Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Robots without Common Chirality^{*}

Arthur Rauch¹, Quentin Bramas¹, Stéphane Devismes², Pascal Lafourcade³,
and Anissa Lamani¹

¹ University of Strasbourg, ICUBE, CNRS

² Université Grenoble Alpes, VERIMAG

³ University Clermont Auvergne, CNRS UMR 6158, LIMOS

Abstract. We consider swarms of luminous myopic robots that run in synchronous Look-Compute-Move cycles. These robots evolve in a finite grid and are disoriented *i.e.*, they have no global compass nor a common chirality. In this context, we propose optimal solutions to the perpetual exploration of a finite grid. Precisely, we investigate optimality in terms of the visibility range and number of robots. In more detail, under the optimal visibility range one, we give an algorithm which is optimal w.r.t. the number of robots: it uses three robots and three colors. Under visibility two, we design an algorithm that uses only one color (*i.e.*, robots are oblivious).

Keywords: Luminous myopic robots, perpetual exploration, finite grid, exclusiveness.

1 Introduction

We consider in this paper *luminous robots* [16] *i.e.* a swarm of autonomous robots endowed with visibility sensors, motion actuators, and lights of different colors. Robots operate in cycles that comprise three phases: Look, Compute and Move. During the look phase, robots sense their surrounding then, in the second phase, using the taken snapshot, they compute a destination and update their light color, and finally in the last phase, they move to the computed destination (if any).

We investigate the computational power of such swarms and hence consider luminous robots with limited capabilities. First, they are *myopic* and hence are only able to sense their surroundings within a limited visibility range. Furthermore, they have neither a *global compass* nor a *common chirality*. That is, they are completely disoriented. Finally, except from their lights, robots do not have persistent memories and cannot communicate directly.

^{*} This study was partially supported by the French ANR projects ANR-16-CE40-0023 (DESCARTES) and ANR-16 CE25-0009-03 (ESTATE).

We are interested in coordinating such weak robots to solve a perpetual common task which is the *perpetual exploration*. In this problem, given a space which is partitioned into locations, each of these locations should be visited by at least one robot infinitely often. Here, we consider a finite discrete space which is conveniently represented as a graph, where nodes represent locations that a robot can occupy at a given time and edges represent the possibility for a robot to move from one location to another.

In this paper, we investigate optimal *exclusive* solutions to the perpetual exploration of a *finite grid* with respect to both the visibility range and the number of robots. Exclusiveness [1] requires any two robots to never simultaneously occupy the same position nor traverse the same edge.

Related Work. The exploration problem is a well known problem that has been excessively investigated. Various topologies have been considered, *e.g.*, lines [13], rings [2,14,10,15,7], trees [12], torus [9], finite [8,3] and infinite grids [4,5].

Various investigations considered the exploration problem with an additional condition that requires the robots to detect the end of the exploration and stop moving [12,13,14,10,7,9,8]. This version of the problem is called *terminating* exploration problem. Some other investigations, considered the exploration problem with the additional constraint that every node is infinitely often visited by at least one robot [2,3]. This version of the problem is called *perpetual* exploration problem.

Most of the investigations in the literature have been devoted to “non-myopic” robots, *i.e.*, robots with an unbounded visibility range allowing them to sense all their surrounding and see the positions of all the robots in the system; see [12,2,3,13,14,10,9,8]. In such a context, robots are always assumed to be anonymous and oblivious, *i.e.*, they have no state and cannot remember past actions. Exploration algorithms satisfying exclusiveness are proposed in both finite [2,3] and infinite graphs [4,5].

Chirality is usually assumed in the 2D Euclidean plan; see for example [11]. However, recently, few works dedicated to discrete environment, *e.g.*, in (infinite) graphs [4], investigated the exploration problem assuming robots which have a common chirality. This has an impact when optimal solutions are considered. Indeed, for example, with visibility range one and few colors ($O(1)$), five (resp. six) synchronous robots are necessary and sufficient to explore an infinite grid with (resp. without) the common chirality assumption [4,5].

A recent work [6] studied the exploration problem in finite grid, with myopic, synchronous and luminous robot (like our model) but assuming robots agree on a common chirality. They showed that, two robots with three colors and a common chirality are sufficient to solve the problem. They also proved that three robots with visibility range 2 are necessary and sufficient when robots have only one color (still without chirality).

To the best of our knowledge, the present work is the first study of (perpetual) exclusive exploration with myopic (luminous) robots in finite grids with robots without chirality.

Contribution. Our contribution is threefold. First we prove that the problem is not solvable with only two robots having any finite number of colors and finite visibility range. Second, we present an optimal algorithm that solves the problem with only three robots having visibility range one, using only three colors. Third, we present an algorithm solving the problem using 5 oblivious (only one color) robots with visibility range two. This algorithm is also optimal with respect to the visibility range, because using the lower bound given in [6], which applies here, states that no oblivious algorithm with visibility range one can solve the problem. Table 1 summarizes our contributions.

Visibility	# Robots	# Colors	Algorithm
finite	2	finite	Impossible (Thm. 1)
1	finite	1	Impossible ([6])
1	3	3	Vone ₃ ³
2	5	1	Vtwo ₁ ⁵

Table 1: Summary of our results.

Roadmap. Section 2 is devoted to the computational model and definitions. In Section 3, we prove our impossibility result and we present our algorithm in Sections 4 and 5,

2 Model

We consider a set of $n > 0$ robots located on a *finite grid* made of $\mathcal{L} \geq n$ lines and $\mathcal{C} \geq n$ columns,⁴ *i.e.*, robots evolve in the undirected graph $G(V, E)$ where $V = \{(i, j) : i \in [0, \mathcal{C} - 1], j \in [0, \mathcal{L} - 1]\}$ and $E = \{\{(i, j), (k, l)\} : (i, j) \in V \wedge (k, l) \in V \wedge |i - k| + |j - l| = 1\}$. The size of the grid is then $\mathcal{L} \times \mathcal{C}$. Grid coordinates are used for the analysis only, *i.e.*, robots cannot access them.

We assume discrete time and 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. Φ 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. 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.

⁴ The requirement on the numbers of lines and columns is only made for the sake of simplicity.

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 Φ from them. We denote by Cl the set of all possible colors.

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 Φ). 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.

Configurations A *configuration* C in a grid $G(V, E)$ 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 Φ 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 (see, for example, Fig. 1): 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:

1. the rotations of angle 0 (to have the identity), $\pi/2$, π and $3\pi/2$, centered at r ,
2. the mirroring (robots cannot distinguish between clockwise and counter-clockwise), and
3. any combination of rotation and mirroring.

Here, 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 grid upside-down (f is the π -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) .

A robot is said to be *isolated* when the only robot in its view is itself.

Algorithm An algorithm A is a tuple $(Cl, Init, T)$ where Cl is the set of possible colors, $Init$ is a mapping from any considered grid to a non-empty set of initial configurations in that grid, and T is the transition function $Views \rightarrow \{Idle, Up, Left, Down, Right\} \times Cl$, 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 $f^{-1}(T(f(V_C(i, j)))) = (Idle, c)$,
- $((i - 1, j), c') \in C$ and $f^{-1}(T(f(V_C(i - 1, j)))) = (Right, c)$,
- $((i + 1, j), c') \in C$ and $f^{-1}(T(f(V_C(i + 1, j)))) = (Left, c)$,
- $((i, j - 1), c') \in C$ and $f^{-1}(T(f(V_C(i, j - 1)))) = (Up, c)$, or
- $((i, j + 1), c') \in C$ and $f^{-1}(T(f(V_C(i, j + 1)))) = (Down, c)$.

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 A in a grid G is then a sequence $(C_i)_{i \in \mathbb{N}}$ of configurations such that $C_0 \in Init(G)$ and $\forall i \geq 0, C_i \mapsto C_{i+1}$.

Perpetual Finite Grid Exploration An execution $(C_i)_{i \in \mathbb{N}}$ in a grid $G = (V, E)$ achieves the *Perpetual Finite Grid Exploration* (PFGE) if for every node $u \in V$ and for every time t , there exists a time $t' \geq t$ such that u is occupied in $C_{t'}$.

An algorithm A that uses n robots solves the *Perpetual Finite Grid Exploration* (PFGE) problem if for every finite grid $G = (V, E)$ with at least n lines and n columns and every initial configuration $C_0 \in Init(G)$, we have every execution of A in G starting from C_0 that achieves the PFGE.

An Algorithm as a Set of Rules We write an algorithm as a set of rules, where a rule is a triplet $(V, d, c) \in Views \times \{Idle, Up, Left, Down, Right\} \times Cl$.

We say that an algorithm $(Cl, Init, T)$ includes the rule (V, d, c) , if $T(V) = (d, c)$. By extension, the same rule applies to indistinguishable views, *i.e.*, $\forall f \in \mathcal{IT}, T(f(V)) = (f(d), c)$. Consequently, we forbid an algorithm to contain two rules (V, d, c) and (V', d', c') such that $V' = f(V)$ for some $f \in \mathcal{IT}$, except the rules that are ambiguous by construction *i.e.*, when the view has an axis of symmetry for instance (see an example below, illustrated in Figure 2). Hence, an algorithm corresponds to a set of rules if each destination is the result of applying one of its rules.

As an illustrative example, consider the rule R_1 given in Fig. 1. This rule is defined for robots having a visibility range of two. This rule means that, when a blue robot B sees three robots with color R , one on top and one in diagonal, then the blue robot is dictated to move Up. By extension the same rule applied if the view is rotated by π , but in that case, the destination would be Down.

In the same figure, Rule R_2 is a rule where the three black nodes represent a part of the outer boundary of the grid, that we call a *wall* in the remaining of the paper. In our algorithms, we often define similar rules that apply regardless of the presence of a wall in some part of the view. Thus, to avoid defining several

time rules with very similar views, we propose a notation to represent several rules in just one picture. For example, Rule R_3 in Fig. 1 has three nodes hatched with vertical lines, which means that the rule applies regardless of the presence of a wall located at those nodes. In practice, every rule that contains such vertical (resp. horizontal) hatched lines, represents a set of rules obtained by replacing each of those lines either by walls, or by empty nodes. For example, Rule R_3 in Fig. 1 is a concise representation of Rules R_1 and R_2 .

Figure 2 defines an ambiguous rule. The robot has a symmetric view so that, depending on the transformation f chosen by the adversary, the robot executing this rule moves either left or right.

Fig. 2: Example of an ambiguous rule.

Fig. 1: Examples of rules.

Algorithms having locally-defined initial configurations In a given grid, the set of possible initial configurations of an algorithm can be reduced to a singleton. In such a case, the scalability and flexibility of the algorithm is weak. To be more general, we propose algorithms with *locally-defined* sets of initial configurations. Configurations in a locally-defined set of initial configurations are defined by colors and relative positions of the robots only. Hence, every two possible initial configurations are equal up to a translation applied to all robot positions and for a given grid, the set of all possible initial configurations is closed by such translations.

3 Impossibility Results

It has been shown in [6] that the PFGE problem is not solvable using only one robot for any finite visibility range. We prove in the following that the PFGE problem is also not solvable using two robots if they have a visibility range of one. **In this section we assume all the robots have visibility range one.**

First, we observe that in large enough grids, if robots travel a long distance without seeing a wall, then they must execute a periodic sequence of movements. The maximum number of distinct position and colors two robots en-

dowed with $|C|$ colors can have is the number of 2-combination with repetitions $B = \binom{|C|+1}{2} = \frac{|C|(|C|+1)}{2}$. Thus, if robots travel a distance at least B without seeing a wall, then they are executing a periodic sequence of movements. The definition of B depends on the algorithm and is used throughout this section.

The above observation is important to prove our impossibility results. First, we use it to show that once robots move far away from the walls, their movements are restricted. In more details, they can only move in straight line (Lemma 1) otherwise they cannot explore the grid (Lemma 2). The proof are straightforward so, due to space constraints, we only provide a sketch and move the full proof to the appendix.

Lemma 1. *Let A be an algorithm solving the PFGE problem with two robots. If there exists an execution with a configuration C where robots are at distance at least $2B$ from any walls and, from C , the robots perform a periodic sequence of movements with no ambiguous rules, then the robots move in a straight line until they reach a wall.*

Sketch of proof: When a robot executes unambiguous rules, it can only move from or towards the other robots, hence remains on the same line. Indeed, any view containing another robot has an axis of symmetry passing through the other robot (recall that we assume visibility range 1), and the destination of an unambiguous rule must be on the axis as well. $\square$

Lemma 2. *Let A be an algorithm solving the PFGE problem with two robots. If there exists an execution with a configuration C where robots are at distance at least $2B$ from any walls and, from C , robots perform a periodic sequence of movements which includes an ambiguous rule, then there exists an execution where the exploration is not completed.*

Sketch of proof: Every time robots execute an ambiguous rule, robots are making a turn, and the adversary can decide on which side the robots are turning. If the periodic sequence of movements contains an ambiguous move, the robots will make at least one turn per period, hence the adversary can make the robots remain in the same square grid of size B (the period of the sequence is at most B). While doing so, the robots do not see any wall, and do not explore all the grid. $\square$

The next Lemma states that, if two robots are on the same line, this line must be an axis of symmetry of their views and they cannot break this symmetry without executing an ambiguous rule (due to the lack of chirality agreement). Hence, the adversary can decide which side of the line the robots move. More formally, for any rectangle having the same line as axis of symmetry, the adversary can decide on which side of the rectangle the robots move out of it (the adversary can decide the nodes used by the robots to leave the rectangle, as long as the nodes, and their symmetric, enclose the rectangle).

Lemma 3. *Let A be an algorithm solving the PFGE problem with two robots. Let C be a configuration where the two robots are on the same line L . Let R be a set of nodes delimiting a rectangle for which L is an axis of symmetry. Let $R_1 \subset R$ such that the union of R_1 and the symmetric of R_1 , with respect to L , is equal to R . Then, starting from C , there exists an execution where a robot reaches a node in R_1 .*

Now we can prove our main Lemma, which states that the robots cannot move further than a distance of $3B$ from all the walls. To do that, we need two definitions. A *corner box* is the set of nodes forming a square of size $2B$ including a corner of the grid. We say robots are in a *T-configuration* when they are adjacent, only one is adjacent to a wall, and they both are at distance at most $3B$ from another wall.

Lemma 4. *If A solves the PFGE problem with two robots, then, if at a given time t , a robot is at a corner box or if robots are in a T-configuration, then there exists an execution such that the robots ends up a time $t' > t$ either in a corner box or in a T-configuration, and between time t and t' the robots remain at distance at most $3B$ from a wall.*

Proof. First assume the robots are in a corner box in a configuration C (the case where robots are in a T-configuration is treated in the last paragraph of this proof). To explore the grid, the robots must leave the corner box. We denote by W_1 and W_2 the two walls adjacent to the corner contained in the corner box where robots are located, see Figure 3 that illustrate the notations. Without loss of generality, we assume that at time t_0 , the first robot, say r , leaving the corner box of size $2B$ is at distance $2B + 1$ from W_1 (hence at distance at most $2B$ from W_2). We can also assume that r is the only robot leaving the corner box at time t_0 , because if both robots leave the corner box at the same time t_0 , it means they executed an ambiguous move at time $t_0 - 1$ so that there exists an execution where the adversary mirrors the views of the robots in order to keep them in the corner box.

When leaving the corner box, the two robots are on a line parallel to the wall W_2 (See Figure 3). Two cases can occur. In the first case they remain on the same line parallel to W_2 , moving away from W_1 , until a robot is at distance $3B + 1$ from W_1 (case 1 in the Figure). If they do so, since, they traveled a distance B , they are executing a periodic sequence of movements, hence, they continue to move on the same line until they reach the wall opposite to W_1 , in a T-configuration.

In the second case, they perform an ambiguous move to make a turn at time t_1 and end up in a line L parallel to W_1 . We consider the set of nodes $R_1 = R_1^{corner} \cup R_1^{right}$ where R_1^{corner} is the segment of nodes at distance $2B$ from the wall W_1 and with distance to W_2 in the interval $[0, 2B]$, and R_1^{right} is the segment of nodes at distance $2B$ from W_2 and at distance from W_1 in the interval $[2B, d_1]$, where d_1 is the distance of the robots to W_1 (see Figure 3). The union of R_1 with its symmetric with respect to L delimits a rectangle (dotted

line in the figure) so that, using Lemma 3, there exists an execution such that a robot reaches R_1 .

If a robot reaches R_1^{corner} , then a robot reaches a corner box and the Lemma is proved. If a robot reaches R_1^{right} , then the robots have traveled a distance at least B without seeing a wall, hence are executing a periodic sequence of movements. The sequence cannot contain an ambiguous rule (using Lemma 1) because the robots are at distance at least $2B$ from any wall, so they are moving in a straight line (by Lemma 1), and they end up in the wall opposite to W_2 and reach a T -configuration, while remaining at distance at most $3B$ from W_1 .

Now we consider that the robots are in a T -configuration in configuration C . Then, they are on a line L perpendicular to a wall, say W_2 . Using a similar argument, we know that either the robot enter the closest corner box, or move in a straight line to the opposite wall until they reach a T -configuration. $\square$

Fig. 3: Illustration of the different cases in Lemma 4

We can now prove our impossibility result.

Theorem 1. *The PFGE is not solvable with a only two robots with visibility range 1, for any bounded number of colors.*

Proof. Assume that algorithm A solves the problem. Consider a grid of size $8B \times 8B$. Since the robots explore the entire grid, there exists a round when a robot is in a corner box. By Lemma 4 repeatedly, we can construct a execution from there where robots remains at distance at most $3B$ from any wall, so that nodes at distance more than $3B$ from all the walls are not visited anymore, a contradiction. $\square$

4 Visibility range one: Vone_3^3

We present in this section an algorithm, denoted by Vone_3^3 , which assumes a visibility range equal to one and uses three robots endowed with three colors. By Theorem 1, Vone_3^3 is optimal with respect to the number of robots. We

Fig. 4: Move in a straight line

Fig. 5: Beginning of the exploration of a line

Fig. 6: Sequence of configurations during a turn around

encourage the reader to follow the explanation of the algorithm while looking at the animations available online [17], published as an additional material.

The algorithm defines three roles for the robots using the colors: *L* (*leader*), *F* (*follower*), *X* (*landmark*). The roles are not fixed and robots may change roles during the execution and roles do not always correspond to their default meanings. Initially, the three robots are aligned, two of them have color *L* while the third one is *F*. The initial positions of the three robots are arbitrary in the grid, except that they have to be located on the same line and the two robots with color *L* needs to be adjacent (we call such configurations, *LLF* configurations). Hence, the set of all possible initial configurations is locally-defined.

The initial configuration is such that the robots can prepare the perpetual exploration, but may never be reach again. Indeed, the goal is to reach a configuration C_p from which the robots will perform a periodic journey around the grid, visit all the nodes, and come back to configuration C_p , to start again the same journey over and over. We first explain how this journey is build, which is most important part of our algorithm, and it will be straightforward to show that robots reach a configuration of the journey starting from the initial configuration previously defined.

The idea of the algorithm is to make two robots move and explore a given line while the third robot remains idle to keep track of the next line to be explored. Every time a line is explored, the robots, including the landmark robot, move “up” by one row (assuming, for the illustration, that robots are visiting a line from left to right). Then, once the robots reach a corner, they change their direction and repeat the same process. More precisely, the role of both the leader and the follower is to explore a given line. Indeed, it is easy to make these two robots move on the same direction; the leader moves away from the follower

Fig. 7: Turn around

Fig. 8: Up and Turn

Fig. 9: Up and turn. This sequence occurs when the robots are not in a corner. The case when the robots are in a corner is presented in Figure 11

while the follower, as its name suggests, follows the leader. The rules executed those two robots to move in a straight line are presented in Figure 4.

During the line exploration by the leader and the follower, the third robot is left at the neighborhood of a wall on an adjacent line with the leader role, refer to Figure 5.

When the leader and the follower reach the wall, as they do not have any sense of direction, the idea is to make them move back and cross the same line again. For this purpose, they need to exchange their respective position. This is done as follows: the first robot to detect the wall is the leader, the leader moves to its adjacent empty node (in case of symmetry, the scheduler chooses the direction to take) changing its color to X . In the next round, the follower reaches the wall and observes the landmark (the previous leader). The follower knows that they are moving back to cross the same line again. That is, the follower moves back to its previous position followed by the landmark. The follower then becomes the leader while the landmark becomes a follower and they both start exploring again their current line. Figure 7 presents the rules that are executed during the moving back process while Figure 6 shows the sequence of configurations during this process.

When the robots reach once again the wall, they can observe this time, in turn, a robot located on a different line at the neighborhood of the wall. Hence, an orientation can be defined to indicate the next line to be explored *i.e.* the empty line that is adjacent to the one hosting the pair of robots. The idea is to make the robots move to the next line in such a way they can repeat the same

Fig. 10: Last corner preparation - Rules

Fig. 11: Corner turn. After the sequence, the exploration continues as before, but everything is rotated by a clockwise angle of π

Fig. 12: First “up and turn” after the corner turn

behavior. The lines of the grid are then explored in a given direction one by one until they reach the last line. The rules that are executed by the robots to change the lines; when they observe the third robot, are presented in Figure 8. Figure 9 illustrates the sequence of configurations during the line change.

Given an orientation of the grid, assume without loss of generality that the robots are exploring the grid line by line in a given direction. As the grid is finite, eventually, the robots change lines to reach the last one with respect to the current exploring direction. When this happens, the robots change the exploring direction by a clockwise angle of π . The robots then exhibit the same behavior as previously and hence explore the lines of the grid with respect to the new orientation. Note that this change of direction is initiated by the first robot to join the last line; as it is located at a corner. The change of direction is done through several rules that are presented in Figure 10 while the sequence of configurations during this process are presented in Figure 11.

Lastly, in order to start from locally-defined initial configuration, robots may start from the middle of the grid. Recall that we assume that the three robots are aligned and their colors are L , L and F respectively. The goal is to move

Fig. 13: Rules used by the robots to reach the wall, starting from an *LLF* configuration

Fig. 14: Rules used by robots to handle grid with only 3 columns

Fig. 15: Sequence of configurations when the grid has only 3 columns

straight to a wall and then, once the robot with color *L* seeing the wall moves on either side, the configuration is exactly the same as the first one of an “up and turn” sequence. So after that, the periodic journey can start. The rules used by the robots to do this are shown in Figure 13. If the robots in an *LLF* initial configuration are all adjacent to a wall, then we also defined few rules in order for the robot to reach, after one round, a configuration of the periodic journey, hence, robots behave exactly as explained previously.

For the 3 grid case, a specific rule is needed as any “up and turn” sequence is considered to be done at a corner. The rule is shown in Figure 14 and the sequence of movement when the grid has only 3 columns is shown in Figure 15 (the specific rule is used in the fifth round of the sequence).

Theorem 2. Vone_3^3 solves the perpetual exploration problem with three robots, having three colors and visibility range one.

Sketch of proof: Using our simulation tool, we are able to prove that our algorithm is correct for any grid $n \times m$, with $m, n \in \{3, 4\}$. Then, we show that when a group of robot is traveling along a row, adding a column does not change the

relative position of the robots when they reach a wall. Similarly, adding a row does not change the relative position of the robots when they reach a corner. The sequence of movement performed in a corner does not depend on the size of the grid, so that, regardless of the size of the grid, the robots explore the entire grid in a perpetual manner.

Due to the lack of space, the full correctness proof of Vone_3^3 is omitted here but can be found in the appendix.

5 Visibility range two : Vtwo_1^5

In this section, we present our second algorithm that works with five robots that have a visibility range of 2, but are oblivious (*i.e.*, they all have the same fixed color). Again, we encourage the reader to follow the explanation of the algorithm while looking at the animations available online [17]. The algorithm is similar, and simpler, than the previous algorithm. Indeed, the robots remain grouped together, and they move from left to right, without rotating their course. Every time the group of robots reaches a wall, they perform a turn sequence to move back to the opposite wall, one row above or below, depending on the current orientation of the group (see Figure 18 for a turn one row below). After moving straight (See Figure 16) to the opposite wall, everything is mirrored so they do the same. They move back and forth until they reach the top wall. After following the top wall (using a specific periodic sequence of movements, see Figure 17), they make a special turn in order to move back and forth in the other direction. Due to space constraints, all the rules are shown in the appendix.

Fig. 16: Move in a straight line

Fig. 17: Follow the wall

Fig. 18: Turn around

Fig. 19: Turn at a corner

6 Conclusion

We have investigated the perpetual exploration problem by myopic and luminous robots which evolve in a finite grid without any sense of direction. We have proposed optimal synchronous solutions with respect to either the number of robots or the visibility range. Indeed, we have first shown that if robots have only a visibility range one then, the problem is not solvable with only two robots regardless of the number of colors. Then, we have presented Vone_3^3 that uses three robots and three colors which is an optimal solution with respect to the number of robots.

Next, following the impossibility result of [6], our algorithm Vone_1^5 , which uses five oblivious robots and visibility range two is optimal with respect to the visibility range.

Direct open questions to this work are to determine whether Vone_3^3 is also optimal with respect to the number of colors and whether Vone_1^5 is optimal with respect to the number of robots. Also, it would be interesting to extend the investigation on other topologies such as torus shaped networks.

References

1. R. Baldoni, F. Bonnet, A. Milani, and M. Raynal. Anonymous graph exploration without collision by mobile robots. *Inf. Process. Letters*, 109(2):98–103, 2008.
2. L. Blin, A. Milani, M. Potop-Butucaru, and S. Tixeuil. Exclusive perpetual ring exploration without chirality. In N. A. Lynch and A. A. Shvartsman, editors, *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings*, volume 6343 of *Lecture Notes in Computer Science*, pages 312–327, Boston, Massachusetts, USA, September 2010. Springer.
3. F. Bonnet, A. Milani, M. Potop-Butucaru, and S. Tixeuil. Asynchronous exclusive perpetual grid exploration without sense of direction. In A. F. Anta, editor, *Proceedings of International Conference on Principles of Distributed Systems (OPODIS 2011)*, number 7109 in *Lecture Notes in Computer Science (LNCS)*, pages 251–265, Toulouse, France, December 2011. Springer Berlin / Heidelberg.
4. Q. Bramas, S. Devismes, and P. Lafourcade. Finding Water on Poleless using Melomaniac Myopic Chameleon Robots. In *FUN 2020, 10th International Conference on Fun with Algorithms*, Favignana, Sicily, Italy, September 28-30 2020. LiPICS. to appear.

5. Q. Bramas, S. Devismes, and P. Lafourcade. Infinite Grid Exploration by Disoriented Robot. In *NETYS'2020, the 8th International Conference on NETworked sYStems*, Marrakech, Morocco, June 3-5 2020. Springer. to appear.
6. Q. Bramas, P. Lafourcade, and S. Devismes. Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Opaque Robots with Common Chirality. 2021.
7. A. K. Datta, A. Lamani, L. L. Larmore, and F. Petit. Enabling ring exploration with myopic oblivious robots. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IPDPS 2015*, pages 490–499, Hyderabad, India, May 25-29, 2015 2015. IEEE Computer Society.
8. S. Devismes, A. Lamani, F. Petit, P. Raymond, and S. Tixeuil. Terminating exploration of a grid by an optimal number of asynchronous oblivious robots. *The Computer Journal*, 1, 03 2020. to appear.
9. S. Devismes, A. Lamani, F. Petit, and S. Tixeuil. Optimal torus exploration by oblivious robots. *Computing*, 101(9):1241–1264, 2019.
10. S. Devismes, F. Petit, and S. Tixeuil. Optimal probabilistic ring exploration by semi-synchronous oblivious robots. *Theoretical Computer Science (TCS)*, 498:10–27, 2013.
11. Y. Dieudonné, F. Petit, and V. Villain. Leader election problem versus pattern formation problem. In N. A. Lynch and A. A. Shvartsman, editors, *Distributed Computing, 24th International Symposium, DISC 2010*, volume 6343 of *Lecture Notes in Computer Science*, pages 267–281, Cambridge, MA, USA, September 13-15 2010. Springer.
12. P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots. *Theor. Comput. Sci.*, 411(14-15):1583–1598, 2010.
13. P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. How many oblivious robots can explore a line. *Inf. Process. Lett.*, 111(20):1027–1031, 2011.
14. P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. *Algorithmica*, 65(3):562–583, 2013.
15. F. Ooshita and S. Tixeuil. Ring exploration with myopic luminous robots. In T. Izumi and P. Kuznetsov, editors, *Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018*, volume 11201 of *Lecture Notes in Computer Science*, pages 301–316, Tokyo, Japan, November 4-7 2018. Springer.
16. D. Peleg. Distributed coordination algorithms for mobile robot swarms: New directions and challenges. In A. Pal, A. D. Kshemkalyani, R. Kumar, and A. Gupta, editors, *Distributed Computing – IWDC 2005*, pages 1–12, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
17. A. Rauch, Q. Bramas, S. Devismes, P. Lafourcade, and A. Lamani. Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Robots without Chirality: The Animations, Mar. 2021.

A Omitted Proofs

Recall that we assume in this section that all the robots have visibility range one.

Lemma 5. *If A solves the PFGE problem with two robots, then, in any execution E of A and in any configuration of E where robots are at distance at least 3 from any walls, no robot is isolated.*

Proof. Assume by contradiction that there exists a configuration C of E such that the two robots are isolated while at distance 3. Let us refer to the two robots by r_1 and r_2 respectively such a configuration exists. Let C be a configuration where a robot r_1 is isolated (hence the other robot r_2 is also isolated), and both robots are at distance at least 3 from any walls. Let a_1 , resp. a_2 , be the position of r_1 , resp. r_2 , in C . Among the three nodes adjacent to a_1 , at least three are not adjacent to a_2 . Let b_1 be one of such nodes. Similarly, we can find a node b_2 adjacent to a_2 that is neither adjacent to a_1 nor to b_1 .

If r_1 remains located at a_1 or b_1 and r_2 remains located at a_2 or b_2 , then the two robots remain isolated, and every time a robot r_i , $i = 1, 2$, is dictated to move (according to an ambiguous rule), the adversary can choose the indistinguishable transformation f such that its destination is in $\{a_i, b_i\}$. Hence, from the C the exploration cannot be performed, a contradiction. $\square$

Lemma 1. *Let A be an algorithm solving the PFGE problem with two robots. If there exists an execution with a configuration C where robots are at distance at least $2B$ from any walls and, from C , the robots perform a periodic sequence of movements with no ambiguous rules, then the robots move in a straight line until they reach a wall.*

Proof. Let A be an algorithm solving the PFGE problem with two robots. Suppose that from a configuration C where the robots are at distance more than $2B$ from any walls, they execute a periodic of sequence of movements with no ambiguous rules. According to the previous lemma, since the robots are at distance more than $2B > 3$ from any walls, they cannot be isolated in C . Moreover, this periodic sequence of movements cannot result in an idle motion or, since A solves the PFGE problem, an immediate contradiction occurs. Therefore, the two robots are adjacent in C and cannot see any walls. Considering the robots can only execute unambiguous rules and their only point of reference is the neighboring robot, they can only move towards the other robot or away from the other robot. If both robots were to move towards each other, they would swap their positions. Since only using such rules would make the robots alternate positions infinitely causing a contradiction, another kind of rule is necessary. The robots can at most use B rules while none of them sees a wall. Since the robots can at most move B times during one period, they cannot be at distance less than 3 from any walls after one period. This means that according to lemma 5, the

robots cannot both move away from the other as they would be isolated at a distance at least 3 from any walls. Therefore, one robot r_a must move towards the other robot r_b while r_b must move away from r_a , making the robots move in a straight line in the direction taken by r_b . Let P be the set of rules comprising the periodic movement, $P_d \subseteq P$ be the set of rules causing both robots to move towards a direction d and $P_{d'} \subseteq P$, with $P_d \cap P_{d'} = \emptyset$ be the set of rules causing both robots to move in the opposite direction d' of d . If $|P_d| = |P_{d'}|$, then after one period, the robots still occupy the same positions, and become trapped in an idle periodic movement, a contradiction. If $|P_d| > |P_{d'}|$, then the robots move in a straight line towards the direction d for a period and, by repeating this period, will continue their movement in a straight line towards d until the loop is broken by a robot seeing something other than the another robot, because this view hasn't been encountered in previous periods and P would not include a corresponding rule. Since the robots are always adjacent during the periodic movement, this can only correspond to a wall. The same logic applies with the direction d' if $|P_d| < |P_{d'}|$. We conclude that if two robots perform a periodic sequence of movements without ambiguous rules, then they are moving in a straight line. $\square$

Lemma 2. *Let A be an algorithm solving the PFGE problem with two robots. If there exists an execution with a configuration C where robots are at distance at least $2B$ from any walls and, from C , robots perform a periodic sequence of movements which includes an ambiguous rule, then there exists an execution where the exploration is not completed.*

Proof. Let A be an algorithm solving the PFGE problem with two robots. Suppose there exists an execution of A in G that contains a configuration C where robots are at distance at least $2B$ from any walls. According to the lemma 5, the robots cannot be isolated in C and are therefore adjacent. Suppose that from C , the robots perform a periodic sequence of movements which includes at least one ambiguous rule. According to lemma 5, this movement cannot result in the robots being isolated, otherwise a contradiction will occur. If robots are on a vertical line L , the possible ambiguous destination for both robots correspond to moving either left or right. If both robots move to either left or right, then the adversary can choose two indistinguishable transformations such that the robots move in two different directions in the global view, resulting in the robots being isolated, a contradiction. Therefore, if one robot executes an ambiguous rule, then the other must execute an unambiguous rule in order for both robots to remain adjacent. That rule can either be moving towards the other robot or moving away from the other robot. However, if a robot moves away from the robot executing an ambiguous rule, then both robots become isolated. The robot executing an unambiguous rule must therefore move towards the other robot so that they both reach a line L' perpendicular to L . We will refer to this behavior as a *turn*. When the robots execute a turn, the adversary can choose the relative position of the robots *i.e.*, for instance whether r_1 precedes r_2 or the r_2 precedes r_1 on L'

Next, we define the direction taken by robots executing a sequence of unambiguous movements starting from given positions of the robots. Using the same reasoning as for ambiguous rules, we deduce that in order to move with unambiguous rules, one robot r_a must move towards the other robot r_b , while r_b must move away from r_a . Then, a sequence of one or more unambiguous rules consists of idle rules, rules causing straight line movement in a direction d , and rules causing straight line movement in the opposite direction to d , denoted d' . If this sequence of unambiguous rules contains more rules towards d than towards the direction d' , then we say that *the robots are moving towards d* . Moreover, since the robots can use at most B rules, then they cannot move more than $B/2 - 1$ times towards d' before moving towards d . If this sequence contains the same number of movements towards d than towards d' , then we consider that the robots can at most move $B/2$ times towards d or d' before going back to their respective initial position.

With this definition, we can build an execution where the robots never reach a wall. In this execution, each time the robots take a turn which is followed by a sequence of unambiguous movements (which can be empty if the robots make two consecutive turns), the adversary can choose the position of the robots after the turn such that the direction of the sequence of unambiguous movements is away from the closest wall.

In our execution, the robots can travel a distance at most B in a given direction before making a turn, and then move a distance $B/2$ in the same direction, regardless of the choice of the adversary. Since we assumed that the robots start at distance at least $2B$ from any walls, the robots will never be able to reach a wall and therefore will not be able to visit all the grid. Hence the PFGE problem cannot be solved by **A** if there is at least one turn during the perpetual movement, a contradiction. $\square$

Lemma 3. *Let A be an algorithm solving the PFGE problem with two robots. Let C be a configuration where the two robots are on the same line L . Let R be a set of nodes delimiting a rectangle for which L is an axis of symmetry. Let $R_1 \subset R$ such that the union of R_1 and the symmetric of R_1 , with respect to L , is equal to R . Then, starting from C , there exists an execution where a robot reaches a node in R_1 .*

Proof. Starting from C , the robots have to explore the nodes outside the rectangle, so there exists an execution E where a robot reaches a node u in R at time t . If $u \in R_1$, the Lemma is proved. Otherwise $u \in R_2$ and there exists a node $u' \in R_2$, symmetric to u with respect to L . If $u' \in L$ then $u = u'$, so $u \in R_1$ like in the first case. Otherwise, a robot moved away from L at some round before t . Let t_0 be the first time, after configuration C , such that a robot, say r , moved away from L . In $C(t_0)$ the view of both robot was symmetric with respect to L so that the rule executed by r is ambiguous. Now we consider the execution E' similar to E , but where at time t_0 the adversary mirrored the view of r , so that the robot r move away from L in the other side of it. After that, we consider

that the execution is similar to E but where everything is mirrored with respect to L . Then, a robot reaches the point u' at time t and the Lemma is proved. $\square$

B Correctness proof of Vone_3^3

We first show some properties of Vone_3^3 .

Property 1. Let l be a line of the grid that hosts exactly two adjacent robots with respectively color F (follower) and color L (leader). Let $U = u_1, u_2, \dots, u_m$ be the sequence of nodes of l on the empty path from a leader to a wall. By executing Vone_3^3 , eventually the leader (respectively the follower) is located on u_m (respectively u_{m-1}) and every node of U is explored.

Proof. Can be shown easily by induction.

Property 2. Let l be a line of the grid which is not adjacent to a wall and which hosts exactly two adjacent robots with respectively color F (follower) and color L (leader). Let $\leftarrow$ be the direction defined from the follower to the leader, which we refer to the line exploring direction. Assume that the leader is adjacent to a wall and both the leader and the follower do not observe any other robot. By executing Vone_3^3 , after three rounds, the robots are on l and the line exploring direction is switched by an angle of 2π .

Proof. As the leader is adjacent to a wall and does not observe any other robot, by the rules of Vone_3^3 , the leader moves to its adjacent empty node changing its color to X . Note that we are sure of the existence of such a node since $k \geq 3$. In case of symmetry, the scheduler chooses the direction to take. In the next round the follower reaches also the wall and observes the landmark (the previous leader). By the rules of Vone_3^3 , the follower moves back to its previous position while the landmark follows the follower. From here, the landmark is adjacent to a wall. Then, the follower moves away from the landmark changing its role to a leader while the landmark follows the leader changing its role to a follower. Hence, the lemma holds.

Property 3. Let l_i, l_{i+1}, l_{i+2} be a sequence of three lines of the grid in a given direction such that none of them is adjacent to a wall. Assume that l_i hosts two adjacent robots with color F and L respectively and that l_{i+1} hosts a single robot with color L . Assume that both leaders are adjacent to the same wall. By executing Vone_3^3 , after three rounds, l_i is empty, l_{i+1} hosts a single robot with color L adjacent to a wall and l_{i+2} hosts two adjacent robots with respectively color F and L being at distance 2 and 3 respectively from a wall.

Proof. As both leaders are adjacent to the same wall and are located in two adjacent lines, they observe each others. By the rules of Vone_3^3 , the leader that does not observe a follower remains idle but updates its colors to X while the other leader (that observes the follower) changes moves to its adjacent empty node changing its color to F . After one round the three robots are on the same

line and are all adjacent to a wall. By the rules of Vone_3^3 , the follower that has only one adjacent occupied node, leaves the wall and updates its role to leader. The second follower, moves toward the adjacent follower updating its role to the landmark and lastly, the landmark moves toward the adjacent follower keeping the same role. In the next round the leader moves away from the landmark keeping its role while its adjacent landmark moves toward the leader becoming a follower. The landmark which does not observe a leader changes its role to become a leader. Hence, the lemma holds.

Given an orientation of the grid, let $l_1, l_2, \dots, l_k$ be the sequence of lines with respect to the chosen orientation. Assume without loss of generality that there exists $i \in \{2, \dots, k-1\}$ such that l_i hosts a single robot and l_{i+1} hosts two robots. We refer to the direction from l_i to l_{i+1} by the *exploring direction*.

Property 4. Let l_i, l_{i+1}, l_{i+2} be a sequence of three lines of the grid in a given direction such that l_{i+2} is adjacent to a wall. Assume that l_i hosts two adjacent robots with color F and L respectively and that l_{i+1} hosts a single robot with color L . Assume that both leaders are adjacent to the same wall. By executing Vone_3^3 , eventually the corner of l_{i+2} which is adjacent to an occupied node is eventually occupied and the exploring direction is change by an angle of π .

Proof. As both leaders are adjacent to the same wall and are located in two adjacent lines, they observe each others. By the rules of Vone_3^3 , the leader that does not observe a follower remains idle but updates its colors to X while the other leader (that observes the follower) changes moves to its adjacent empty node changing its color to F . After one round the three robots are on the same line and are all adjacent to a wall. However, this time one of the follower is adjacent to two wall. That is, the robot knows that they have reached the last line of the grid with respect to the current exploring direction. By the rules of Vone_3^3 , this follower moves to its unique adjacent node keeping its color. The second follower and the initial landmark both move along the wall. Observe that the follower updates its state to X . That is, in the configuration reached, l_{i+2} hosts two adjacent robots; a follower and a landmark, the landmark being adjacent to a wall while l_{i+1} hosts a unique landmark which is also adjacent to a wall. By the rules of Vone_3^3 , the follower moves in the opposite direction of the wall reaching line l_{i+1} . At the same time both landmarks remain idle but change their color to L . As the follower becomes adjacent to a leader, the follower moves towards the adjacent leader and the leader that observes the follower moves along the wall changing its color to F . That is, in the configuration reached l_i and l_{i+1} host a follower and l_{i+2} hosts a leader. The three robots are adjacent to same wall. By the rule of Vone_3^3 , the two followers move in the opposite direction of the wall. Note that the follower having a unique adjacent occupied node changes its color to L . Simultaneously, the leader that is at the corner of the grid remains idle but changes its color to X . By Properties 1, 2, the leader and the follower explore line ℓ_k and return. When the leader reaches the wall, it observes a landmark and moves in the opposite direction changing its color to F . After one round we can deduce that the lemma holds.

Theorem 2. Vone_3^3 solves the perpetual exploration problem with three robots, having three colors and visibility range one.

Proof. By induction on $n \times m$, where n is the number of rows and m is the number of columns in the grid. We have validated several base cases using our simulation tool [17]. Precisely, we have checked that Vone_3^3 solves the problem on every grid of size $n \times m$ for $n, m \in \{3, 4\}$.

We assume now that Vone_3^3 solves the problem in all grids $x \times y$ with $3 \leq x \leq n$ and $3 \leq y \leq m$ for some values $n, m \geq 3$. We should show that Vone_3^3 solves the perpetual finite grid exploration problem in the grids of size $(n+1) \times m$ and $n \times (m+1)$.

Let us focus on the grid of size $n \times (m+1)$ (the case $(n+1) \times m$ can be proved similarly). First assume that the current exploring direction is vertical *i.e.*, the robots are exploring the grid row by row. One can see that every node, except the topmost and bottommost rows, are explored. Indeed, by Properties 1 and 2, the robots cross the row from one side to another in both directions as they move in a straight line until they reach a wall.

On another hand, by Property 3, when robots cross back the row and reach a wall and observe another robot on a adjacent row, they leave their current row to move to the next one with respect to the current exploring direction. As the grid is finite, by Property 4, one of the corner of the grid is explored and the exploring direction is switched by an angle of π . That is, the robots start exploring the grid horizontally, column by column, repeating the same process. That is, after three changes of the exploring direction by an angle of π , every corner is visited. Hence, every node of the added column is explored.

Next, assume that the current exploring direction is horizontal *i.e.*, the robots are exploring the grid column by column. By Properties 1, 2, 3 and 4, we know that the exploring direction eventually changes by an angle of π . We retrieve, the previous discussed case.

C Rules of Vtwo_1^5

We present here all the rules of algorithm Vtwo_1^5 and when they are used.

Fig. 20: Rules used by robots to move in a straight line

Fig. 21: Rules used by robots when start from a d configuration

Fig. 22: Rules used by robots to make a turn

Fig. 23: Rules used by robots to follow a wall

Fig. 24: Rules used by robots at the corner

Fig. 25: Rules used by robots when starting from the bottom right corner