# Highlights

**Infinite Grid Exploration with Synchronous Myopic Robots without Chirality**[⋆]

Quentin Bramas, Pascal Lafourcade, Stéphane Devismes

- The infinite grid exclusive exploration by synchronous oblivious robots is impossible under visibility range one whatever by their number.
- The infinite grid exclusive exploration can be achieved by eight synchronous oblivious robots under the optimal visibility range two.
- The infinite grid exclusive exploration can be achieved by six synchronous robots under visibility range one using only two colors.
- The infinite grid exclusive exploration can be achieved by only five synchronous robots under visibility range one, yet using twelve colors.

# Infinite Grid Exploration with Synchronous Myopic Robots without Chirality

Quentin Bramas[a,], Pascal Lafourcade[b], Stéphane Devismes[c]

[a] *University of Strasbourg, ICUBE, CNRS, Strasbourg, France*
[b] *LIMOS, University Clermont Auvergne, Aubière, France*
[c] *Université de Picardie Jules Verne, Amiens, France*

**Abstract**

In this paper, we consider the exploration of an infinite grid by a swarm of fully-synchronous robots with weak capabilities: they are disoriented, opaque, do not communicate explicitely, have limited visibility, and cannot occupy the same position at the same time.

Our first result shows that, in this context, minimizing the visibility range and the number of used colors are two orthogonal issues: it is impossible to design a solution to our exploration problem that is optimal *w.r.t.* both parameters simultaneously. Consequently, we address optimality of these two criteria separately by proposing two algorithms; the former being optimal in terms of visibility range, the latter being optimal in terms of number of used colors. More precisely, the first algorithm solves the problem using eight oblivious robots under visibility range two (this visibility being optimal when considering oblivious robots), and the second algorithm solves the problem under visibility range one using six robots and two colors (which is optimal under this visibility range). Finally, we also tackle the optimality in terms of number of robots. According to the lower bound given in [7], we propose an algorithm working with a minimum number of robots (five) under visibility range one. This latter uses twelve colors and also guarantees that nodes are visited infinitely often.

⋆This study was partially supported by the French ANR project SkyData (ANR-22-CE25-0008-01).

*Email addresses:* bramas@unistra.fr (Quentin Bramas), pascal.lafourcade@uca.fr (Pascal Lafourcade), stephane.devismes@u-picardie.fr (Stéphane Devismes)

*Keywords:* Robot swarms, Luminous Robots, Grid, Infinite Exploration, FSYNC model

## 1. Introduction

Robot swarms are a major field of distributed computing that aims to develop distributed algorithms allowing the coordination of autonomous mobile entities, called *robots*, in order to solve tasks global to the system. An important research axis of this field consists in identifying the weakest capabilities allowing robots to solve a given coordination problem. The motivation behind this fundamental question is quite natural, since the use of robots that are physically limited –in terms of sensors, actuators, memory, battery, and computing capabilities– helps to reduce their manufacturing and maintenance costs, and consequently enables their massive deployment.

In this paper, we tackle this issue by studying how a finite set of weak robots can solve an infinite task. More precisely, we consider a set of *fully-synchronous* robots evolving in an infinite grid, where nodes represent possible locations and edges represent the possibility for a robot to move from one location to another. Our goal is to make robots explore the infinite grid, *i.e.*, each node of the grid should be visited within finite time by at least one robot. In the following, we refer to this problem as the *Infinite Grid Exploration* (IGE) problem. We also impose the exploration to be *exclusive* (also called *collisionless* in the literature): robots can neither occupy the same node simultaneously nor traverse the same edge at the same time.

The robots we consider have very low capabilities. First, they are *disoriented* in the sense that they have no compass and do not agree on any common chirality. Moreover, they are *silent,* i.e.*, they have no direct communication mean. They are also* myopic *meaning that endowed by visibility sensors allowing them to perceive their environment within a given constant (typically small) distance, called the* visibility range. *Our robots are* luminous *[12, 22, 25],* i.e.*, endowed with a light that can take different colors and that can be perceived by other robots whenever it is within their visibility range. Finally, we assume that robots are* opaque *meaning that a robot cannot see what is behind another robot. In a nutshell, in this model, lights are both the only indirect communication mean and the only persistent memory storage. In the special case where lights have a just one color, the same for all robots, robots are said to be* oblivious.

*Our robots operate in the well-known Look-Compute-Move (LCM) model proposed by Suzuki and Yamashita [28, 29]. That is, they perform cycles that comprise three synchronous phases: Look, Compute, and Move. During the first phase (Look), robots take a snapshot of their environment using their visibility sensors. In the second phase (Compute), they decide, based on the previous snapshot, a destination in their surrounding and update their color. Finally, in the last phase (Move), they move to the computed destination, if the destination is different from their current location.*

### *1.1. Contribution*

*We look for solutions to the IGE problem that are optimal* w.r.t. *three parameters: the number of robots, the visibility range, and the number of colors used by the robot lights.*

*We begin with an impossibility result: there is no algorithm solving the infinite grid exploration problem with oblivious fully-synchronous robots under visibility range one, whatever be their number.*

*A natural question is then: is-it possible to design an oblivious solution by slightly increasing the visibility range? We positively answer to this question by proposing a solution that uses eight oblivious fully-synchronous robots under visibility range two, which is then the optimal range in case of obliviousness.*

*Another orthogonal question is to ask whether there is a solution that uses only a few colors under visibility range one. We also positively answer to that question by proposing a solution with six robots that use only two colors, which is also optimal.*

*Finally, we investigate the optimal number of robots allowing to solve the IGE problem under visibility range one. Based on a lower bound given in [7], we propose an algorithm showing that five fully-synchronous robots under visibility range one are necessary and sufficient to solve the IGE problem. This latter solution requires twelve colors. Also notice that it actually solves a stronger specification as it is the first* perpetual *IGE algorithm. In this latter problem, every node of the grid is visited infinitely often.*

### *1.2. Related Work*

*Since their introduction [12, 22, 25], luminous robots have been widely studied. For example, they have been used to solve various dedicated problems including* weak gathering *[22] and* mutual visibility *[23]. In a more general framework, Das* et al. *[12] compare the computational power of luminous*

*robots with respect to the three main execution models: fully-synchronous, semi-synchronous, and asynchronous.*

*The exploration problem is one of the benchmark tasks when it comes to robots evolving on graphs. Exploration tasks have been first considered in the context of finite graphs. Various topologies have been studied: lines [20], rings [2, 13, 15, 21, 24], tori [14], grids [3, 16], cuboids [6], trees [19]. In this context, two main variants, respectively called the* terminating *and* perpetual *exploration, have been considered. The terminating exploration requires every possible location to be eventually visited by at least one robot, with the additional constraint that all robots stop moving after task completion. In contrast, the perpetual exploration requires each location to be visited infinitely often by all or a part of robots. In [16], authors solve terminating exploration of any finite grid using few asynchronous anonymous oblivious robots, yet assuming unlimited visibility range. The exclusive perpetual exploration of a finite grid is considered in the same model in [3]. Synchronous solutions to exclusive perpetual exploration of a finite grid by myopic luminous robots are investigated in [9, 26]. The asynchronous case is studied in [5].*

*Various terminating problems have been investigated in infinite grids such as* arbitrary pattern formation *[4],* mutual visibility *[1, 11], and* gathering *[27, 17].*

*A more related problem is that of the treasure search where robots should traverse an infinite environment to find a treasure left at an unknown position. Emek* et al. *[18] have investigated the treasure search problem in an unbounded size grid [10]. They consider robots operating in two models: the semi-synchronous and fully-synchronous ones. However, they do not impose the exclusivity at all since their robots can only sense the states of the robots located at the same node (in that sense, the visibility range is zero). Moreover, in contrast with our work, they assume all robots agree on a* global compass, i.e., *they all agree on the same directions North-South and East-West. They propose two algorithms that respectively need three fully-synchronous and four semi-synchronous robots. Moreover, they exclude solutions for two robots. In a followup paper [10], Brandt* et al. *extend the impossibility result of Emek* et al. *by showing the impossibility of exploring an infinite grid with three semi-synchronous deterministic robots that agree on a* global compass.

*In [7], we have investigated the IGE problem by a swarm of fully-synchronous luminous myopic robots. Those robots agree on a common chirality, but have no global compass, while here we neither assumed a common chirality, nor*

*a global compass. Precisely, we show that using only three fixed colors, six robots, with a visibility range one, are necessary and sufficient to solve the IGE problem. We also show that using modifiable colors with only five states, five such robots, under a visibility range one, are necessary and sufficient to solve the IGE problem. Finally, assuming visibility range two, we provide an algorithm that solves the IGE problem using only seven oblivious robots.*

### *1.3. Roadmap*

*Section 2 is devoted to the computational model and definitions. We propose our impossibility result in Section 3. Then, each of our three algorithms are presented in a dedicated section (Sections 4-6). Finally, we conclude with some research directions in Section 7.*

## *2. Preliminaries*

*We consider a swarm $\mathcal{R}$ of $n > 0$ robots located on (nodes of) an* infinite grid *with vertex set in $\mathbb{Z} \times \mathbb{Z}$,* i.e.*, there is an edge between two nodes $(i, j)$ and $(k, l)$ if and only if the* Manhattan distance *between those two nodes,* i.e.*, $|i - k| + |j - l|$, is one. The coordinates are used for the analysis only,* i.e.*, robots cannot access them.*

*Robots are* luminous*: each robot is endowed with a light that may take with different colors and that can be see by robots in the surrounding. We denote by Cl the set of all possible colors.*

*We consider the* fully synchronous model *(FSYNC),* i.e.*, we assume time is discretized into an infinite sequence of rounds 1,2, ... At each round, robots simultaneously perform a* Look-Compute-Move *cycle. In the* Look *phase, a robot gets a snapshot of the subgraph induced by the nodes within distance $\Phi \in \mathbb{N}^*$ from its position; $\Phi$ is called the* visibility range*. Notice that 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, based only on its last snapshot and its own color, each robot* computes *a destination (either Up, Left, Down, Right or, Idle) and may change its color. Finally, it* moves *towards its computed destination.*

*We forbid robots to simultaneously occupy the same node nor traverse the same edge (*exclusiveness*). In such a context, a node is* occupied *when a robot is located at the node, otherwise it is* empty*. The* state *of a node is then either the color of the robot located at this node, if it is occupied, or*

$\perp$ *otherwise. A snapshot includes the state of each node within distance* $\Phi$ *from the robot, except for those that are obstructed by the presence of another robots, indeed robots are* opaque.

*2.1. Configurations*

*A* configuration $C$ *is a set of pairs* $(p, c)$ *where* $p \in \mathbb{Z} \times \mathbb{Z}$ *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. Also, by a slight abuse of notation, we sometimes partition the configuration into several subsets* $C_1, \ldots, C_k$ *and write* $C = \{C_1, \ldots, C_k\}$ *instead of writing* $(C = C_1 \cup \ldots \cup C_k) \wedge (\forall i \neq j, C_i \cap C_j = \emptyset)$*.*

*2.2. Views*

*We denote by* $G_r$ *the* globally oriented view *centered at robot* $r$*,* i.e.*, the subset of the configuration containing the states of the nodes at distance at most* $\Phi$ *from* $r$*, translated so that the coordinates of* $r$ *is* $(0, 0)$*. We use this globally oriented view in our analysis to describe the movements of the robots: when we say "the robot moves Up", it is according to the globally oriented view. However, since robots not have any compass, they have no access to the globally oriented view. When a robot looks at its surroundings, it obtains a* local view*. To model the absence of compass, we assume that any* local view *acquired by a robot* $r$ *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),* $\pi/2$*,* $\pi$ *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.*

*Moreover, since robots may obstruct visibility, the function that removes the state of a node* $u$ *if there is another robot on the segment linking* $u$ *and* $r$ *(when coordinates are embedded in* $\mathbb{R} \times \mathbb{R}$*) is* systematically *applied to obtain the local view. Here, we assume that robots are* self-inconsistent*, meaning that different transformations may be applied at different rounds; the choice being made by an adversary.*

*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 $\pi$-rotation), then the robot moves $Down = f^{-1}(Up)$. In a configuration $C$, $V_C(i,j)$ denotes the globally oriented view of a robot located at $(i,j)$.*

*2.3. Algorithm*

*An algorithm $\mathcal{A}$ is a tuple $(Cl, I, T)$ where $Cl$ is the set of possible colors, $I$ is the initial configuration, 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 $C$, the 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 $\mathcal{A}$ is then a sequence $(C_i)_{i\in\mathbb{N}}$ of configurations such that $C_0 = I$ and $\forall i \geq 0$, $C_i \mapsto C_{i+1}$.*

*2.4. Exploration*

*An algorithm $\mathcal{A}$ solves the* infinite grid exploration (IGE) *if for every execution $(C_i)_{i\in\mathbb{N}}$ of $\mathcal{A}$ and every node $(i,j) \in \mathbb{Z}\times\mathbb{Z}$ of the grid, there exists $t \in \mathbb{N}$ such that $(i,j)$ is occupied in $C_t$.*

*A stronger form of IGE is the* perpetual IGE *where each node is visited infinitely often. More formally, an algorithm $\mathcal{A}$ solves the* perpetual IGE *if for every execution $(C_i)_{i\in\mathbb{N}}$ of $\mathcal{A}$, every node $(i,j) \in \mathbb{Z}\times\mathbb{Z}$ of the grid, and every $t \in \mathbb{N}$, there exists $t' \geq t$ such that $(i,j)$ is occupied in $C_{t'}$.*

*2.5. 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* $(Cl, I, T)$ *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* $f \in \mathcal{IT}$.

*As an illustrative example, consider local views given in Figure 1. A rule* $R$ *can associate View* $V_1$ *with the direction* Up*. Since* Up *is relative to the view, it means for the robot "I move towards the only robot I see". View* $V_1'$ *is obtained by rotation from* $V_1$*, so a robot cannot distinguish* $V_1$ *and* $V_1'$*, so the same rule* $R$ *applies in* $V_1'$ *and the robot moves Left towards the only robot it sees. However, if in* $V_1$ *a robot decides to move to the right towards an empty node, then, since it does not distinguish its right from its left, the actual destination between left and right will be decided according to the applied indistinguishable transformation* $f \in \mathcal{IT}$*. Similarly, Views* $V_2$ *and* $V_2'$ *are indistinguishable for the robots (one is the mirror-rotation of the other), so any rule that applies to* $V_2$ *also applies to* $V_2'$*, and conversely. For example, if a robot decides to move towards its blue neighbor* $B$ *in* $V_2$*, it will also move towards its blue neighbor in* $V_2'$.

*2.6. Well-defined Algorithms*

*Recall that robots are assumed to be self-inconsistent. In this context, we say that an algorithm* $(Cl, I, T)$ *is* well-defined *if the global destination computed by a robot does not depend on the applied indistinguishable transformation* $f$*,* i.e.*, for every globally oriented view* $V$*, and every transformation* $f \in \mathcal{IT}$*, we have* $T(V) = f^{-1}(T(f(V)))$*. Every algorithm we will propose, except the last one, will be well-defined. However, to be as general as possible, we will consider both well-defined and not well-defined algorithms in our impossibility results. Finally, let us remark that a well-defined algorithm has a unique execution.*

*We denote by* $\vec{t}_{(i,j)}(C)$ *the translation of the configuration* $C$ *by the vector* $(i, j)$*,* i.e.*, the configuration obtained by translating the coordinates of all occupied node positions in* $C$ *by* $(i, j)$.

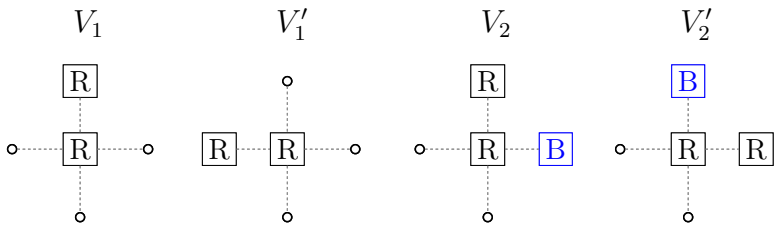


Figure 1: Example of four views. $V_1$ and $V_1'$ are indistinguishable. Similarly, $V_2$ and $V_2'$ are indistinguishable.

## *3. Impossibility Result*

*We start by proving the "fence crossing" Lemma, one of the key points of our impossibility result.*

### *3.1. The Fence Crossing Lemma*

*To explore an infinite grid, robots have to regularly cross what we call* fences. *A* fence $L$ *is composed of two infinite adjacent vertical lines* $L = (l_1, l_2)$ *with* $l_1 = \{(i_L, j) | j \in \mathbb{Z}\}$ *and* $l_2 = \{(i_L + 1, j) | j \in \mathbb{Z}\}$*, for some* $i_L \in \mathbb{Z}$*, such that each robot is initially located at some coordinates* $(x_0, y_0)$ *satisfying* $x_0 < i_L$*; see Figure 2. Informally, this means that a fence is made of two infinite adjacent vertical lines that are initially at the right of all robot's positions.*

*We say that a set of robots have crossed a fence when they are all at the right of the fence at a given time; see Figure 3. Notice that this does not mean that the robots always stay on the right of the fence afterward.*

*Formally, we say that a set of robots* $S$ has crossed the fence $L = (l_1, l_2)$ at Round $t$ *if there exists* $t' \leq t$ *such that every robot* $r \in S$ *is located at some coordinates* $(x_1, y_1)$ *with* $x_1 > i_L + 1$ *at Round* $t'$*.*

*We say a set of robots* $S$ single-handed crosses the fence $L$ between $t$ and $t'$ *if for every robot* $r \in S$*,*

1. *$r$ is located at some coordinates $(x_0, y_0)$ satisfying $x_0 < i_L$ at Round $t$ (see Figure 2);*
2. *$r$ is located at some coordinates $(x_1, y_1)$ with $x_1 > i_L + 1$ at Round $t'$ (see Figure 3);*

3. *only robots of $S$ are within distance one of $r$ between Round $t$ and Round $t'$.*

*We say that a set of robots $S$* has single-handed crossed *the fence $L$ at Round $t$ if $\exists t' < t'' \leq t$ such that $S$ single-handed crosses the fence $L = (l_1, l_2)$ between $t'$ and $t''$.*

*To be more general, we now consider any algorithm, i.e., well-defined or not. We first prove that if $n$ robots solve the IGE problem, then there is a fence that is* single-handed crossed *by a non-empty subset of robots within a finite number of rounds; see Lemma 1. This result will be used later to show that, if robots are anonymous and cannot change their color, the exploration of an infinite grid is impossible under visibility range 1, whatever the number of robots is; see Theorem 1.*

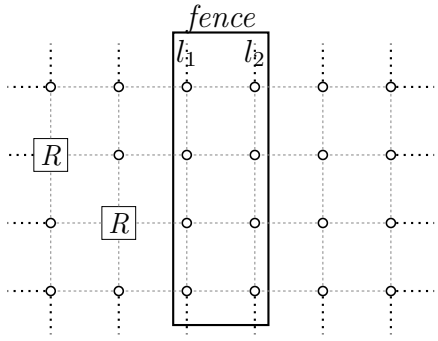


Figure 2: A team of robots in front of a fence.

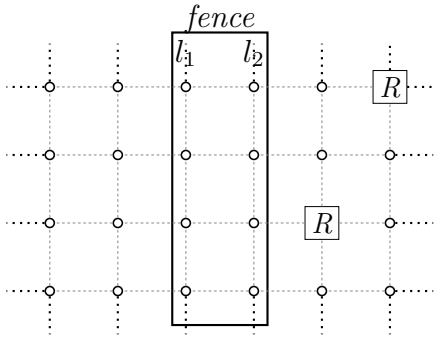


Figure 3: A team of robots has crossed a fence.

**Lemma 1** (The test of the fence)**.** *If an algorithm $A$ solves the IGE problem, then in every execution (starting fron the initial configuration of $A$) there exists at least fence $L$ and a subset of robots $S$ such that $S$ single-handed crosses $L$ within a finite number of rounds.*

*Proof.* If $A$ solves the IGE problem, then any node is eventually visited by at least a robot. So, we can choose a node $u = (i, j)$ where $i$ is arbitrarily large: $u$ should be visited within finite number of rounds despite an arbitrary number of fences have to be crossed before.

We denote by $U(x)$, resp. $\overleftarrow{U}(x)$, $\overrightarrow{U}(x)$ the set of nodes having $x$-coordinates $x$, resp. at most $x$, at least $x$.

We prove by induction the following result: Let $S$ be a set of $n$ robots in $\overleftarrow{U}(x_0)$, for some integer $x_0$ at time $t_0$. If at time $t_1$ at least one robot in $S$

reaches $U(x_0+4(n!)^2)$, and if the robots in $S$ do not see other robots between time $t_0$ and time $t_1$, then there exists a subset of $S$ that single-handed crosses a fence (between some times in the intervale $[t_0, t_1]$).

The induction is on the number $n$ of robots in $S$. The base case is trivial because if a single robots moves to a node with $x$-coordinate $x_0 + 4$, then it alone single-handed crosses the fence $(x_0 + 1, x_0 + 2)$.

Assume now that the result is true for less than $n$ robots. Let $S$ is a set of $n$ robots in $\overleftarrow{U}(x_0)$, for some integer $x_0$ at time $t_0$. Assume at time $t_1$ at least one robot in $S$ reaches $U(x_0 + 4(n!)^2)$. The idea is to find a group of robots strictly included in $S$ with a similar property (with a distance $4((n-1)!)^2$ to travel) to apply the induction hypothesis. We do this in two steps, first we find a group $S'$ of robots that are traveling to the right and after some time do not see the other robots remaining on the left. Then we find a group $S''$ of robots in $S'$ that are far enough from the other robots in $S'$ to apply the induction.

Let $b = 4(n-1)!n!$ and let $m_i$, $0 \leq i \leq n$, be the maximum number of robots simultaneously in $\overrightarrow{U}(x_0 + bi)$, between time $t_0$ and $t_1$. Clearly $(m_i)_i$ is non-increasing, $n \geq m_i \geq 1$, and there are $n + 1$ values so there exists a index $k$ such that $m_k = m_{k+1}$.

If $m_k = n$, then the $n$ robots in $S$ single-handed cross the fence $(x_0 + 1, x_0 + 2)$ because they all reached $U(x_0 + b(k + 1))$ and they do not see any other robot between time $t_0$ and time $t_1$.

Assume now that $m_k < n$. Let $t'_1$ be the first time $m_k$ robots reach $\overrightarrow{U}(x_0 + b(k+1))$ and let $S'$ be the set of $m_k$ robots reaching $\overrightarrow{U}(x_0 + b(k+1))$ at time $t'_1$. Let $t'_0$ be the first time such that, between $t'_0$ and $t'_1$, the robots in $S'$ are in $\overrightarrow{U}(x_0 + 2 + bk)$. This means informally that after time $t'_0$, robots in $S'$ travel towards $\overrightarrow{U}(x_0 + b(k + 1))$ without going back to $\overleftarrow{U}(x_0 + 1 + bk)$, hence without seeing any other robot outside $S'$.

$S'$ is not yet the set we are looking for because some robots might already be too far to the right to apply the induction. Let $b' = 4((n-1)!)^2$ and let $m'_j$, $0 \leq j \leq m_k$, be the maximum number of robots of $S'$ that are in $\overleftarrow{U}(x_0+2+bk+(b'+2)j)$ between time $t'_0$ and $t'_1$. Clearly $m'_0 \geq 1$ by definition of $t'_0$. Also, $(m'_j)_j$ is non-decreasing, $m'_j \leq m_k$, there are $m_k + 1$ values so there exists an index $k'$ such that $m'_{k'} = m'_{k'+1}$ $(k' + 1 \leq m_k)$.

Let $t''_0$ be the time, between $t'_0$ and $t'_1$, when $m'_k$ robots are in $\overleftarrow{U}(x_0 + 2 + bk + (b' + 2)k')$. Let $S''$ be the robots in $S'$ that are in $\overleftarrow{U}(x_0 +$

$2 + bk + (b' + 2)k'$) at time $t''_0$.

$S''$ is the set we use for the induction. To apply the induction we have to show that all the robots in $S''$ at time $t''_0$ are in $\overleftarrow{U}(x'_0)$ with $x'_0 = x_0 + 2 + bk + (b' + 2)k'$ (which is clearly true) ; at least one robot reaches $U(x'_0 + b')$ ; and the robots in $S''$ do not see other robots before this happens.

To prove that at least one robot in $S''$ reaches $U(x'_0 + b')$, we even show that $x'_0 + b' + 2$ is at most $x_0 + (k + 1)b$ (because we know all the robots in $S''$ eventually reach $\overrightarrow{U}(x_0 + (k + 1)b)$ because $m_k = m_{k+1}$). We have

$$\begin{aligned} 2(n-1) &\leq b' - 2 \\ \Rightarrow 2m_k &\leq b' - 2 \\ \Rightarrow m_k(b' + 2) &\leq nb' - 2 = b - 2 \\ \Rightarrow (k' + 1)(b' + 2) &\leq b - 2 \\ \Rightarrow x_0 + 2 + bk + (k' + 1)(b' + 2) &\leq x_0 + 2 + bk + b - 2 \\ \Rightarrow x'_0 + b' + 2 &\leq x_0 + b(k + 1) \end{aligned}$$

It remains to prove that the robots in $S''$ do not see other robots before this happens. We already showed that the robots in $S''$ do not see other robot outside $S'$ (that are all on the left, this is why we use $+2$ in our definition of $x'_0$). Now, what about the robot in $S' \setminus S''$, on the right? By the definition of $m'_{k'}$, no robots in $S' \setminus S''$ can enter $\overleftarrow{U}(x'_0 + b' + 2)$ after $t'_0$ ($t'_0$ included) and before at least a robot in $S''$ reaches $U(x'_0 + b' + 3)$ (otherwise more than $m_{k'}$ robots would be in $\overleftarrow{U}(x'_0 + b' + 2)$). So when the first robot in $S''$ reaches $U(x'_0 + b')$, say at time $t''_1$, the robots in $S'' \setminus S'$ are in $\overrightarrow{U}(x'_0 + b' + 3)$ hence are not visible from $S''$.

So, by the induction hypothesis, there is a subset of $S''$ that single-handed crosses a fence between time $t''_0$ and time $t''_1$.

□

*3.2. The Impossibility Result*

*With only one color and under visibility one, there is at most six possible indistinguishable views since every node should contain at most one robot (by exclusiveness); see Figure 4. One can see that any rule associated with view $V_0$ and a non-idle movement is ambiguous, i.e., the destination depends on the indistinguishable transformation applied to the view. Indeed, the robot in $V_0$ has no way to distinguish between the four neighboring nodes. The same is*

*true for $V_2$, $V_2'$, and $V_4$ of Figure 4. Now, as we do not require algorithms to be well-defined, an algorithm may include some ambiguous rules. Actually, there are only two views that can result in non-ambiguous non-idle movement: $V_1$ where a robot sees only one robot around it and $V_3$ where a robot sees three robots around it. We denote by $R_1^{in}$, resp. $R_1^{out}$, the rule that orders a robot with view $V_1$ to move towards the neighboring robot, resp. away from the neighboring robot. Similarly, we denote by $R_3^{in}$, resp. $R_3^{out}$, the rule that orders a robot with view $V_3$ to move towards the middle robot (i.e., the robot opposite to the empty node), resp. towards the empty node; see Figure 5. Note that, since $R_1^{in}$ and $R_1^{out}$ (resp. $R_3^{in}$ and $R_3^{out}$) are associated with the same view, they cannot be part of the same algorithm.*

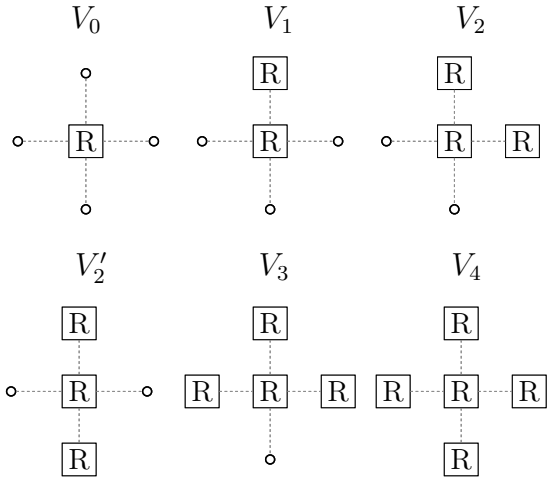


Figure 4: The possible views of a robot with visibility one and without colors.

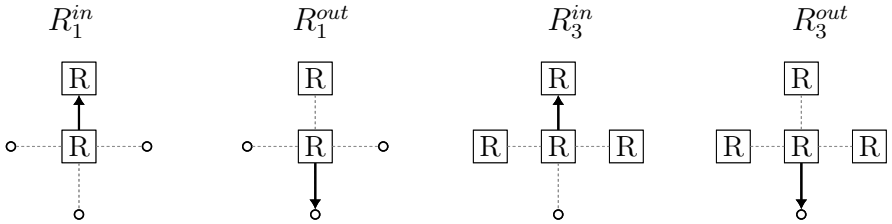


Figure 5: Non-ambiguous and non-idle rules, with visibility one and no color.

**Theorem 1.** *There is no algorithm that solves the IGE problem with oblivious synchronous robots under visibility range one.*

*Proof.* Assume, by contradiction, that an algorithm $\mathcal{A}$ solves the IGE problem with oblivious robots and assuming visibility range one. We show the contradiction by proving that using $\mathcal{A}$, the robots fail the test of the fence (Lemma 1).

To that goal, we first construct an execution by choosing carefully which indistinguishable function is applied to views that are associated with ambiguous rules. If a robot $r$ has a view $V$ where an ambiguous rule applies we do the following:

1. if $V = V_0$, then we apply $f$ such that the global destination is Left.
2. if $V = V_2$, and the rule dictates the robot to move toward an empty node, then we apply $f$ such that the global destination is the unique empty node that is either Up or Down.
3. if $V = V_2$, and the rule dictates the robot to move toward an occupied node, then we apply $f$ such that the global destination is the unique occupied node that is either Up or Down.
4. if $V = V_2'$, and the rule dictates the robot to move toward an empty node, then we apply $f$ such that the global destination is the unique empty node that is either Up or Left.
5. if $V = V_2'$, and the rule dictates the robot to move toward an occupied node, then we apply $f$ such that the global destination is the unique occupied node that is either Up or Left.
6. if $V = V_4$, then we apply $f$ such that the global destination is Left.

We will see that $\mathcal{A}$ cannot contain ambiguous rules for $V_1$ and $V_3$. By choosing those indistinguishable transformations, we obtain a unique execution $E$. According to Lemma 1, there exists a fence $L = (l_1, l_2)$ and a subset of robots $S$ such that $S$ has single-handed crossed $L$ at time $t$.

By definition, robots in $S$ are initially located on the left of the fence. We define the Round $t_1$, resp. $t_2$, as the last round, before $t$, when there is a robot of $S$ on $l_1$, resp. on $l_2$. Hence, we have, $t_1 < t_2 < t$.

**Claim 1:** $\mathcal{A}$ *includes at least one* out*-rule, i.e.,* $R_1^{out}$ *or* $R_3^{out}$.

> *Proof of the claim:* The first robots that enter $l_1$ move Right (in the global view) towards empty nodes. Moreover, they do so using a non-ambiguous rule since the chosen indistinguishable transformation forces any robot with such rules to move either Up, Down or Left. Thus, $\mathcal{A}$ must include at least one *out*-rule, *i.e.*, $R_1^{out}$ or $R_3^{out}$.

**Claim 2:** *$\mathcal{A}$ includes at least one* in*-rule, i.e., $R_1^{in}$ or $R_3^{in}$.*

*Proof of the claim:* At Round $t_2$, all the robots on $l_2$ move Right to complete the fence-crossing. Again, they do so using a non-ambiguous rule since the chosen indistinguishable transformation forces any robot with such rules to move either Up, Down or Left. Thus, $\mathcal{A}$ must include at least one *in*-rule, *i.e.*, $R_1^{in}$ or $R_3^{in}$ since they see no robot on the left.

**Claim 3:** *$\mathcal{A}$ includes Rules $R_1^{in}$ and $R_3^{out}$, but neither $R_3^{in}$ nor $R_1^{out}$.*

*Proof of the claim:* Since an algorithm cannot have two rules based on the same view, $\mathcal{A}$ either includes Rules $R_1^{in}$ and $R_3^{out}$, or Rules $R_3^{in}$ and $R_1^{out}$, by Claims 1 and 2. So, assume, by contradiction, that $\mathcal{A}$ includes $R_3^{in}$ and $R_1^{out}$, but neither $R_1^{in}$, nor $R_3^{out}$. At Round $t_2$, all robots on $l_2$ (at least one) leave it. Again, in this case, these robots necessarily execute a non-ambiguous rule: the only available rule is $R_3^{in}$. Yet, this implies that there is an infinite chain of robots on $l_2$, which contradicts the fact that there is a finite number of robots.

Using Claim 3 we can show the following claim.

**Claim 4:** *There are two adjacent robots $r_a$ and $r'_a$ (of $S$) on $l_2$ at Round $t_1 + 1$.*

*Proof of the claim:* At Round $t_1$, let $r_a$ be any robot on $l_1$. Then, $r_a$ leaves $l_1$ towards $l_2$. Again, $r_a$ should execute a non-ambiguous rule at Round $t_1$, *i.e.*, $R_1^{in}$, by Claim 3. So, $r_a$ moves towards a robot $r_d$. This implies that $r_d \in S$ is not idle at Round $t_1$ since otherwise this would create a collision, violating exclusiveness. So, $r_d$ has only the three following possibilities at Round $t_1$:

- (a) $r_d$ executes $R_1^{in}$ or an ambiguous rule toward an occupied node,
- (b) $r_d$ executes an ambiguous rule towards an empty node,
- (c) $r_d$ executes Rule $R_3^{out}$.

We now show that in all theses cases, we either obtain a contradiction, or there are two adjacent robots $r_a$ and $r'_a$ (of $S$) on $l_2$ at Round $t_1 + 1$.

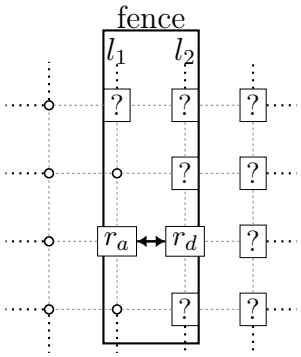


Figure 6: Case (a), reaching a contradiction.

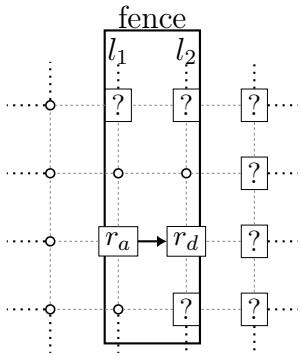


Figure 7: Case (b), $r_a$ and $r_d$ are neighbors at Round $t_1 + 1$.

- In case (a), illustrated in Figure 6, if $R_1^{in}$ is executed by $r_d$, then $r_a$ and $r_d$ exchange their positions, violating then exclusiveness, a contradiction. If an ambiguous rule orders $r_d$ to move towards an occupied destination, then there is an indistinguishable transformation that makes move $r_d$ to the Left. Hence, there is a possible execution that behaves as the execution $E$ until Round $t_1 - 1$, but where $r_a$ and $r_d$ exchange their positions during Round $t_1$, violating then exclusiveness, a contradiction.
- In case (b), illustrated in Figure 7, $r_d$ sees either $V_2$ or $V_2'$ (the only ambiguous views with at least one occupied neighbor and one empty neighbor). So $r_d$ has two neighbors, one of which is $r_a$. The chosen indistinguishable transformation makes it moves Up or Down towards an empty node on $l_2$ and becomes a neighbor of $r_a$ at Round $t_1 + 1$. So, by letting $r_d = r_a'$, we obtain that there are two adjacent robots $r_a$ and $r_a'$ (of $S$) on $l_2$ at Round $t_1 + 1$.

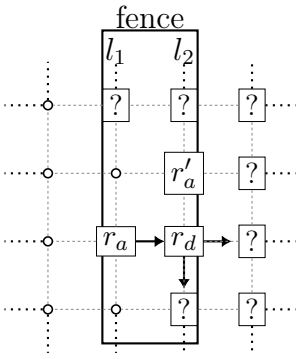


Figure 8: Case (c), $r_a$ and $r'_a$ are neighbors at Round $t_1 + 1$.

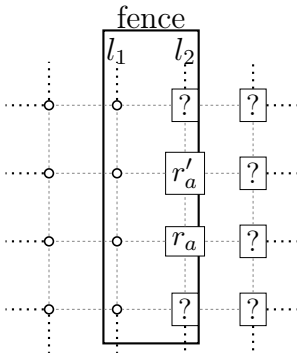


Figure 9: Robots $r_a$ and $r'_a$ are stuck on the fence.

- In case (c), illustrated in Figure 8, one of $r_d$'s neighbor, denoted $r'_a$, is also located on $l_2$. $r'_a$ cannot execute $R_1^{in}$ to move towards $r_d$, otherwise it would create a collision with $r_a$, violating then exclusiveness. Also, $r'_a$ does not have a neighbor on $l_1$ because that would prevent $r_a$ from applying Rule $R_1^{in}$. So, if $R_3^{out}$ applies to $r'_a$, it moves towards $l_1$, contradicting the definition of $t_1$. An ambiguous rule cannot apply to $r'_a$ either. Indeed, if an ambiguous rule with an empty destination applies, the chosen indistinguishable transformation makes $r'_a$ moves towards $l_1$ (and so violating the definition of $t_1$), and if an ambiguous rule with an occupied destination applies, then there is an indistinguishable transformation that makes $r'_a$ move toward $r_d$. So, again, there is an execution that behaves as the execution $E$ until Round $t_1 - 1$ but where both $r_a$ and $r'_a$ move to the same position during Round $t_1$, creating a collision with $r_a$ at Round $t_1 + 1$, a contradiction. Hence, $r'_a$ stays idle and $r'_a$ and $r_a$ are adjacent on $l_2$ at Round $t_1 + 1$, and we are done.

From Claim 4, we have an execution where $r_a$ and $r'_a$ are adjacent on $l_2$ at Round $t_1 + 1$ (Figure 9). To conclude the proof, we show that if two robots are adjacent on $l_2$ at Round $t'$ with $t_1 < t' \leq t_2$, then they are adjacent on $l_2$ Round $t' + 1$. This contradicts the fact all the robots leave $l_2$ at Round $t_2$.

When $r_a$ and $r'_a$ are adjacent on $l_2$ at time $t'$ (with $r_a$ below $r'_a$), one can observe that $R_3^{out}$ cannot apply to any of them, nor any ambiguous rule with an empty destination, otherwise the chosen transformation would make them

move toward $l_1$, violating the definition of $t_1$.

Now either $r_a$ executes:

- (i) $R_1^{in}$,
- (ii) an ambiguous rule towards an occupied destination,
- (iii) stays idle.

If $r_a$ executes $R_1^{in}$ or an ambiguous rule towards an occupied destination, it moves Up towards $r'_a$. The robot $r'_a$ cannot stay idle (since otherwise it would create a collision), and cannot execute $R_1^{in}$, otherwise it would violates the exclusiveness, so it executes an ambiguous rule toward an occupied destination and moves Up. At Round $t' + 1$, $r_a$ and $r'_a$ are still adjacent on $l_2$. If $r_a$ stays idle, $r'_a$ cannot execute $R_1^{in}$, otherwise it would create a collision, nor an ambiguous rule towards an occupied destination, otherwise we can construct a possible execution that behaves as the execution $E$ until Round $t'$, but where $r'_a$ moves towards $r_a$ to create a collision, using the appropriate indistinguishable transformation. So $r'_a$ stays idle as well. At Round $t' + 1$, $r_a$ and $r'_a$ are still adjacent on $l_2$.

This contradicts the fact that all the robots on $l_2$ at Round $t_2$ move Right. In the execution $E$, fence $L$ is never single-handed crossed, which contradicts our initial assumption. □

## *4. Algorithm with eight anonymous oblivious robots under visibility two*

*Our first algorithm, denoted by $\mathcal{A}_1$ in the following, uses eight anonymous oblivious robots, assuming a visibility range two. The algorithm can be executed in the companion website* [1]*.Initially, robots are placed as shown in Figure 10. The robots are divided into two categories: the* beacon *robots and the* moving group*. There are four beacon robots. The four other belong to the moving group. The actual exploration of the grid is made by the moving group. The beacon robots are just used to delimit the visited area. Notice that the two categories remain fixed all along the execution: the beacon robots are always the same. To maintain this property, we ensure that beacon robots are*

[1] `https://robots.app.bramas.fr/?FUN2021/1`
Use the left arrow ⟵ and right arrow ⟶ of the keyboard to move the robots.

*never adjacent to any other robot. Since the visibility range is two, a beacon can see other robots and move before becoming their neighbor.*

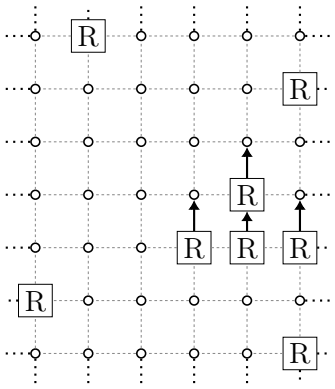


Figure 10: Initial configuration $I$ of $\mathcal{A}_1$.

*Observe that since the visibility range is two, the obstructed visibility can impact the local view of a robot because a robot at distance one can hide a robot behind it at distance two. So, the rules of $\mathcal{A}_1$ should not depend on the states of the nodes that are hidden by a robot. To make it clear, those nodes will be crossed out in the illustrations of our rules; see,* e.g.*, Figure 11.*

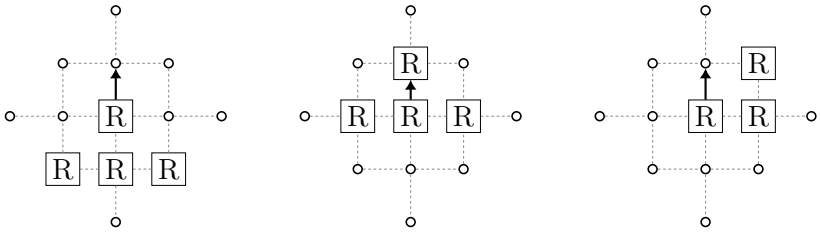


Figure 11: Rules to move along a straight line.

*The first three rules (see Figure 11) allow the moving group to move along a straight line. The moving group always forms a* spaceship shape *where one robot is at the bow, one robot is at the stern, and there is one robot on each side, adjacent to the stern. When in formation, each robot knows whether it is at the bow, the stern, or at a side of the spaceship. However, the robots on the side do not know on which side they are, since there is no common*

*chirality. The first rule orders the bow robot to move away from the other robots, the second rule orders the stern robot to move towards the bow robot, and the third rule orders the side robots to move to the same direction as the stern robot.*

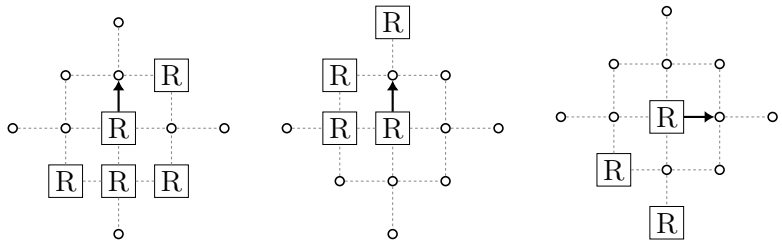

Figure 12: Rules to make the spaceship moving along a straight line when seeing the beacon, and to make the beacon move away.

*Then, when the moving group meets a beacon robot, an adjustment is made in two rounds. The sequence of configurations during an adjustment is shown in Figure 13. The first round of the adjustment, robots execute the rules defined in Figure 12. The first two rules allow the moving group to act as if the beacon was not there;* i.e.*, they continue to move in the same direction. The third rule orders the beacon to move away from the bow robot. The beacon robot can distinguish the correct direction because it also sees a side robot. In the second round of the adjustment, the two rules given in Figure 14 are used. The first rule orders the bow robot to continue as usual (*i.e.*, as if the beacon was not here) and the second rule orders the side robot that sees the beacon robot in diagonal to move towards the stern robot. Two other robots of the moving group act as when they move along a straight line (their views are identical). After the execution of those rules, the spaceship shape is preserved, but the bow robot has become a side robot,*

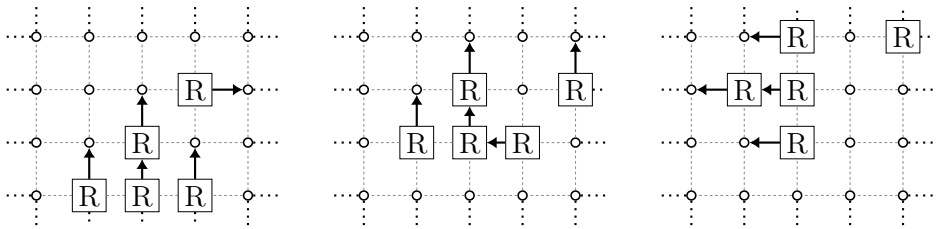

Figure 13: Sequence of configurations during an adjustment.

*and this side robot has become the bow robot. In the same round, the beacon robot executes the same rule as in the first round of the adjustment (the third rule of Figure 12) to move away from the group. The view is mirrored from the first round of the adjustment, so after the two rounds of the adjustment, the beacon has moved diagonally.*

*The last rule given in Figure 15 is necessary to make moving as expected the side robot that still sees the beacon robot right after an adjustment.*

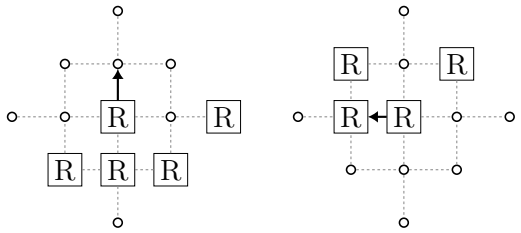


Figure 14: The leader moves in straight line again to become a side follower, the side follower that sees the beacon moves left (the beacon move away again using the same rule as before).

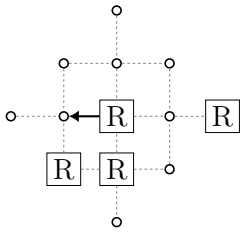


Figure 15: The moving group moves away from the beacon. The side follower that still see the beacon moves away from it.

**Theorem 2.** *Algorithm $\mathcal{A}_1$ solves the exclusive IGE problem using eight oblivious synchronous robots and visibility range of two.*

*Proof.* The proof consists in (1) decomposing the execution into phases, (2) showing by induction that each phase is eventually reached, and finally (3) exhibiting a particular rectangle that is visited during each phase.

We fix a global coordinate system, not accessible to robots, where node $(0, 0)$ is the one below the leftmost robot in the initial configuration described in Figure 10. Thus, the initial configuration, denoted by $C^0$, can be split as follows:

$$C^0 = \{M^0, C^0_0, C^0_1, C^0_2, C^0_3\},$$

where

$$\begin{aligned} M^0 &= \{(3,2),(4,2),(5,2),(4,3)\},\\ C_0^0 &= \{(5,5)\},\\ C_1^0 &= \{(1,6)\},\\ C_2^0 &= \{(0,1)\}, \text{ and}\\ C_3^0 &= \{(5,0)\}. \end{aligned}$$

We define the configuration $C^i = \{M^i, C_0^i, C_1^i, C_2^i, C_3^i\}$ in Phase $i \in \mathbb{N}$, where $M^i = \vec{t}_{(i,i)}(M^0)$, $C_0^i = \vec{t}_{(i,i)}(C_0^0)$, $C_1^i = \vec{t}_{(-i,i)}(C_1^0)$, $C_2^i = \vec{t}_{(-i,-i)}(C_2^0)$, and $C_3^i = \vec{t}_{(i,-i)}(C_3^0)$.

Here, $C_0^i$ contains the first beacon robot visited in Phase $i$, located at the upper right corner of the configuration.

Assume we reach the first configuration $C^i$ of Phase $i$ at time $t$. Recall that $C^i = \{(i+3,i+2),(i+4,i+2),(i+5,i+2),(i+4,i+3),C_0^i,C_1^i,C_2^i,C_3^i\}$. We now prove that configuration $C^{i+1}$ is eventually reached from configuration $C^i$. Figure 16 will illustrate our arguments.

After three rounds, the configuration is $\{(i+3,i+5),(i+4,i+4),(i+4,i+5),(i+4,i+6),C_0^{i+1}, C_1^i, C_2^i, C_3^i\}$.

Then, the moving group has to travel along a straight line during $2i+1$ rounds until the bow robot sees the second beacon robot. Indeed, at time $t+3$, the bow robot is located at $(3+i,5+i)$ and the second beacon robot is at $(1-i,6+i)$.

At time $t+2i+4$, when the bow robot sees the second beacon robot, the second adjustment occurs. At time $t+2i+6$, the configuration is $\{(1-i,i+4),(-i,i+5),(1-i,i+5),(2-i,i+5),C_0^{i+1}, C_1^{i+1}, C_2^i, C_3^i\}$. Then, the moving group travels during $2i+2$ rounds until it reaches the third beacon robot. Indeed, the bow robot is at $(1-i,4+i)$ at time $t+2i+4$ and the third beacon is at $(-i,1-i)$.

This continues until the configuration $C^{i+1}$ is reached.

Inductively, the robots start from configuration $C^0$ and reach configuration $C^i$ within finite time, for any $i \geq 0$. The set $V_i$ of nodes visited between Phase $i$ and $i+1$ includes the edges of the rectangle

$$\left\{\vec{t}_{(-i,-i)}(0,1), \vec{t}_{(i,-i)}(5,1), \vec{t}_{(i,i)}(5,5), \vec{t}_{(-i,i)}(0,5)\right\}.$$

Also, the set $V_0$ contains the nodes inside rectangle $\{(0,1),(5,1),(5,5),(0,5)\}$ as they are visited during the first phase. Since $\bigcup_{i\geq 0} V_i = \mathbb{Z}\times\mathbb{Z}$, our algorithm solves the IGE problem.

□

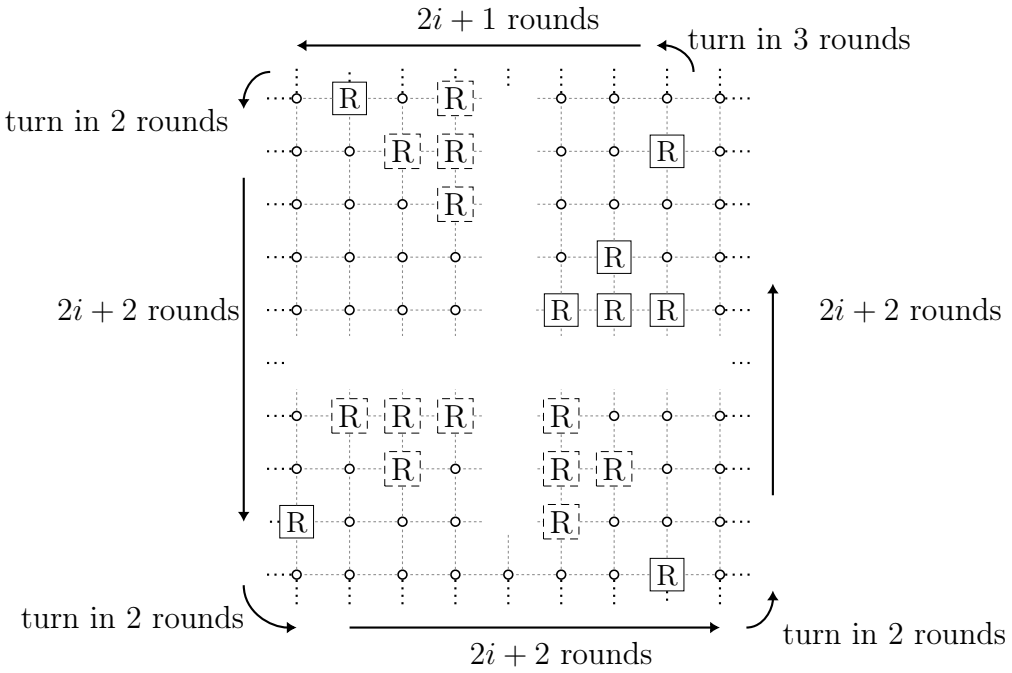


Figure 16: Visualization of one phase.

## *5. Algorithm with six robots and two colors under visibility one*

*In this section, we present an algorithm, denoted by $\mathcal{A}_2$, that uses 6 robots under visibility range of 1 and only 2 colors, respectively denoted by L (for leder) and F (for follower) in the following. According to Theorem 1, this number of colors is optimal under visibility range 1. $\mathcal{A}_2$ is based on principles similar to those of an algorithm for 6 robots presented in [7], however this latter uses three fixed colors and assumes a common chirality. $\mathcal{A}_2$ is also an improved version of the algorithm proposed in the conference version of this paper [8] since it uses 2 colors instead of 3. The algorithm can be executed in the companion website* [2].

[2] `https://robots.app.bramas.fr/?unpublished/3`
Use the left arrow ⟵ and right arrow ⟶ of the keyboard to move the robots.

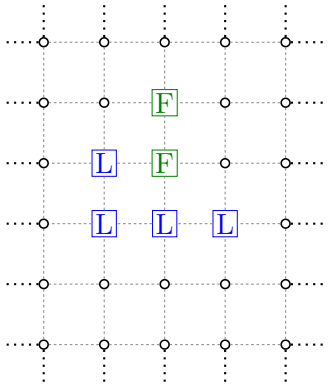


Figure 17: Initial configuration of $\mathcal{A}_2$.

*Initially, all robots are close together and organized as shown in Figure 17. Then, following the rules given in Figures 18 and 19, robots executes some preliminary moves during the first two rounds; see Figure 21.*

*After the two first rounds, the six robots are divided into two categories: the* beacon *robots and the* moving group *according to their colors and relative positions. The moving group consists of two robots: the one colored F and its neighbor (with color L). In the moving group, the L-colored (resp. the F-colored) robot is called the* leader *(resp. the* follower*). The four remaining robots (with color L) are called the beacons and delimit an area which has been already explored: the four nodes in the square whose outerborder is delimited by the beacons have been already visited.*

*From that point on, $\mathcal{A}_2$ works by* phases*. At the end of each phase, the square whose outerborder is delimited by the beacons has grown but still only contains visited nodes. A phase is performed as follows. The moving group aims at reaching the beacons one by one. The moving group first move in a straight line using the rules given in Figure 20. Yet, each time it reaches a beacon, robots make an* adjustment *in two rounds. At the end of the adjustment, the new beacon position is in the diagonal, two hops away from the previous one, and the moving group has made a turn toward the next beacon. This adjustment allows, in particular, to take the newly explored nodes into account. The moving group then continues toward the next beacon, and so on. Figure 22 shows the second adjustment of the moving group.*

*Each time the moving group comes back to the first beacon (bottom right beacon for instance), the current phase terminates: the border of the area*

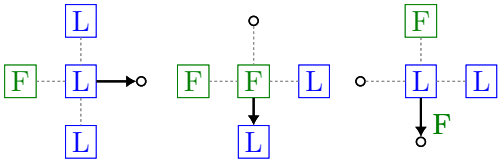


Figure 18: The three rules executed in the first round of $\mathcal{A}_2$.

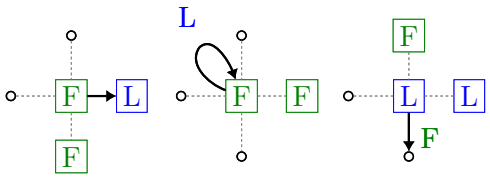


Figure 19: The three rules executed during an adjustment (the third rule is the same a in Figure 18).

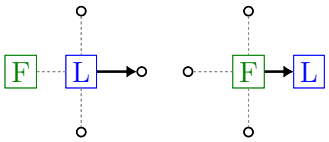


Figure 20: Two rules to make the moving group move in a straight line.

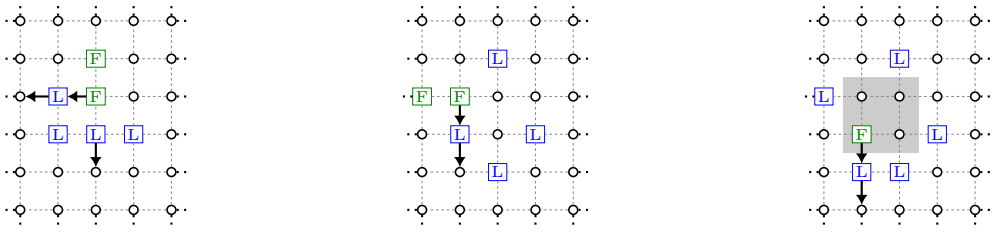

Figure 21: The first two rounds of $\mathcal{A}_2$. The gray square represents the already explored square.

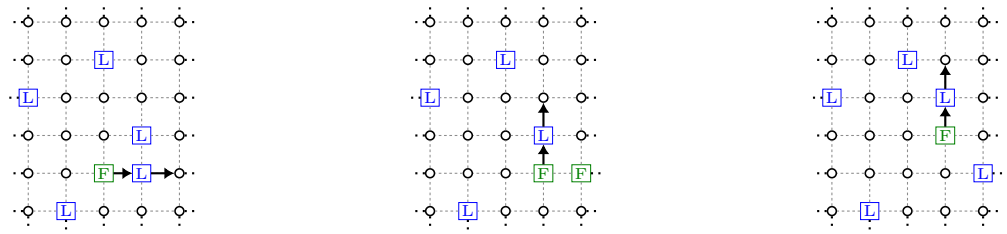

Figure 22: The second adjustment performed in the next two rounds of $\mathcal{A}_2$.

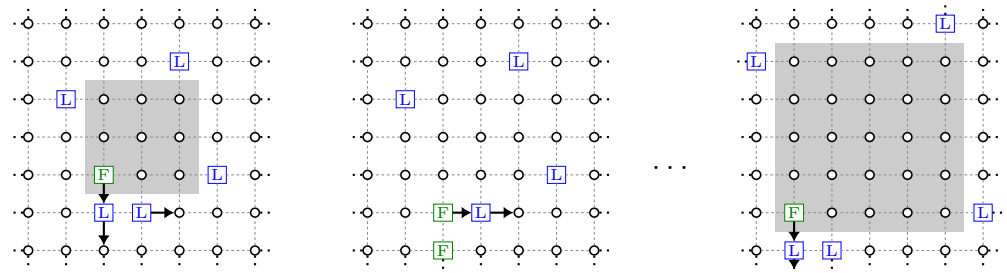

Figure 23: A complete phase of $\mathcal{A}_2$. The gray squares represents the already explored square at the beginning and the end of the phase.

*initially delimited by the four beacons is now fully visited, and the area newly delimited by the beacons is bigger; see Figure 23 to visualize the increasing area that is explored by the moving group.*

**Theorem 3.** *Algorithm $\mathcal{A}_2$ solves the IGE problem using six synchronous robots, two colors and visibility range of one.*

*Proof.* In the following we assume, w.l.o.g., that node $(0, 0)$ is the one where the bottom-left-most L robot is located in the third configuration; see Figure 21. The third configuration corresponds to the beginning of the first phase, where the moving group (located at $(0, 0)$ and $(0, 1)$) is adjacent to the bottom beacon. Recall that these global coordinates are used for the analysis only: robots cannot access those coordinates.

Using this coordinate system, the third configuration is denoted $C^0$ and is decomposed as follow:

$$C^0 = \{M^0, C^0_0, C^0_1, C^0_2, C^0_3\},$$

where

$$M^0 = \{((0,0), L), ((0,1), F)\},$$
$$C_0^0 = \{((1,0), L)\},$$
$$C_1^0 = \{((2,1), L)\},$$
$$C_2^0 = \{((1,3), L)\}, \text{ and}$$
$$C_3^0 = \{((-1,2), L)\}$$

corresponding to the moving group and the four beacons. We define the configuration $C^i = \{M^i, C_0^i, C_1^i, C_2^i, C_3^i\}$ in phase $i$, where $M^i = \vec{t}_{(-i,-i)}(M^0)$, $C_0^i = \vec{t}_{(-i,-i)}(C_0^0)$, $C_1^i = \vec{t}_{(i,-i)}(C_1^0)$, $C_2^i = \vec{t}_{(i,i)}(C_2^0)$, and $C_3^i = \vec{t}_{(-i,i)}(C_3^0)$. Informally, the configuration in phase $i$ is obtained by diagonally translating $i$ times the positions of the beacons and the moving group from the $C^0$ configuration.

We now prove that starting from configuration $C^i$, configuration $C^{i+1}$ is eventually reached. Since the third configuration of our algorithm is $C^0$, this implies that every configuration $C^i$, for every $i \geq 0$, is gradually reached. By doing so, the leader robot visits all the edges of growing squares. The illustration of one cycle is presented in Figure 24.

Assume we reach the first configuration $C^i$ of phase $i$ at time $t$.

Recall that:

$$C^i = \{((-i,-i), L), ((-i, 1-i), F)\} \cup C_0^i \cup C_1^i \cup C_2^i \cup C_3^i.$$

After two rounds, the configuration is:

$$\{((-i+2,-i), L), ((-i+1,-i), F)\} \cup C_0^{i+1} \cup C_1^i \cup C_2^i \cup C_3^i.$$

Then, the moving group travels along a straight line during $2i$ rounds until the leader (the robot with color $L$) sees the second beacon robot. Indeed, at time $t+2i+2$, it is located at $(i+2,-i)$ and the second beacon robot is at $(i+2,-i+1)$.

At time $t+2i+2$, when the leader sees the second beacon robot, the second adjustment occurs. After two rounds, at time $t+2i+4$, the configuration is:

$$\{((i+2,-i+2), L), ((i+2,-i+1), F)\} \cup C_0^{i+1} \cup C_1^{i+1} \cup C_2^i \cup C_3^i.$$

Then, the moving group travels during $2i+1$ rounds until it reaches the third beacon robot. Indeed, the leader is at $(i+2, i+3)$ at time $t+4i+5$ and the third beacon is at $(i+1, i+3)$.

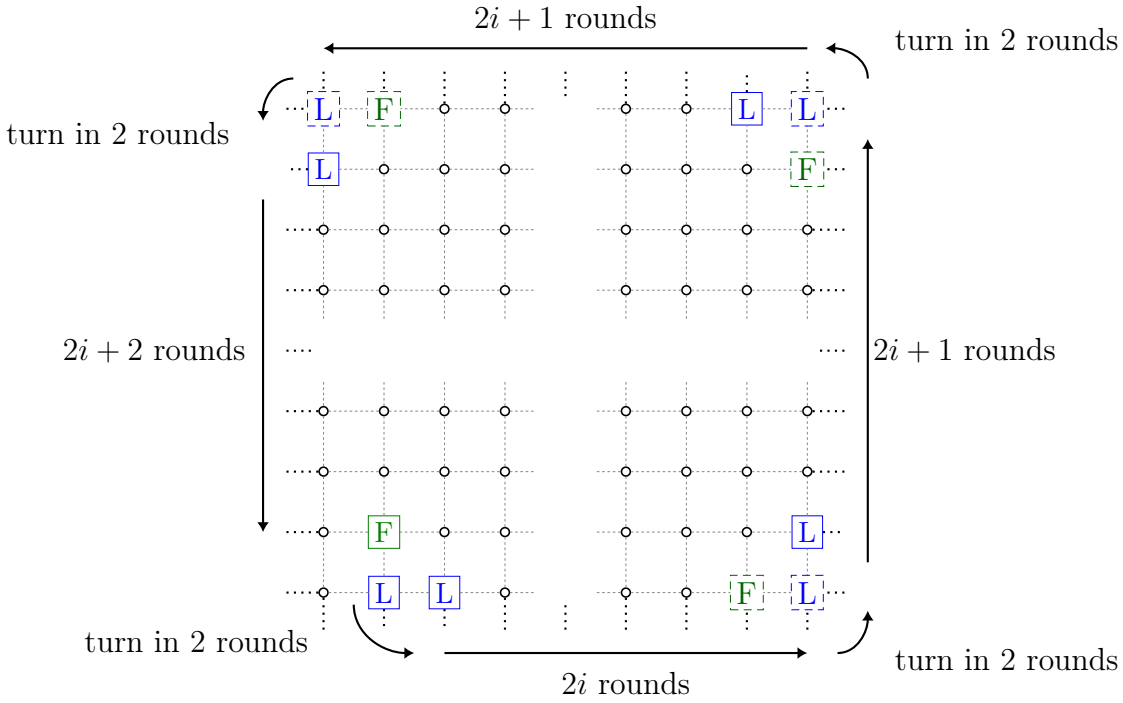


Figure 24: Visualization of one phase.

When the robot with color $L$ sees the third beacon, the third adjustment occurs and the reached configuration is:

$$\{((i, i+3), L), ((i+1, i+3), F)\} \cup C_0^{i+1} \cup C_1^{i+1} \cup C_2^{i+1} \cup C_3^i.$$

Then, the moving group travels during $2i+1$ rounds until the leader sees the fourth beacon robot. The last adjustment is performed to obtain the configuration:

$$\{((-i-1, i+1), L), ((-i-1, -i+2), F)\} \cup C_0^{i+1} \cup C_1^{i+1} \cup C_2^{i+1} \cup C_3^{i+1}.$$

Finally, after $2i+2$ rounds, the moving group comes back to the first beacon robot and the configuration is exactly $C^{i+1}$ at time $t + 8i + 12$.

Inductively, from configuration $C^0$, the robots reach configuration $C^i$ within finite time, for any $i \geq 0$. Also, nodes $(0,1), (0,2), (1,1), (1,2)$ are visited in the first two rounds, and the set $V_i$ of nodes visited by the leader between phase $i$ and $i+1$ contains the edges of the square:

$$\left\{\vec{t}_{(-i,-i)}(-1,0), \vec{t}_{(i,-i)}(2,0), \vec{t}_{(i,i)}(2,3), \vec{t}_{(-i,i)}(-1,3)\right\}.$$

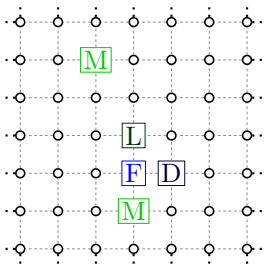


Figure 25: Initial configuration of $\mathcal{A}_3$.

Since $\{(1,1)\} \cup \bigcup_{i\geq 0} V_i = \mathbb{Z} \times \mathbb{Z}$, Algorithm $\mathcal{A}_2$ solves the IGE problem. □

### *6. Algorithm with five robots under visibility one*

*We now present algorithm $\mathcal{A}_3$, the first perpetual infinite grid exploration algorithm. This algorithm only uses 5 robots under visibility range 1 with 12 colors. In [7], the following property is proven:*

**Property 1.** *There is no IGE algorithm that uses less than 5 robots under visibility range 1, whatever be the number of used colors and even if a common chirality is assumed.*

*Consequently, algorithm $\mathcal{A}_3$ is optimal in terms of number of robots. Algorithm $\mathcal{A}_3$ can be executed in the companion website* [3].

*The initial configuration of $\mathcal{A}_3$ is given in Figure 25. It contains two beacons robots that are initially $M$-colored, a moving group with a leader robot (colored $L$) and a follower (with color $F$), and a pebble robot with color $D$.*

*Actually, the way the robots explore the grid is very different from the previous infinite grid exploration algorithms. In particular, the moving group does not explore the boundary of a shape formed by the beacon robots. Here, all the robots always remain in a strip of width two whose height is delimited by the two beacons robots.*

*The exploration is first done by switching the strip from left to right. The idea is that the moving group goes and back into the strip. During each*

[3] `https://robots.app.bramas.fr/?unpublished/2`
Use the left arrow ⟵ and right arrow ⟶ of the keyboard to move the robots.

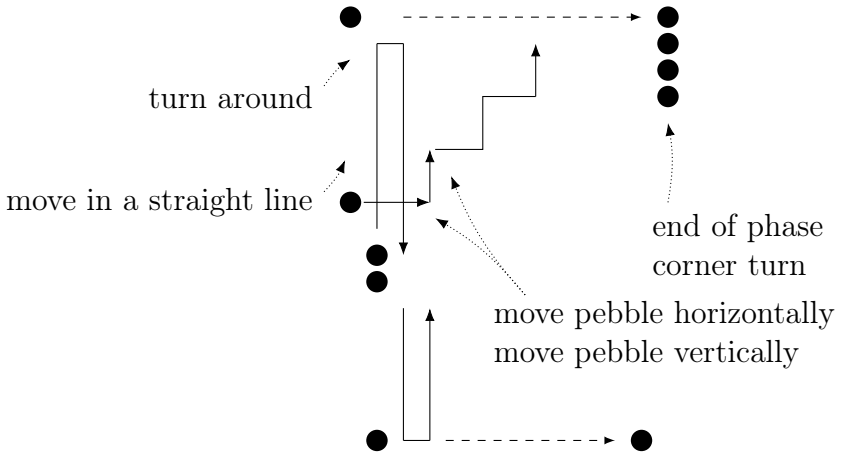


Figure 26: A phase of the exploration.

*complete travel, the moving group (1) pushes each beacon two nodes on the right, and (2) translates the pebble diagonally, two nodes on the right and two nodes up. More precisely, the moving group* moves in a straight line *and each time it meets a beacon, it* turns around *the beacon. During a travel between the two beacons, the moving group meets the pebble twice: when the moving group is moving top-down and meets the pebble, the pebble is vertically moved two nodes up; and when the moving group meets the pebble in the other way, the pebble is moved to the side, precisely two nodes on the right. Figure 26 illustrates this process. Eventually, the moving group, the pebble, and the top beacon are close together, as shown in Figure 26. Then,*a corner turn *is performed: the height of the strip is increased of two nodes and the visit direction is completely reversed;* i.e.*, the moving group makes a turn, the strip now moves from right to left, and the pebble is now translated left-down when hit by the moving group. A phase consists in moving the strip in one direction and then in the other direction. As illustrated in Figure 27, after a phase is over, a new phase begins where the strip is longer on both sides, and being longer, the strip will allow to visit more nodes in each direction during this phase.*

*A phase of the exploration can be split into several subphases, as depicted in Figure 26: moving in a straight line, turning around, moving the pebble vertically, moving the beacon to the side, and performing a corner turn. We now give more details about each of these subphases.*

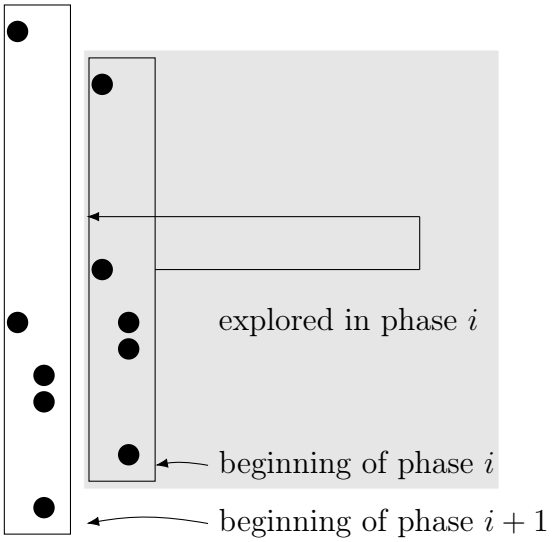


Figure 27: Phases of the exploration.

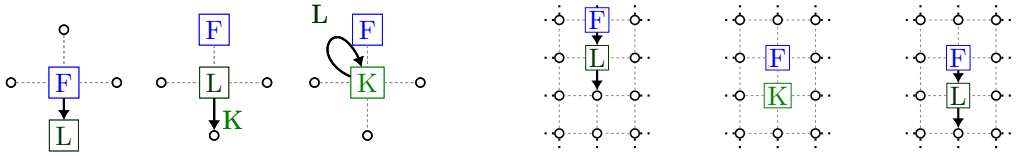


Figure 28: The moving group moves in a straight line. The rules on the left are executed, resulting in the sequence of configurations on the right.

*Moving in a straight line. The moving group travels in a straight line but stops once every two rounds, so that the leader of the moving group can indicate to the follower some information about what is in front of it. When the moving group is not in the neighborhood of other robots, the follower with color $F$ moves towards the leader when it has color $L$ and stops otherwise. The leader with color $L$ moves forward and changes its color to $K$. The leader having color $K$, when it sees only the follower behind it, just change its color to $L$. The rules and the sequence of configurations are shown in Figure 28.*

*The moving group turns around. When the moving group reaches one of the two beacons, the moving group turns around. When doing so, the beacon moves two nodes to the side. This sequence of movements is performed in four rounds by executing the rules shown in Figure 29.*

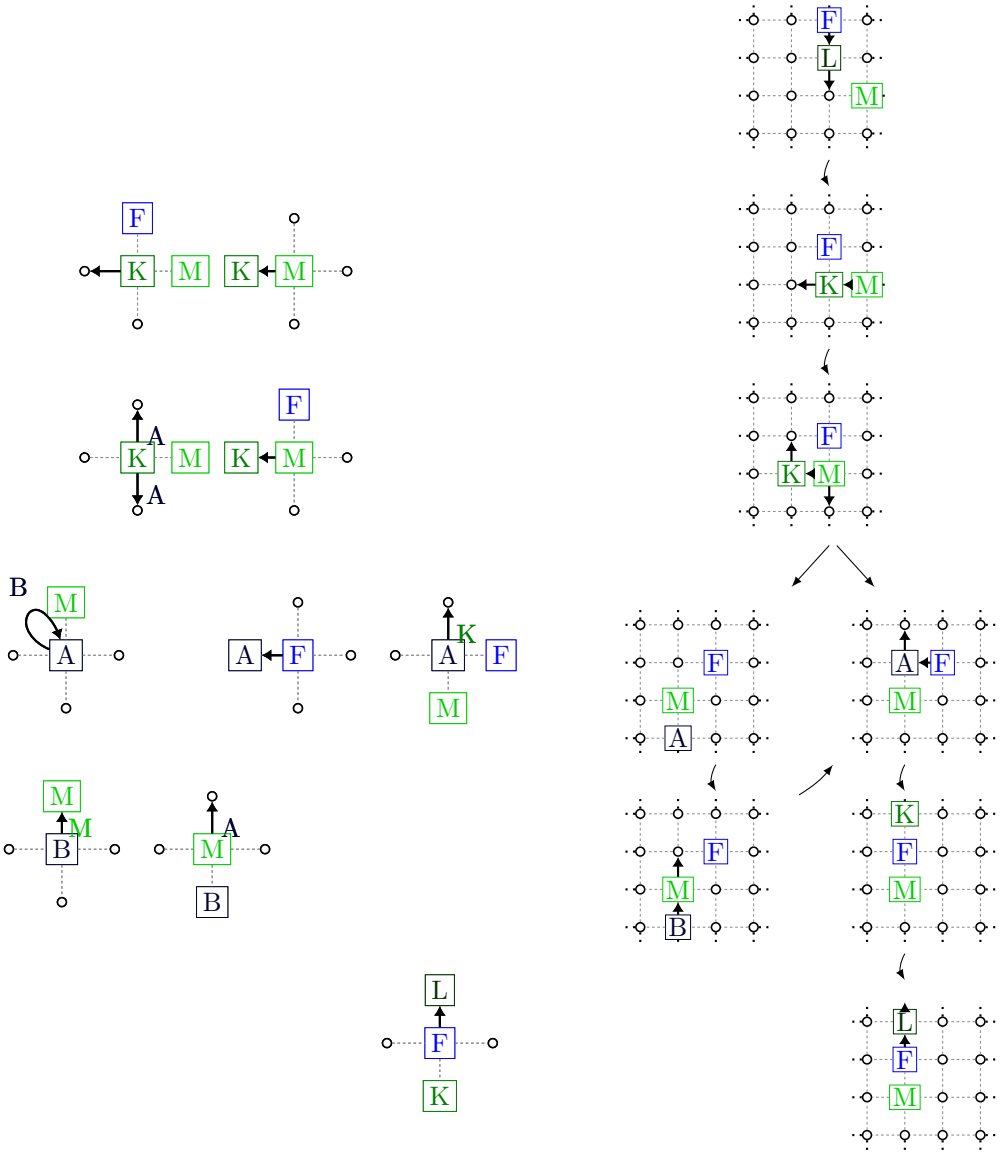


Figure 29: The moving group turns around when it reaches a beacon. The rules on the left are executed, resulting in the sequence of configurations on the right. The two branches correspond to the two possible choices for the unique non-deterministic rule.

*Move the pebble vertically. When the moving group meets the pebble robot with color D in front of it, the robots perform a sequence of moves so that*

*the pebble becomes the leader of the moving group and the leader becomes the pebble. The pebble is then moved two nodes vertically. The rules and the sequence of configurations are shown in Figure 30.*

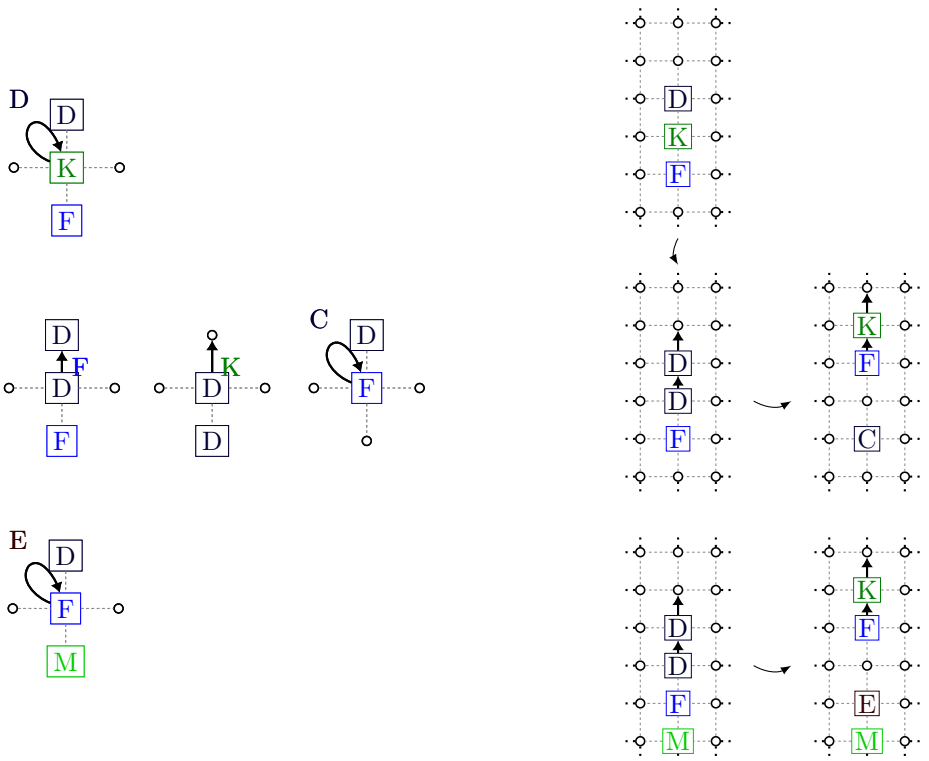


Figure 30: When the moving group meets the pebble $D$, the pebble becomes the leader of the moving group and the follower becomes the pebble. After the sequence, the new pebble has color $C$ and has been translated by two nodes. The rules on the left are executed, resulting in the sequence of configurations on the right. The last row shows the case where the pebble becomes a neighbor of the beacon.

*Move the beacon to the side. When the moving group is meeting with the pebble having color $C$, in the opposite direction, the pebble becomes the leader of the moving group and the leader becomes the pebble. The pebble is then moved two nodes to the side. The rules and the sequence of configurations are shown in Figure 31.*

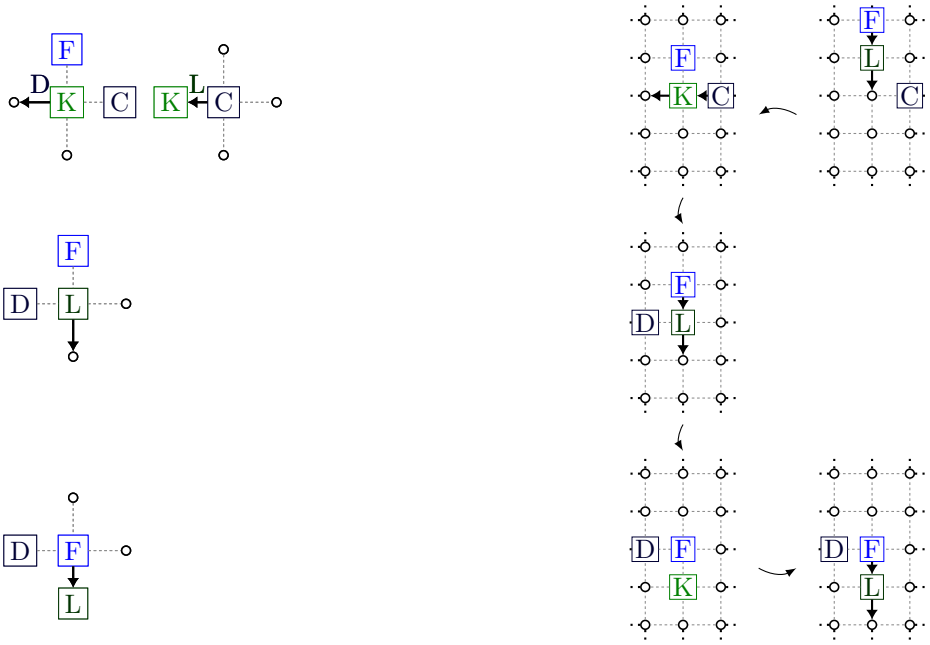


Figure 31: When the moving group meets the pebble $C$ in the opposite direction, the pebble becomes the leader of the moving group and the leader becomes the pebble. After the sequence, the new pebble has color $D$ and has been translated by two nodes. The rules on the left are executed, resulting in the sequence of configurations on the right.

*Performing a corner turn. After the moving group has made two turns, the pebble has been moved diagonally by two nodes. Moreover, the moving group and the two beacons have been moved horizontally by two nodes in the same direction. Eventually, the pebble reaches one of the beacon. When the pebble is close enough to the beacon, the four robots together performs a sequence of moves placing them so that they now move to the opposite direction and the pebble now travels toward the opposite beacon diagonally. The rules and the sequence of configurations are shown in Figure 32.*

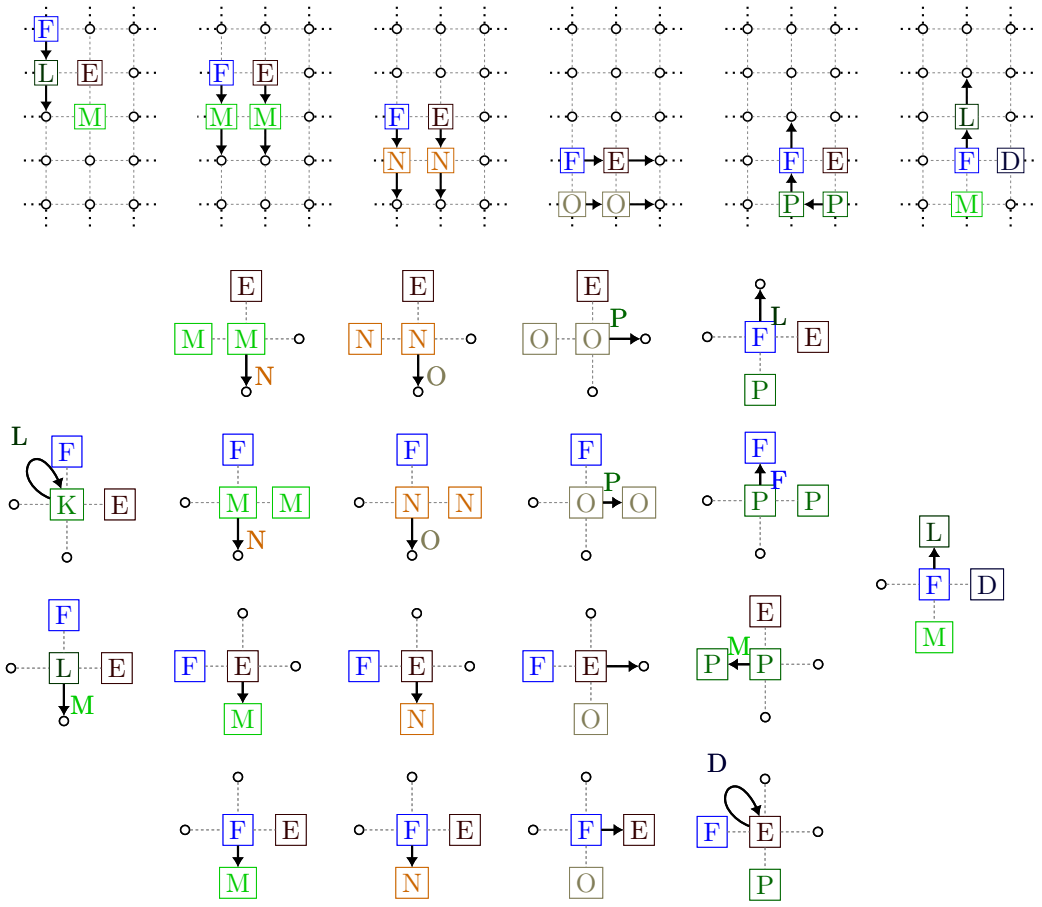


Figure 32: The pebble, the moving group, and the beacons perform a sequence of moves to initiate the exploration in the opposite direction. The rules below the sequence are executed, resulting in the sequence of configurations in the first row.

**Theorem 4.** *Algorithm $\mathcal{A}_3$ solves the perpetual IGE problem with five synchronous robots having visibility range 1 and 12 colors.*

*Proof.* Observe first that the execution of the algorithm may depend on the choice, by the adversary, of the indistinguishable transformation applied to the robot with color $K$ having a single neighbor with color $M$ (see Figure 29). All the other movements are deterministic and do not depend on the adversary. However, we can easily show that, regardless of the choice of the adversary, the same configuration is obtained after either 1 or 2 rounds. If the robot with color $K$ moves Down (assuming a global coordinate system

as shown in Figure 29 for the analysis), then the configuration after one more round is exactly the same as if the robots had moved Up. Since we are only interested in which nodes are visited by the robots, we can assume in the worst case that the robot $K$ always moves Up (since in the other case the robots visit more nodes). With this assumption, there is a unique execution to consider.

We can split the execution in phases, where each phase is composed of the following subphases: repeat round-trip explorations until the pebble reaches a beacon; a corner move; repeat round-trip explorations in the opposite direction until the pebble reaches the other beacon; a corner move. A round-trip exploration consists of moving in a straight line; moving the pebble horizontally; turning around; moving the pebble vertically; and turning around. The subphases are shown in Figure 26. The entire phase is shown in Figure 27.

Let $C_i$ be the following configuration:

$$C_i = \vec{t}_{(-2i,-2i)}(\{((3,0),M),((3,1),F),((3,2),L),((4,1),D)\}) \\ \cup \{((2-2i,4+2i),M)\}$$

$C_0$ is the initial configuration given in Figure 25. In this configuration, the bottom-left robot is initially at $(3,0)$. The configuration $C_i$ is the configuration at the beginning of the $i$-th phase. We call the *phase rectangle* $R_i$ of phase $i$ the rectangle having $6+2i$ columns and $7+2i$ rows, and the bottom-left corner at $(-2i,-2i)$. We now show that, starting from configuration $C_i$, the robots visit all the nodes of the phase rectangle $R_i$ in phase $i$ and that the configuration at the end of the phase is $C_{i+1}$. The theorem follows by induction.

In phase 0, we can simply check that this is the case by executing the algorithm from configuration $C_0$. Then, we can prove the result by induction because adding two rows between the two beacons in the initial configuration does not change the execution until the pebble reaches the corner. By adding two rows, the pebble reaches the corner after one more round-trip exploration, so two more columns are visited (which corresponds to the width of the phase rectangle). When the phase ends, we can observe that two more rows are visited (which corresponds to the height of the phase rectangle). The configuration at the end of the phase is $C_{i+1}$.

□

## 7. Open Questions

*There are several immediate research directions to your work. First, our solution for five synchronous robots under visibility one uses one non-deterministic rule. Is-it possible to remove this non-deterministic rule without increasing the visibility nor the number of robots?*

*Second, what is the optimal number of colors required to solve the IGE problem under visibility range one using an optimal number of synchronous robots, that is, five robots?*

*Finally, long-term open questions concern the scalability of solutions. This may be tackled through several complementary axes: considering asynchronous settings, providing solutions that works with a unfixed number of robots, increasing the number of possible initial configurations, addressing grids in dimension $d > 2$, guaranteeing fault tolerance properties.*

## References


[1] *Ranendu Adhikary, Kaustav Bose, Manash Kumar Kundu, and Buddhadeb Sau. Mutual visibility by asynchronous robots on infinite grid. In* Algorithms for Sensor Systems - 14th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2018, Helsinki, Finland, August 23-24, 2018, Revised Selected Papers*, pages 83–101, 2018.*

[2] *Lélia Blin, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. Exclusive perpetual ring exploration without chirality. In* DISC*, volume 6343, pages 312–327, 2010.*

[3] *François Bonnet, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. Asynchronous exclusive perpetual grid exploration without sense of direction. In Antonio Fernández 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. URL:* `http://www.springerlink.com/content/9l3v424157681707/`.

[4] *Kaustav Bose, Ranendu Adhikary, Manash Kumar Kundu, and Buddhadeb Sau. Arbitrary pattern formation on infinite grid by asyn-*

*chronous oblivious robots. In* WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings*, pages 354–366, 2019.*

*[5] Quentin Bramas, Stéphane Devismes, Anaïs Durand, Pascal Lafourcade, and Anissa Lamani. Optimal asynchronous perpetual grid exploration. In Toshimitsu Masuzawa, Yoshiaki Katayama, Hirotsugu Kakugawa, Junya Nakamura, and Yonghwan Kim, editors,* Stabilization, Safety, and Security of Distributed Systems - 26th International Symposium, SSS 2024, Nagoya, Japan, October 20-22, 2024, Proceedings*, volume 14931 of* Lecture Notes in Computer Science*, pages 89–105. Springer, 2024.* doi:10.1007/978-3-031-74498-3_6.

*[6] Quentin Bramas, Stéphane Devismes, Anaïs Durand, Pascal Lafourcade, and Anissa Lamani. Beedroids: How luminous autonomous swam of UAVs can save the world? In* FUN*, pages 7:1–7:21, 2022.*

*[7] Quentin Bramas, Stéphane Devismes, and Pascal Lafourcade. Infinite grid exploration by disoriented robots. In Chryssis Georgiou and Rupak Majumdar, editors,* Networked Systems - 8th International Conference, NETYS 2020, Marrakech, Morocco, June 3-5, 2020, Proceedings*, volume 12129 of* Lecture Notes in Computer Science*, pages 129–145. Springer, 2020.* doi:10.1007/978-3-030-67087-0_9.

*[8] Quentin Bramas, Stéphane Devismes, and Pascal Lafourcade. Poleless Exploration with Melomaniac Myopic Chameleon Robots: The Animations, January 2020.* doi:10.5281/zenodo.3606387.

*[9] Quentin Bramas, Pascal Lafourcade, and Stéphane Devismes. Optimal exclusive perpetual grid exploration by luminous myopic opaque robots with common chirality.* Theor. Comput. Sci.*, 977:114162, 2023. URL:* https://doi.org/10.1016/j.tcs.2023.114162*,* doi:10.1016/J.TCS.2023.114162.

*[10] Sebastian Brandt, Jara Uitto, and Roger Wattenhofer. A tight lower bound for semi-synchronous collaborative grid exploration. In Ulrich Schmid and Josef Widder, editors,* 32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of* LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl -*

*Leibniz-Zentrum für Informatik, 2018.* doi:10.4230/LIPIcs.DISC.2018.13.

*[11] Serafino Cicerone, Alessia Di Fonso, Gabriele Di Stefano, and Alfredo Navarra. The geodesic mutual visibility problem: Oblivious robots on grids and trees.* Pervasive Mob. Comput.*, 95:101842, 2023. URL:* https://doi.org/10.1016/j.pmcj.2023.101842*,* doi:10.1016/J.PMCJ.2023.101842.

*[12] Shantanu Das, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Masafumi Yamashita. Autonomous mobile robots with lights.* Theor. Comput. Sci.*, 609(P1):171–184, January 2016.* doi:10.1016/j.tcs.2015.09.018.

*[13] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. Enabling ring exploration with myopic oblivious robots. In* IPDPS*, pages 490–499, 2015.*

*[14] Stéphane Devismes, Anissa Lamani, Franck Petit, and Sébastien Tixeuil. Optimal torus exploration by oblivious robots.* Computing*, 101(9):1241–1264, 2019.*

*[15] Stéphane Devismes, Franck Petit, and Sébastien Tixeuil. Optimal probabilistic ring exploration by semi-synchronous oblivious robots.* Theor. Comput. Sci.*, 498:10–27, 2013.*

*[16] Stéphane Devismes, Anissa Lamani, Franck Petit, Pascal Raymond, and Sébastien Tixeuil. Terminating Exploration Of A Grid By An Optimal Number Of Asynchronous Oblivious Robots.* The Computer Journal*, 03 2020.* doi:10.1093/comjnl/bxz166.

*[17] Durjoy Dutta, Tandrima Dey, and Sruti Gan Chaudhuri. Gathering multiple robots in a ring and an infinite grid. In* Distributed Computing and Internet Technology - 13th International Conference, ICDCIT 2017, Bhubaneswar, India, January 13-16, 2017, Proceedings*, pages 15–26, 2017.*

*[18] Yuval Emek, Tobias Langner, David Stolz, Jara Uitto, and Roger Wattenhofer. How many ants does it take to find the food?* Theor. Comput. Sci.*, 608(P3):255–267, December 2015.* doi:10.1016/j.tcs.2015.05.054.

[19] *Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots.* Theor. Comput. Sci.*, 411(14-15):1583–1598, 2010.*

[20] *Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. How many oblivious robots can explore a line.* Inf. Process. Lett.*, 111(20):1027–1031, 2011.*

[21] *Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots.* Algorithmica*, 65(3):562–583, 2013.*

[22] *Giuseppe Antonio Di Luna, Paola Flocchini, Sruti Gan Chaudhuri, Federico Poloni, Nicola Santoro, and Giovanni Viglietta. Mutual visibility by luminous robots without collisions.* Inf. Comput.*, 254:392–418, 2017.* doi:10.1016/j.ic.2016.09.005.

[23] *Fukuhito Ooshita and Ajoy K. Datta. Brief announcement: Feasibility of weak gathering in connected-over-time dynamic rings. In* Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018, Tokyo, Japan, November 4-7, 2018, Proceedings*, pages 393–397, 2018.*

[24] *Fukuhito Ooshita and Sébastien Tixeuil. Ring exploration with myopic luminous robots. In* SSS*, pages 301–316, 2018.*

[25] *David Peleg. Distributed coordination algorithms for mobile robot swarms: New directions and challenges. In* Proceedings of the 7th International Conference on Distributed Computing*, IWDC'05, pages 1–12, Berlin, Heidelberg, 2005. Springer-Verlag. URL:* http://dx.doi.org/10.1007/11603771_1*,* doi:10.1007/11603771_1.

[26] *Arthur Rauch, Quentin Bramas, Stéphane Devismes, Pascal Lafourcade, and Anissa Lamani. Optimal exclusive perpetual grid exploration by luminous myopic robots without common chirality. In Karima Echihabi and Roland Meyer, editors,* Networked Systems - 9th International Conference, NETYS 2021, Virtual Event, May 19-21, 2021, Proceedings*, volume 12754 of* Lecture Notes in Computer Science*, pages 95–110. Springer, 2021.* doi:10.1007/978-3-030-91014-3_7.

[27] *Gabriele Di Stefano and Alfredo Navarra. Gathering of oblivious robots on infinite grids with minimum traveled distance.* Inf. Comput., *254:377–391, 2017.* doi:10.1016/j.ic.2016.09.004.

[28] *Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns.* SIAM J. Comput., *28(4):1347–1363, 1999.* doi:10.1137/S009753979628292X.

[29] *Ichiro Suzuki and Masafumi Yamashita. Erratum: Distributed anonymous mobile robots: Formation of geometric patterns.* SIAM J. Comput., *36(1):279–280, 2006.* doi:10.1137/050631562.