of pairs $(p, c)$, where $p \in V$ is an occupied node and $c \in Cl$ is the color of the robot located at $p$. A node $p$ is empty if and only if $\forall c,\ (p, c) \notin C$. We sometimes just write the set of occupied nodes when the colors are clear from the context.

**Views** We denote by $G_r$ the *globally oriented view* centered at the robot $r$, *i.e.*, the subset of the configuration containing the states of the nodes at distance at most $\Phi$ from $r$, translated so that the coordinates of $r$ is $(0, 0)$. We use this globally oriented view in our analysis to describe the movements of the robots: when we say "the robot moves Up", it is according to the globally oriented view. However, since robots do not agree on a common North, they have no access to the globally oriented view. When a robot looks at its surroundings, it instead obtains a snapshot. To model this, we assume that the *local view* acquired by a robot $r$ in the Look phase is the result of an arbitrary *indistinguishable transformation* on $G_r$. The set $\mathcal{IT}$ of indistinguishable transformations contains the rotations of angle 0 (to have the identity), $\pi/2$, $\pi$ and $3\pi/2$, centered at $r$. Moreover, since robots may obstruct visibility, the function that removes the state of a node $u$ if there is another robot between $u$ and $r$ is *systematically* applied to obtain the local view. Finally, we assume that robots are *self-inconsistent*, meaning that different transformations may be applied at different rounds.

It is important to note that when a robot $r$ computes a destination $d$, it is relative to its local view $f(G_r)$, which is the globally oriented view transformed by some $f \in \mathcal{IT}$. So, the actual movement of the robot in the *globally oriented view* is $f^{-1}(d)$. For example, if $d = Up$ but the robot sees the torus upside-down ($f$ is the $\pi$-rotation), then the robot moves $Down = f^{-1}(Up)$. In a configuration $C$, $V_C(i, j)$ denotes the globally oriented view of a robot located at $(i, j)$.

**Algorithm** An algorithm $\mathcal{A}$ is a tuple $(Cl, Init, T)$ where $Cl$ is the set of possible colors, $Init$ is a mapping from any considered torus to a non-empty set of initial configurations in that torus, and $T$ is the transition function $Views \to \{Idle, Up, Left, Down, Right\} \times Cl$, where $Views$ is the set of local views. When the robots are in Configuration $C$, a configuration $C'$ obtained after one round satisfies: for all $((i, j), c) \in C'$, there exists a robot in $C$ with color $c' \in Cl$ and a transformation $f \in \mathcal{IT}$ such that one of the following conditions holds:

- $((i, j), c') \in C$ and $f^{-1}(T(f(V_C(i, j)))) = (Idle, c)$,
- $(((i-1) \bmod l, j), c') \in C$ and $f^{-1}(T(f(V_C((i-1) \bmod l, j)))) = (Right, c)$,
- $(((i+1) \bmod l, j), c') \in C$ and $f^{-1}(T(f(V_C((i+1) \bmod l, j)))) = (Left, c)$,
- $((i, (j-1) \bmod L), c') \in C$ and $f^{-1}(T(f(V_C(i, (j-1) \bmod L)))) = (Up, c)$, or
- $((i, (j+1) \bmod L), c') \in C$ and $f^{-1}(T(f(V_C(i, (j+1) \bmod L)))) = (Down, c)$.

We denote by $C \mapsto C'$ the fact that $C'$ can be reached in one round from $C$ (*n.b.*, $\mapsto$ is then a binary relation over configurations). An execution of Algorithm $\mathcal{A}$ in a torus $G$ is then a sequence $(C_i)_{i\in\mathbb{N}}$ of configurations such that $C_0 \in Init(G)$ and $\forall i \geq 0$, $C_i \mapsto C_{i+1}$.

**Definition 1 (Perpetual Torus Exploration).** *An algorithm $\mathcal{A}$ solves the* Perpetual Torus Exploration *(PTE) problem if in any execution $(C_i)_{i\in\mathbb{N}}$ of $\mathcal{A}$ and for any node*

$(i, j) \in V$ *of the torus and any time* $t$*, there exists* $t' > t$ *such that* $(i, j)$ *is occupied in* $C_{t'}$*.*

*Notations.* $\vec{t}_{(i,j)}(C)$ denotes the translation of the configuration $C$ of vector $(i, j)$.

## 4 Impossibility Results

**Lemma 1.** *Let* $\mathcal{A}$ *be an algorithm using a set* $\mathcal{R}$ *of* $n > 0$ *robots. If* $\mathcal{A}$ *solves the exploration problem (perpetual or with termination) for any torus then, there exists a tori such that for any execution* $(C_i)_{i \in \mathbb{N}}$ *of* $\mathcal{A}$ *on this torus, there is a configuration* $C_i$ *such that the distance between the two farthest robots is at least* $2\Phi + 3$*.*

*Proof.* We proceed by contradiction. Assume, there is an algorithm $\mathcal{A}$ that solves the PTE problem and let $0 < B$ be the farthest any of the robots will be from each other, in any torus. Let $(C_i)_{i \in \mathbb{N}}$ be the execution of $\mathcal{A}$ on a very large torus $l, L \gg B$. When all robots are at distance at most $B$, then the occupied positions are included in a square sub-grid of size $B \times B$. Since the number of possible configurations included in a sub-grid of size $B \times B$ is finite, there must be two indices $t_1$ and $t_2$, when the positions and colors of the robots in the corresponding sub-grids are the same, formally, such that $C_{t_2} = \vec{t}_{(i,j)}(C_{t_1})$ and $t_1 < t_2$ for a given translation $\vec{t}_{(i,j)}$. By making the adversary choose the same rotation, the movements done by the robots in configurations $C_{t_1}$ and $C_{t_2}$ are the same as each robot has the same globally oriented view in both configurations, only their positions on the torus change. Thus $C_{t_2+1} = \vec{t}_{(i,j)}(C_{t_1+1})$ and so on so forth, so that $\forall x$, $C_{t_2+x} = \vec{t}_{(i,j)}(C_{t_1+x})$. We obtain that the configurations are periodic with period $p = t_2 - t_1$, up to translation.

Suppose, that the torus being explored is of dimensions $l \times L$ with $l = 3np^3 \max(|i|, 1)$ and $L = 3np^3 \max(|j|, 1)$. The dimensions of the torus are proportional to the non-null scalar components of translation $\vec{t}_{(i,j)}$ *i.e.*, $i3np^3 \equiv 0 \mod l$ and $j3np^3 \equiv 0 \mod L$. This means that,

$$(\vec{t}_{(i,j)})^{3np^3}(C_{t_1}) = \vec{t}_{(i3np^3, j3np^3)}(C_{t_1}) = \vec{t}_{(0,0)}(C_{t_1}) = C_{t_1}.$$

Since translation $\vec{t}_{(i,j)}$ is performed in $p$ rounds, after $p \times 3np^3 = 3np^4$ rounds, all robots will retake their initial positions, so the whole configuration is periodic with period $3np^4$. In this setting, a node is visited infinitely often if and only if it is visited between round $t_1$ and $t_1 + 3np^4$. Now we have to prove that some nodes are left unvisited between round $t_1$ and $t_1 + 3np^4$.

Between time $t_1$ and $t_1 + 3np^4$, each robot visits at most $3np^4$ nodes, hence all the robots visit at most $n \times 3np^4$ nodes after $t_1$. However, there are at least $9n^2p^6 \leq l \times L$ nodes in the torus. Hence, there exist some nodes which are not visited infinitely often, which is a contradiction.

Note that we only proved there are some nodes that are not perpetually visited. Nevertheless, observe that at most $nt_1$ nodes are visited before $t_1$ and we can increase arbitrarily the chosen period $p$ by a factor $f \in \mathbb{N}^*$ without changing the result (in particular $t_1$ does not depend on $f$). By taking $f \geq 1$ such that $9n^2(fp)^6 - 3n^2(fp)^4 > nt_1$, we have that the number of visited nodes (before or after $t_1$) is $nt_1 + 3n^2(fp)^4$

and is smaller than the number of nodes in the torus ($9n^2(fp)^6$), hence there is at least one node that is never visited. This implies that the impossibility also holds for non-perpetual algorithms as well (where each node must be visited at most once). □

We restate the following lemma proven in [**?**].

**Lemma 2.** *A robot with self-inconsistent compass and that sees no other robot, either stays idle or the adversary can make it alternatively move between two chosen adjacent nodes.*

**Theorem 1.** *It is impossible to solve the exploration problem (perpetual or with termination) with two myopic robots equipped with self-inconsistent compasses that agree on a common chirality.*

*Proof.* By Lemma **??**, there is a torus and a configuration where the two robots are at distance $2\Phi + 3$ from each other. In this case the two robots are isolated. By Lemma **??**, the two robots will remain idle or the adversary can make them alternatively move between two nodes, never being in vision from each other and never visiting another node. □

**Theorem 2.** *It is impossible to solve the exploration problem (perpetual or with termination) with three anonymous, oblivious and myopic robots equipped with self-inconsistent compasses that agree on a common chirality.*

*Proof.* By Lemma **??**, there is a torus and a configuration where the distance between the two farthest robots is $2\Phi + 3$ from each other. We have one of the two following possibilities, (i) there are three isolated robots, or (ii) there is an isolated robot and two robots in vision from each other.

In the first case, it is easy to see that the three isolated robots cannot explore the torus because, by Lemma **??**, they have to stay idle or the adversary can make them alternatively move between two nodes, never being in vision from each other and never visiting another node.

In the second case, the two robots that see each other cannot travel together in a direction (because they have the same view). All they can do is get either closer to each other or further from each other. Formally, there is a point $P$ at the middle of the two robots and, if they stay in vision, they will always be at the same distance from that point. The two robots can explore a subgrid $\Phi \times \Phi$ centered at a given middle point. This point is at distance at least $\frac{3\Phi}{2} + 2$ from the isolated robots.

If the two robots in vision gets isolated from one another, they will be at distance $\frac{\Phi}{2} + 1$ from the middle point. In this case, the closest robot to the originally isolated robot will be at distance $\Phi + 1$. Now the three robots are isolated, and, as in the first case, they cannot explore the torus. □

## 5 Visibility range one: $\mathcal{A}_3^2$

We present an algorithm, denoted by $\mathcal{A}_3^2$, which assumes a visibility range one and uses three robots and two colors. By Theorem **??**, $\mathcal{A}_3^2$ is optimal w.r.t. the number of

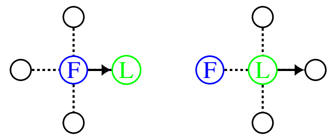


Fig. 1: Rules for moving straight.

robots, and by Theorem **??**, $\mathcal{A}_3^2$ is also optimal w.r.t. the number of colors. Animations are made available online [**?**] to help the reader visualize the algorithm.

The idea of the algorithm is to make the robots alternate between exploring a row and exploring a column. To explore the whole torus, robots move so that all the nodes of the torus are explored. More precisely, after exploring row $r_i$ and column $c_j$, the robots will proceed at exploring row $r_{i-1 \bmod L}$ and then column $c_{j-1 \bmod l}$ and so on.

Initially the robots are co-linear with respectively color $L$, $F$, $F$ [3]. The line of the torus on which they are located is considered as a row. The robot with color $F$ which does not sense the leader moves up changing its color to $L$ while the two other robots move along their current row in the following manner: the robot initially with color $L$ moves away from the one with color $F$ and the remaining robot just follows it.

To explore a row (resp. column), one robot stays idle while the two others travel in a straight line along the nodes of the row (resp. column) being explored until they reach the idle robot. The idle robot is located on an neighboring row (resp. column). The idle robot has color $L$ and is called the *landmark*. The two robots traveling together on a straight line have different colors. One robot, called the *follower*, has color $F$ and the other robot, called the *leader*, has color $L$. To explore a row (resp. column), the two robots have to be next to each other on that row (resp. column). The follower always follows the leader and leader always moves away from the follower. This is done by executing the rules presented in Fig **??**. [4]

The tricky part of this algorithm is how robots switch from exploring a column to exploring a row and *vice versa*. When exploring a column, the robot left behind (aka the landmark) is on the right side of the traveling group. When the leader of the traveling group reaches the landmark, it moves away from the landmark on its current row and updates its color to $F$. Meanwhile, the follower continues to follow the leader. In the next round, the three robots are aligned on the same row. The landmark then moves away from the follower and remains on its row followed by the follower. These two robots become the new traveling group. Whereas the leader, moves to the next row so that it becomes on the left side of the traveling group. That is, the landmark and the leader switch their roles and the new traveling group proceed at the exploration of the row on which there are located. The rules relative to this operation are presented in Fig **??**. The corresponding sequence of configurations is illustrated in Fig **??**.

[3] Note that any reachable configuration can be an initial configuration

[4] In all figures, colored letters inside nodes indicate the color of the robots occupying the nodes. Moreover when a colored letter is given next to a node, it indicates which color the robot will take in the next round.

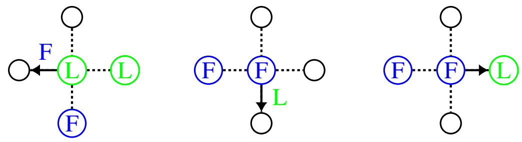


Fig. 2: Rules for switching from moving upward to sideward.

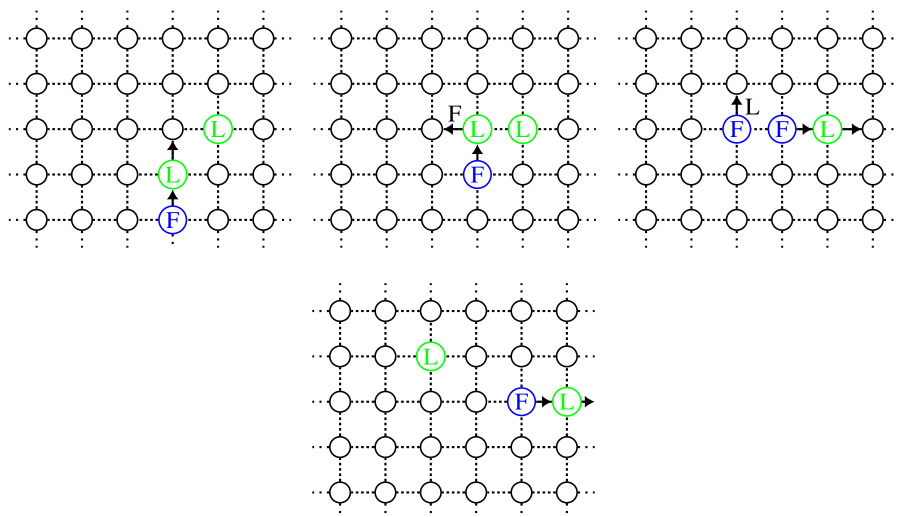


Fig. 3: Sequence of configurations when robots move from exploring a column to exploring a row.

The traveling group are now exploring a row, when they reach the landmark again, the landmark is this time, on the right side. The robots proceed to move to the next column to be explored. More precisely, when the leader reaches the landmark, it continues forward on its current row and changes its color to $F$. The follower also continues to move towards the leader. After one round, the robots will be in a L-shaped form with the two robots colored $F$ and the one robot colored $L$. In the next round, the two robots on the left form the new traveling group and they both move to explore the new column. The robot on the right, moves down and changes its color to $L$, it becomes the new landmark. The set of rules relevant to this sequence is in Fig **??** and the corresponding sequence of configurations are presented in Fig **??**.

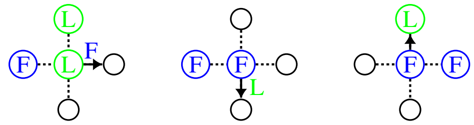


Fig. 4: Rules for switching from moving upward to sideward.

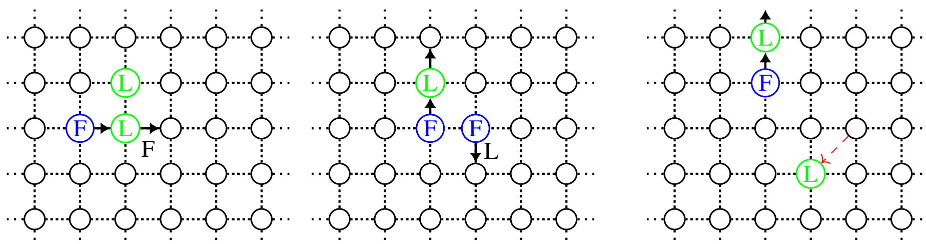


Fig. 5: Sequence of configurations when moving from exploring a row to exploring a column.

It is important to note that every node on a column/row is visited during the exploration of that column/row. Also, the landmark moves two nodes to the left and one node up when going from exploring a column to exploring a row. And, it moves one node to the right and two nodes downward when going from exploring a row to exploring a column. This means that between two consecutive columns (rows) exploration, the landmark moves one node to the left and one node downward.

**Theorem 3.** *$\mathcal{A}_2^3$ solves the PTE problem with three robots and two colors.*

*Proof.* By induction on $l \times L$, where $l$ is the number of columns and $L$ is the number of rows of the torus.

We have validated the base case, for torus of size $4 \times 4$, using our simulation tool. Such a checking is easy since, from a given initial configuration, there is only one possible execution (the algorithm is well-defined and the execution is synchronous). So, we just have to execute the algorithm until reaching an already encountered configuration from which all the nodes have been visited.

We assume now that $\mathcal{A}_2^3$ solves the PTE problem in all tori $x \times y$ with $4 \leq x \leq l$ and $4 \leq y \leq L$ for some values $l, L \geq 4$ and show that $\mathcal{A}_2^3$ solves also the PTE problem in a torus of size $l \times (L+1)$ and $(l+1) \times L$.

Consider first the torus of size $l \times (L+1)$. Then, it is easy to see that after adding one row, our algorithm still solves the PTE problem. Indeed, when robots are traveling upward (*i.e.* they are exploring a row), they move in a straight line periodically until they reach the landmark, so adding one row just increases by one the number of times they perform their periodic movement. And, when robots are traveling sideward (*i.e.* they are exploring a column), they visit all the nodes of the corresponding column.

Now, for the torus of size $(l+1) \times L$. The same argument from the torus of size $l \times (L+1)$ could be used. When robots are traveling sideward, they will perform an extra step for the added column. And, when they travel upward, they will revisit the same nodes visited during the exploration of rows. □

## 6 Visibility range two: $\mathcal{A}_4^1$

We present an algorithm, denoted by $\mathcal{A}_4^1$, which assumes a visibility range two and uses four oblivious robots. $\mathcal{A}_4^1$ is optimal w.r.t. the number of colors. By Theorem **??**, $\mathcal{A}_4^1$ is optimal w.r.t. the number of robots, for oblivious robots. Animations are made available online [**?**] to help the reader visualize the algorithm.

The idea of the algorithm is to make the robots explore the torus rows by rows in a given direction. This is achieved as follows: Three robots, referred to as the traveling group, move to explore three adjacent rows at the same time, and one robot is left behind to be used as their landmark. When the traveling group reaches the landmark, all four robots perform a three rounds sequence to move to the next rows to be explored.

When exploring the rows, the traveling group form a $>$ shape. That is, two robots are located on the same column separated by one empty node, denoted $u$. And, on the right of $u$, the third robot is placed. The three robots move to the right until they sense the landmark. Note that the direction is pointed by the third mentioned robot. Fig **??** presents the rules executed by the robots part of the traveling group.

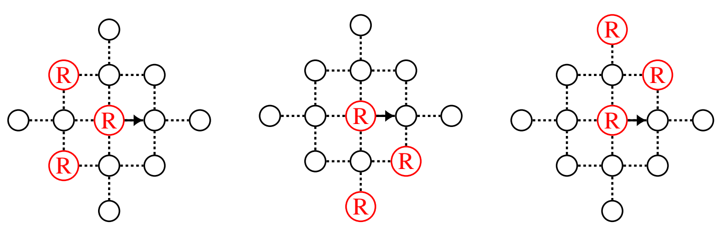


Fig. 6: Rules for three robots moving straight.

The landmark is left behind so that the traveling group knows when they are done exploring the current rows and have to move to the next ones. Note that the landmark is on the same row as the top most robot. When that robot is one node away from the landmark it goes down, same for the landmark since they have the same view. The bottom robot keeps going right because it does not see the landmark. And, the center robot stays idle. After one round, the robots form a T-shape. The rules executed by the robots are presented in Fig **??**.

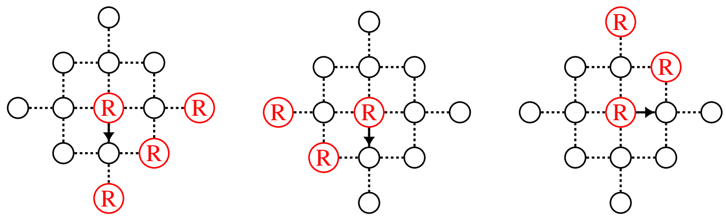

Fig. 7: Rules executed when robots initiate rows change.

From the T-shape, the robot move to create a reverse L shape *i.e.* the two robots in the center of the T-shape move down while the robot on the right goes left. Fig **??** presents the rules executed during this process.

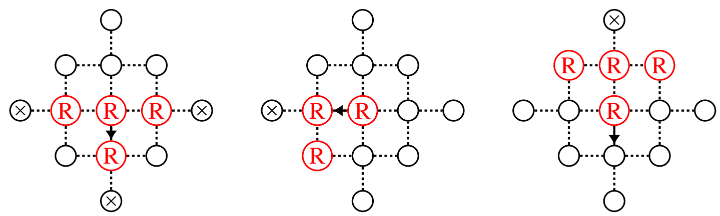

Fig. 8: Rules for the creation of the reverse L shape.

Within the reverse L shape, three robots are co-linear (the ones located on the long side). Among these robots, the one in the middle moves to the right to recreate the $>$ shape while all the other robots remain idle. Refer to the rule presented Fig **??**. That is, after three rounds the robots changes rows and the $>$ shape is built again.

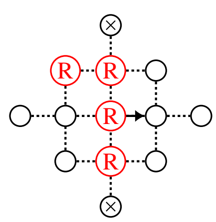

Fig. 9: Rule for restoring the $>$ shape.

Now the three robots on the right form the new traveling group. The robots repeat the same behavior and hence start moving right until they reach the landmark once more. There are two more rules to tell the top most robot in the traveling group to keep following the group even if it sees the landmark at the back. These rules are presented in Fig **??**.

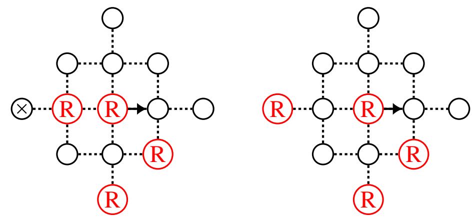


Fig. 10: Rules for the top most robot to keep traveling with the group.

It is important to note that the landmark changes its position two nodes to the right and one node down. The fact that it moves down makes the robots always explore a new row. Fig **??** presents the sequence of configuration during this process.

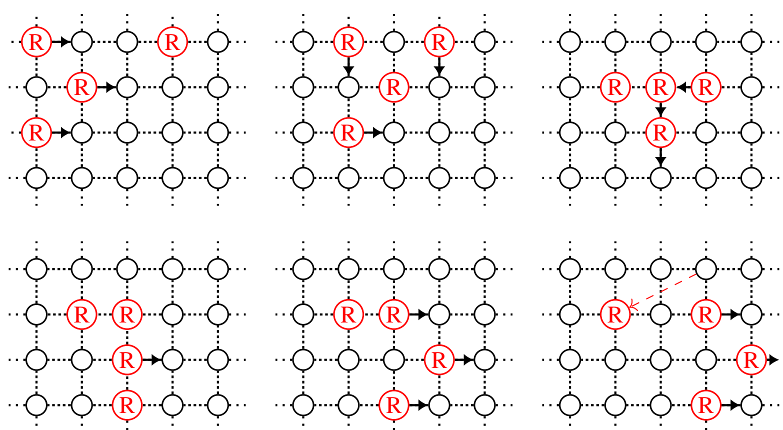

Fig. 11: Sequence for changing rows. The red dashed arrow highlights the movement of the landmark.

Robots form initially the reverse L shape.

**Theorem 4.** *$\mathcal{A}_1^4$ solves the PTE problem with four oblivious robots.*

*Proof.* By induction on $l \times L$, where $l$ is the number of columns and $L$ is the number of rows of the torus.

Similar to the proof of Theorem **??**, we have validated one base case for $l, L = 9$, using our simulation tool.

We assume now that $\mathcal{A}_1^4$ solves the PTE problem in all tori $x \times y$ with $9 \leq x \leq l$ and $9 \leq y \leq L$ for some values $l, L \geq 9$. We should show that $\mathcal{A}_1^4$ solves the PTE problem in the tore of size $l \times (L+1)$ and $(l+1) \times L$.

Consider first the torus of size $(l+1) \times L$. When we add one column the traveling group will have to perform an extra round to reach the landmark again as the robots perform a periodic movement when traveling until they observe the landmark.

Now, consider the torus of size $l \times (L+1)$. When we add a row. The robots will have to perform an extra row exploration: an additional three round sequence to change rows followed by the row exploration.

□

## 7 Conclusion

We presented two optimal solutions for the PTE problem with respect to both the number of robots and the number of colors when robots share a common chirality and have visibility one and two respectively. Indeed, we have shown that three robots endowed with two colors are necessary and sufficient to solve the problem when robots have visibility one and four oblivious robots are necessary and sufficient to solve the problem when robots have visibility two.

One direct open question is to extend the study to consider $(L, l)$-tori such that $l, L < n\Phi + 1$. Ad-hoc solutions might be needed in this case as robots observe the same robots on different sides. Another interesting extension would be to investigate the case in which robots are completely disoriented, *i.e.*, they do not have a common chirality. We conjuncture that three robots remain sufficient to solve the problem with an additional color in the case where robots have visibility one and an additional robot might be needed in the case of oblivious robots with visibility two.