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

Arthur Rauch[1], Quentin Bramas[1][0000−0003−0612−5616], Stéphane Devismes[2], Pascal Lafourcade[3][0000−0002−4459−511X], and Anissa Lamani[1][0000−0001−7774−8402]

[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 neither global compass nor a common chirality. In this context, we propose optimal solutions for the perpetual exploration of a finite grid. Precisely, we investigate optimality in terms of the visibility range, number of robots, number of colors. 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 five robots and only one color, *i.e.*, robots are oblivious.

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

1 Introduction

We consider swarms of *luminous robots* [16], *i.e.*, autonomous robots endowed with visibility sensors, motion actuators, and lights of different colors. Those robots operate in *synchronous* Look-Compute-Move cycles, where they first sense the environment (Look), then choose a destination and update their light color (Compute), and finally move to the chosen destination (Move).

Our goal is to investigate the computational power of such robot swarms. Hence, we consider luminous robots with limited capabilities. First, they are *myopic*, *i.e.*, they are only able to sense their surroundings within a bounded visibility range. Furthermore, they are fully disoriented since they have neither a *global compass* nor a *common chirality*. Finally, except from their lights, robots have neither persistent memories nor communication capabilities.

* 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 infinite global task called the *perpetual exploration*. Given a space which is partitioned into locations, it requires each of these locations to be visited infinitely often by at least one robot. Here, we conveniently discretize the space by a finite graph, where nodes represent locations and edges represent the possibility for a robot to move from one location to another.

In this paper, we look for optimal *exclusive* solutions to the perpetual exploration of a *finite grid*, both in terms of visibility range and 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 problem that has been extensively 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]. In particular, it is shown in [4] that, without a common chirality, exploring an infinite grid with oblivious[4] synchronous robots is impossible under visibility range one, whatever be the number of robots. This latter result is established by proving that, under these settings, robots are not able to move from an arbitrary distance. Hence, it also applies to grid of unbounded size.

In finite graphs, many papers [12,13,14,10,7,9,8] consider the terminating version of the exploration (henceforth called *terminating* exploration), which requires that all robots eventually stop moving after all nodes have been visited. The perpetual exploration requires each location to be visited infinitely often by all or a part of robots. Perpetual exploration of finite graphs has been, for example, considered in [2,3,6].

A large part of the literature deals with "*non-myopic*" robots, *i.e.*, robots with an unbounded visibility range allowing them to sense the whole graph and the positions of all the robots; see [12,2,3,13,14,10,9,8]. In such a context, robots are always assumed to be anonymous and oblivious. Exploration algorithms satisfying exclusiveness are proposed in both finite [2,3,6] 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. Chirality is important when dealing with optimal solutions. 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] studies the exploration problem in finite grid, with myopic, synchronous, and luminous robot (like our model here), yet assuming robots agree on a common chirality. In a nutshell, it is shown in [6] that two robots with three colors and a common chirality are necessary and sufficient to solve the problem under visibility range one. Moreover, under visibility range two and assuming a common chirality, three robots are necessary and sufficient when robots have only one color.

[4] Oblivious robots have no state and cannot remember past actions.

Visibility	# Robots	# Colors	Algorithm
1	2	finite	Impossible (Thm. 1)
1	**3**	3	Vone_3^3
2	5	**1**	Vtwo_1^5

Table 1: Summary of our results.

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

Our contribution is threefold. We prove that, under any finite visibility range, the perpetual exploration is not solvable using only two robots, whatever be the finite number of available colors. Then, we present a perpetual exploration algorithm that is optimal in terms of visibility range (1) and number of robots (3). Moreover, this algorithm only requires 3 colors per robots. Finally, we propose an algorithm that requires five oblivious robots, *i.e.*, each of those five robots needs only one color (the optimal), yet assuming visibility range two. Nevertheless, following results in [4], visibility range two is the smallest range where a solution with oblivious robots is possible. Table 1 summarizes our contributions.

Roadmap. Section 2 is devoted to the computational model and definitions. In Section 3, we prove our impossibility result. We present our algorithm in Sections 4 and 5. We make concluding remarks in Section 6.

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,[5] *i.e.*, robots evolve in an 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 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. 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.

[5] 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 counterclockwise), 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 $Views \to \{Idle, Up, Left, Down, Right\} \times Cl$, 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 $f^{-1}(T(f(V_C(i,j)))) = (Idle, c)$,
- $((i-1,j),c') \in C$ 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 $f \in \mathcal{IT}$. 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 two 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.

Notice also that, due to the absence of orientation and chirality, a rule (V, d, c) may be ambiguous, meaning that there exists $f \in \mathcal{IT}$ such that $T(V) \neq f^{-1}(T(f(V)))$. In the figures, we illustrate such ambiguities by depicting the possible destinations with several arrows. For example, Figure 2 shows an ambiguous rule where the robot has a symmetric view. Hence depending on the transformation f chosen by the adversary, the robot moves either left or right when executing this rule.

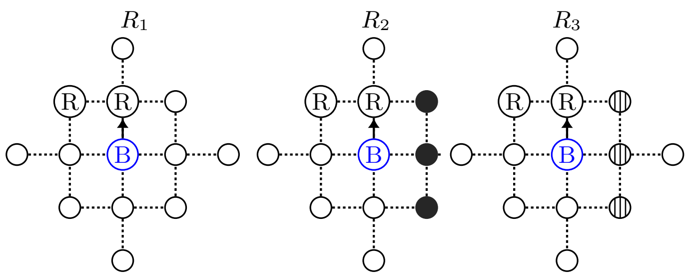

Fig. 1: Examples of rules.

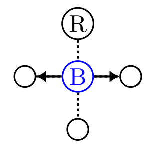

Fig. 2: Example of an ambiguous rule.

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 one and the same pattern which fixes the colors and relative positions of the robots. Hence, for a given grid, every two possible initial configurations are equal up to a translation applied to all robot positions and 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 now extend this result by proving that the PFGE problem is also not solvable using two robots if they have a visibility range of one. Hence, throughout this section, **we assume two robots under visibility range one.**

First, we observe that in large enough grids, if robots travel a long distance without seeing any wall, or seeing one and the same wall without reaching its corner, then they must execute a periodic sequence of movements. Indeed, the maximum number of distinct relative positions and colors two robots endowed with $|Cl|$ colors can have is the number of 2-combination with repetitions $B = \left(\!\binom{|Cl|}{2}\!\right) = \frac{|Cl|(|Cl|+1)}{2}$. Thus, if robots travel a distance at least B without seeing a wall, or seeing one and the same wall without reaching its corner, then they are actually executing a periodic sequence of movements. Of course, the value of B depends on the algorithm, yet it is always finite. Notice also that $|Cl| > 1$, since it has been shown in [6] that two oblivious robots with visibility range 1 are not sufficient to solve the PFGE problem. Hence, $B \geq 3$.

The above observations are important to prove our impossibility results. First, we use them to show that once robots move far away from the wall, their movements are restricted. In more detail, they can only move in straight line; see Lemmas 1 and 2.

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

Lemma 2. *Let* A *be an algorithm solving the PFGE problem with two robots. If there exists an execution of* A *containing a configuration* C *where robots are at distance at least* $2B$ *from any wall and, from* C*, robots perform a periodic sequence of movements, then this sequence does not include any ambiguous rule.*

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 the whole grid. □

Due to the limited visibility range, the two robots cannot be to far from each other, as stated in the following three lemmas.

Lemma 3. *Robots are always at distance at most 6 for each other.*

Lemma 4. *No exploration algorithm can reach a configuration where the two robots are at distance at least 3, one robot sees no wall, and the other sees zero or one wall.*

Lemma 5. *If both robots see no wall, then they should be at distance one from each other.*

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 on which side of the axis it will keep the robots.

Lemma 6. *Let* A *be an algorithm solving the PFGE problem using 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, from* C*, a configuration where a robot is located at a node in* R_1 *is reachable.*

We now prove our main lemma, which states that the robots cannot move further than a distance of $4B$ from all walls. To achieve this, we need two additional 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 are both at distance at most $3B$ from another wall.

Lemma 7. *If* A *solves the PFGE problem with two robots, then, if at a given time* $t > B$*, a robot is in a corner box or if robots are in a* T*-configuration, then there exists an execution after* C *such that a robot 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* $4B$ *from a wall.*

Proof. We consider a grid of size greater than $4B$, otherwise the lemma is proven regardless of what the robots are doing (a robot is infinitely often in a corner box and any wall at distance $4B$).

Then, assume a robots is 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) at a given time $t > B$. To explore the grid, the robots must leave the corner box. Indeed, if a robot stays forever in a corner box, then both robots remain as distance at most $2B + 6$ (by Lemma 3) from that corner and, since, $B \geq 3$, $2B + 6 < 4B$ meaning that some node are only finitely often visited. We denote by W_1 and W_2 the two walls adjacent to the corner contained in the corner box where a robot was located in C; see Figure 3. Without the loss of generality, we assume that at a given time t_0, the last robot, say r, leaving the corner box of size $2B$ is at distance $2B + 1$ from W_1, and so at distance at most $2B$ from W_2.

Claim 1: *After leaving the corner box from a given side, either* (i) *the robots move until reaching the wall opposite to* W_1*, in a* T*-configuration, while remaining at distance* $2B$ *from Wall* W_2*, or* (ii) *end up in a line* L *parallel and at distance at most* $4B$ *to* W_1*, while remaining at distance at most* $2B+1$ *from Wall* W_2*.*

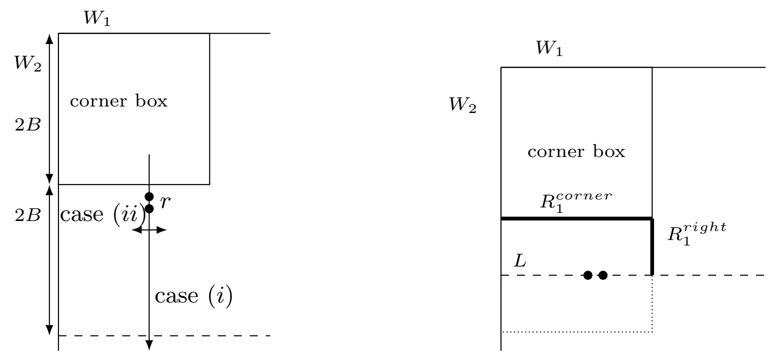

Fig. 3: The different cases in Lemma 7.

From the previous claim, we saw that two cases can occur; see Figure 3. In the first case, the lemma is proven.

In the second case, robots end up in a line L parallel to W_1 at time a given $t_1 \geq t_0$, while remaining at distance at most $2B + 1 < 4B$ from wall W_2. 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 $[0, 2B+1]$, and R_1^{right} is the segment of nodes at distance $2B+1$ 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 (from the previous Claim, $d_1 \leq 4B$). The union of R_1 with its symmetric with respect to L delimits a rectangle (dotted line in the figure) so that, using Lemma 6, 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 proven. 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 (by Lemma 2) 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 $4B$ from W_1.

We now consider the case where 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. □

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 PFGE problem and consider a grid of size $9B \times 9B$. Since the robots should explore the entire grid, a robot is eventually in a corner box. Using Lemma 7 repeatedly, we can construct a execution from there where the two robots remain at distance at most $4B$ from any wall. Hence,

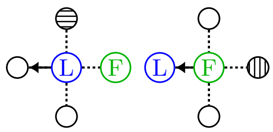

Fig. 4: Move in a straight line.

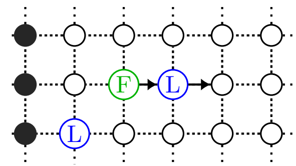

Fig. 5: Beginning of the exploration of a line.

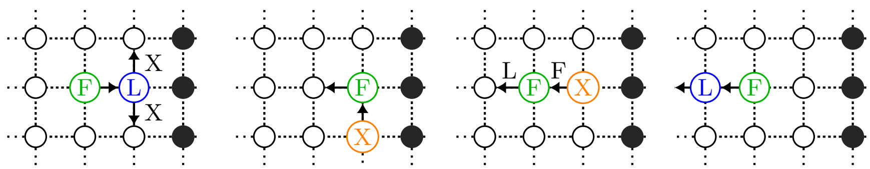

Fig. 6: Sequence of configurations during a turn around.

nodes at distance more than $4B$ from all the walls are not visited anymore, a contradiction. □

4 Visibility range one: Vone_3^3

In this section, we present an algorithm, denoted by Vone_3^3, which assumes visibility range one (the optimal) and uses three robots endowed with three colors. By Theorem 1, Vone_3^3 is optimal in terms of number of robots. We encourage the reader to take a look at the online animation illustrating the behavior of Vone_3^3 [17] while reading the following explanation.

The algorithm defines three roles for the robots using the colors: L (*leader*), F (*follower*), X (*landmark*). The roles are not fixed, robots will alternate between several roles along the execution. Moreover, in few particular situations, roles will not exactly correspond to their default meanings.

Initially, the three robots are aligned, two of them have color L while the third one has color F; moreover the two robots with color L are adjacent. In the following, we denote this pattern by LLR. Since initial configurations are locally-defined, the possible initial configurations are then all those containing the pattern LLR.

Since we assume the synchronous model and we consider the perpetual exploration, the execution is necessarily eventually periodic. So, from an initial configuration, the goal is to lead robots to a configuration C_p from which they will start to perform periodic journeys around the grid. We first explain how periodic journeys are built. Then, we will see how robots can easily reach a configuration of the journey starting from any initial configuration.

The main idea of the algorithm is to make the leader and the follower move and explore a given line while the landmark robot remains idle to keep track of

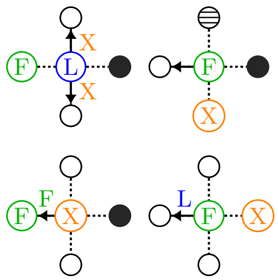

Fig. 7: Turn around.

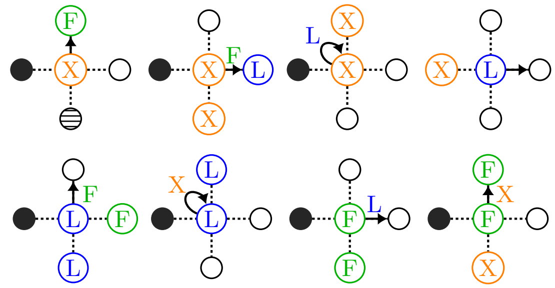

Fig. 8: Up and Turn.

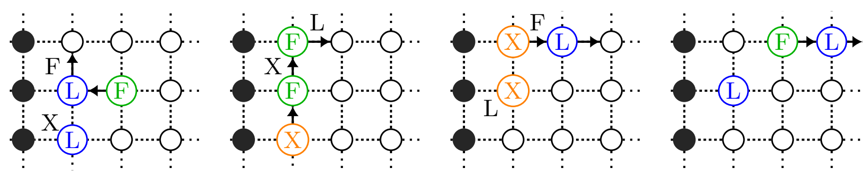

Fig. 9: Up and turn. This sequence occurs when the robots are not in a corner. The case when the robots are in a corner is presented in Figure 11.

the next line to explore. Every time a line is explored, the three robots, including the landmark robot, move “up” by one row (assuming, for illustration purpose, 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.

It is easy to make move the leader and the follower on the same direction to explore a line: the leader moves away from the follower while the follower, as suggested by its name, follows the leader. The rules executed by those two robots to move along a straight line are presented in Figure 4.

During the line exploration by the leader and the follower, the landmark robot is left beside a wall on a line adjacent to the line traversed by the two other robots; refer to Figure 5. When the leader and the follower reach the other wall, the idea is to make them move back and cross the same line again since they do not have any sense of direction. For this purpose, they need to swap their respective positions. This is done as follows: the first robot that detects the wall is the leader, in this case, it moves to an adjacent empty node (except for the last line, there is a symmetry and so the scheduler chooses which direction to take) and changes its color to X. In the next round, the follower reaches the wall and observes the landmark, *i.e.*, the previous leader. Since the follower sees only one other robot, it detects that they are moving back to traverse the same line in the other direction. So, the follower moves back to its previous position followed by the landmark. Moreover, the follower becomes the leader while the landmark becomes a follower. Finally, they both start moving straight on the opposite direction. The rules executed during the moving back process are those

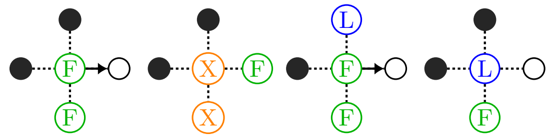

Fig. 10: Last corner preparation - Rules.

given in Figure 7 plus the first rule of Figure 9 (this latter will also be used when switching to an upper line).

When the leader and follower reach again the wall again, the leader can observe this time that a robot (the landmark robot) is located on a different line in the neighborhood of the wall. Hence, an orientation can be defined to indicate the next line to be explored *i.e.*, the line containing the only unoccupied node adjacent to the one hosting the leader. The idea is to make the robots move to the next line in such a way they can repeat the previous behavior. The lines of the grid are then explored in a given direction one by one until robots reach the last line. The rules that are executed to make a line change, when the landmark robot is reached, are presented in Figure 8. Figure 9 shows the sequence of configurations occurring during a line change.

Given an orientation of the grid, assume without the 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 reach the last line 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: they 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 (the leader) 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 composing this process are presented in Figure 11.

Assume initially the robots are all adjacent to a wall (remember that they are aligned and their colors are respectively F, L and L). Then, we have defined few rules in order for the robot to reach, after one round, a configuration of the periodic journey. After that, robots behave exactly as previously explained. Starting from any other initial configuration, the goal is to move straight toward a wall. Once the leader robots see the wall, it moves to an unoccupied node and the reached configuration is exactly the same as the first one of an "up and turn" sequence. Hence after that, the periodic journeys start. The rules used by the robots to do this are shown in Figure 13.

For the grids with 3 lines or 3 columns, 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 movements when the grid has only 3 lines or 3 columns is shown in Figure 15 (the specific rule is used in the fifth round of the sequence).

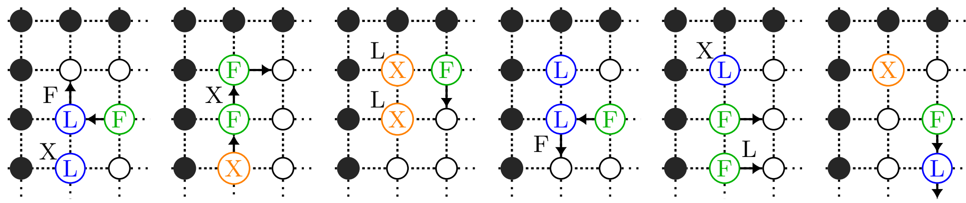

Fig. 11: Corner turn. After the sequence, the exploration continues as before, but everything is rotated by a clockwise angle of π.

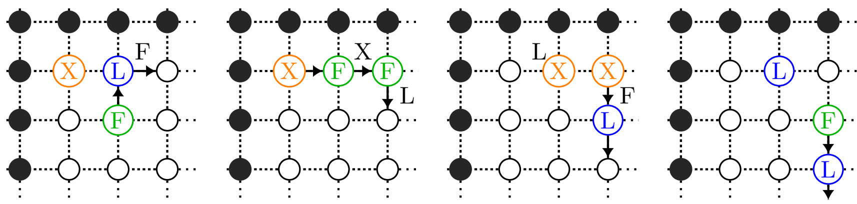

Fig. 12: First “up and turn” after the corner turn.

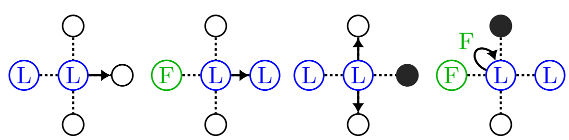

Fig. 13: Rules used by the robots to reach the wall starting from a configuration where they shape an LLF pattern.

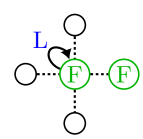

Fig. 14: Rules used by robots to handle grid with only 3 columns.

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 were able to prove that our algorithm is correct for any grid $n \times m$, with $m, n \in \{3, 4\}$. Then, we have shown that when a group of robots 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. □

5 Visibility range two : Vtwo_1^5

We now outline our second algorithm, Algorithm Vtwo_1^5, which requires five oblivious robots (*i.e.*, they all have the same fixed color) and visibility range of 2.

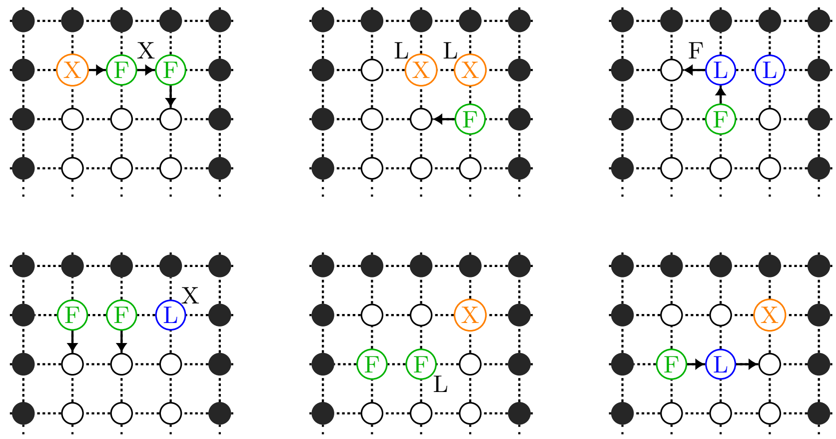

Fig. 15: Sequence of configurations when the grid has only 3 columns.

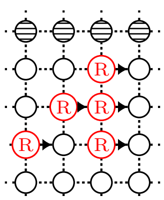

Fig. 17: Move in a straight line.

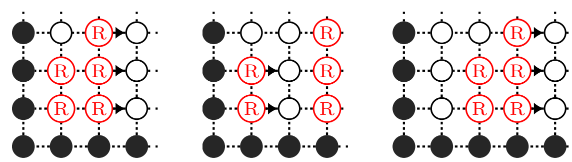

Fig. 18: Follow the wall.

Again, we encourage the reader to follow the explanation of the algorithm while looking at the animations available online [17].

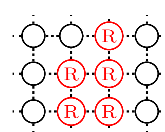

Fig. 16: Initial relative positions.

The initial relative position of robots in Vtwo_1^5 is given in Figure 16. Starting from an initial configuration, the principles of the algorithm are similar, yet simpler, than for the previous one. Indeed, the robots remain grouped together, and they move from left to right, without making any rotation when reaching a wall. 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 19 for a turn one row below). After moving straight (see Figure 17) to the opposite wall, everything is mirrored, so they do the same. They move back an forth until they reach the top wall. After following the top wall (using a specific periodic sequence of movements, see Figure 18), they make a special turn in order to move back and forth in the other direction.
The proof of the next theorem is similar to that of Theorem 2.

Theorem 3. Vtwo_1^5 *solves the perpetual exploration problem with five oblivious robots under visibility range two.*

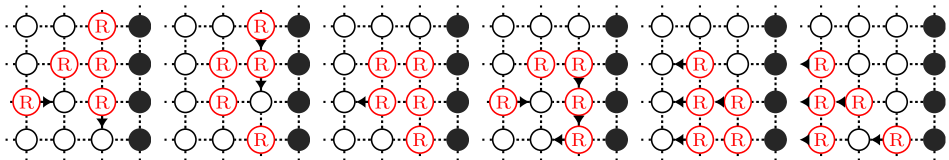

Fig. 19: Turn around.

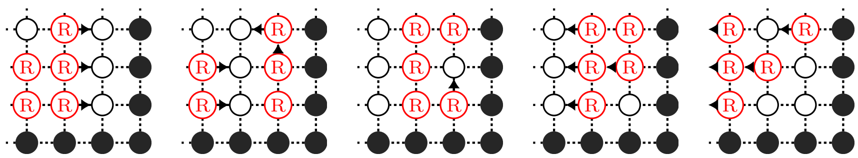

Fig. 20: Turn at a corner.

6 Conclusion

We have investigated the perpetual exclusive exploration of a finite grid by a swarm of myopic luminous synchronous robots that have neither a common sense of direction nor a common chirality. In these settings, We have proposed optimal solutions with respect to either the number of robots, the visibility range, and the number of colors.

In more detail, we have first shown that if robots have only a visibility range one, then the problem is not solvable with two robots, regardless of the number of colors. Then, we have proposed Vone_3^3 which uses three robots and three colors. This algorithm is optimal both in terms of visibility range and number of robots.

Next, under visibility range two, we gave Algorithm Vone_1^5. This latter requires five oblivious robots, *i.e.*, five robots that use the minimal number of color (one). Following the impossibility result of [4], visibility range two is the smallest range admitting a solution in our settings.

The immediate open questions related to this work are about determining 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. Finally, it would be interesting to extend our study to other topologies such as torus-shaped networks.

References

1. Baldoni, R., Bonnet, F., Milani, A., Raynal, M.: Anonymous graph exploration without collision by mobile robots. Inf. Process. Letters **109**(2), 98–103 (2008)
2. Blin, L., Milani, A., Potop-Butucaru, M., Tixeuil, S.: Exclusive perpetual ring exploration without chirality. In: Lynch, N.A., Shvartsman, A.A. (eds.) Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6343, pp. 312–327. Springer, Boston, Massachusetts, USA (September 2010)
3. Bonnet, F., Milani, A., Potop-Butucaru, M., Tixeuil, S.: Asynchronous exclusive perpetual grid exploration without sense of direction. In: Anta, A.F. (ed.) Proceedings of International Conference on Principles of Distributed Systems (OPODIS 2011). pp. 251–265. No. 7109 in Lecture Notes in Computer Science (LNCS), Springer Berlin / Heidelberg, Toulouse, France (December 2011), http://www.springerlink.com/content/9l3v424157681707/
4. Bramas, Q., Devismes, S., Lafourcade, P.: Finding Water on Poleless using Melomaniac Myopic Chameleon Robots. In: FUN 2020, 10th International Conference on Fun with Algorithms. LiPICs, Favignana, Sicily, Italy (September 28-30 2020), to appear
5. Bramas, Q., Devismes, S., Lafourcade, P.: Infinite Grid Exploration by Disoriented Robot. In: NETYS'2020, the 8th International Conference on NETworked sYStems. Springer, Marrakech, Morocco (June 3-5 2020), to appear
6. Bramas, Q., Devismes, S., Lafourcade, P.: Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Opaque Robots with Common Chirality. In: ICDCN'21: International Conference on Distributed Computing and Networking, Virtual Event. pp. 76–85. ACM, Nara, Japan (5-8 January 2021)
7. Datta, A.K., Lamani, A., Larmore, L.L., Petit, F.: Enabling ring exploration with myopic oblivious robots. In: 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IPDPS 2015. pp. 490–499. IEEE Computer Society, Hyderabad, India (May 25-29, 2015 2015)
8. Devismes, S., Lamani, A., Petit, F., Raymond, P., Tixeuil, S.: Terminating exploration of a grid by an optimal number of asynchronous oblivious robots. The Computer Journal **1** (03 2020), to appear
9. Devismes, S., Lamani, A., Petit, F., Tixeuil, S.: Optimal torus exploration by oblivious robots. Computing **101**(9), 1241–1264 (2019)
10. Devismes, S., Petit, F., Tixeuil, S.: Optimal probabilistic ring exploration by semi-synchronous oblivious robots. Theoretical Computer Science (TCS) **498**, 10–27 (2013)
11. Dieudonné, Y., Petit, F., Villain, V.: Leader election problem versus pattern formation problem. In: Lynch, N.A., Shvartsman, A.A. (eds.) Distributed Computing, 24th International Symposium, DISC 2010. Lecture Notes in Computer Science, vol. 6343, pp. 267–281. Springer, Cambridge, MA, USA (September 13-15 2010)
12. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Remembering without memory: Tree exploration by asynchronous oblivious robots. Theor. Comput. Sci. **411**(14-15), 1583–1598 (2010)
13. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: How many oblivious robots can explore a line. Inf. Process. Lett. **111**(20), 1027–1031 (2011)
14. Flocchini, P., Ilcinkas, D., Pelc, A., Santoro, N.: Computing without communicating: Ring exploration by asynchronous oblivious robots. Algorithmica **65**(3), 562–583 (2013)

15. Ooshita, F., Tixeuil, S.: Ring exploration with myopic luminous robots. In: Izumi, T., Kuznetsov, P. (eds.) Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018. Lecture Notes in Computer Science, vol. 11201, pp. 301–316. Springer, Tokyo, Japan (November 4-7 2018)
16. Peleg, D.: Distributed coordination algorithms for mobile robot swarms: New directions and challenges. In: Pal, A., Kshemkalyani, A.D., Kumar, R., Gupta, A. (eds.) Distributed Computing – IWDC 2005. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
17. Rauch, A., Bramas, Q., Devismes, S., Lafourcade, P., Lamani, A.: Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Robots without Chirality: The Animations (Mar 2021). https://doi.org/10.5281/zenodo.4640462, https://doi.org/10.5281/zenodo.4640462