Finding Water on Poleless Using Melomaniac Myopic Chameleon Robots

Quentin Bramas
University of Strasbourg, ICUBE, France
bramas@unistra.fr

Pascal Lafourcade
LIMOS, University Clermont Auvergne, Aubière, France
pascal.lafourcade@uca.fr

Stéphane Devismes
Université Grenoble Alpes, VERIMAG, France
Stephane.Devismes@univ-grenoble-alpes.fr

Abstract

In 2042, the exoplanet exploration program,[1] launched in 2014 by NASA, finally discovers a new exoplanet so-called *Poleless*, due to the fact that it is not subject to any magnetism. A new generation of autonomous mobile robots, called M2C (for Melomaniac Myopic Chameleon), have been designed to find water on Poleless. To address this problem, we investigate optimal (*w.r.t.*, visibility range and number of used colors) solutions to the infinite grid exploration problem (IGE) by a small team of M2C robots. Our first result shows that minimizing the visibility range and the number of used colors are two orthogonal issues: it is impossible to design a solution to the IGE 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. It is worth noticing that these two algorithms use a very small number of robots, respectively six and eight.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Luminous Robots, Grid, Infinite Exploration, Treasure Search Problem

Digital Object Identifier 10.4230/LIPIcs.FUN.2021.6

Supplementary Material `https://doi.org/10.5281/zenodo.3606387`

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

1 Introduction

Poleless is a far-off exoplanet discovered in 2042 that is not subject to any magnetism. Hence, all terrestrial compasses are ineffective on this planet. After the discovery of Poleless, the *National Aeronautics and Space Administration (NASA)* has decided to launch a robotic spacecraft mission toward it in order to evaluate the possibility of a future human presence. For this purpose, they have designed a new generation of autonomous mobile robots called *M2C*, for Melomaniac Myopic Chameleon. These robots are *melomaniac*: in order to synchronously move and to easily coordinate their actions they continuously play and listen the same melody. Of course, the choice of the right song was critical. After a huge campaign of experiments, an international expert panel (notably including several Nobel prizes) has selected the song "Heigh-ho" of the Seven Dwarfs.[2] The M2C robots are also *myopic*, i.e.,

[1] `https://exoplanets.nasa.gov/`
[2] See `https://www.youtube.com/watch?v=HI0x0KYChq4`

10th International Conference on Fun with Algorithms (FUN 2021).
Editors: Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara; Article No. 6; pp. 6:1–6:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

they are equipped with visibility sensors of typically small range. The choice of this feature has been led by several concerns, mainly reducing both the manufacturing costs (still in 2042, NASA has to endure important cuts in its budget) and the energy consumption (of course, no human intervention can be envisioned to recharge their batteries). Additionally, the researchers thought to these two technologies (myopia and melomania) to make the design of the robots as simple as possible. Indeed, simplicity usually implies more robustness and allows to decrease the weight of robots by avoiding the use of fancy, heavy, and costly components. Finally, the M2C robots are *chameleons* meaning that they have the ability to change their color whenever they want. These colors are used for two main reasons:

1. robot intercommunication, since the colors are captured by the visibility sensor of others robots in their surroundings, and
2. persistent memory; actually, colors are the only available persistent memory.

Notice since any modulation in the melody playing may cause an irreversible disynchronization, robots can only use their colors to exchange information.

By analyzing the light spectrum of Poleless, researchers have established the presence of water and a breathable atmosphere with high probability. However, in order to be sure of these facts, an exploration mission is mandatory. Once robots will have landed on the planet, it will be easy for them to test the chemical composition of the atmosphere. Now, to confirm the presence of water, they will have to explore exhaustively the ground of Poleless. Especially since the second important goal of the mission is to find an appropriate place, near a water source, where a future human mission could land. Again, for cost issues, only a typically small team of M2C robots can be used to achieve this task. Basically, they have to coordinate together to explore the planet until (at least) one of them find water. Once it will happen, the robot will both stop moving and singing (informing then the others of the task completion so that they all stop in turn), and send a signal to Earth in order to be precisely localized.

The exact size and the relief of Poleless is unknown, even if it seems to be quite flat. Hence, the surface of Poleless is conveniently discretized as grid of *unbounded* size, where nodes represent locations that can be sensed by robots and edges represent the possibility for a robot to move from one location to another. Hence, the task to be solved by the team of robots is the *treasure search problem* in a grid of unbounded size [11]. Now, this problem is known to be equivalent to the *Infinite Grid Exploration* (IGE) problem [7], which requires each node of an *infinite* grid to be visited within finite time by at least one robot.

We have decided to answer the NASA call for bids by designing new solutions to the IGE problem that are well-suited to the NASA requirements. Notice, in particular, that despite the scientific progress, physical boundaries are still in the agenda in 2042. Namely, the M2C robots are opaque, i.e., a robot is able to see another robot if and only if no other robot lies in the line segment joining them. Moreover, any solution to the IGE problem should achieve *exclusiveness* [2], meaning that, in the grid, any two robots cannot simultaneously occupy the same node nor traverse the same edge. Indeed, even if in 2042 holograms and teleportation techniques are commonly used, they are still not mature enough to be used in a long distance spacecraft mission.

1.1 Contribution

We address the NASA call for bids by investigating low-cost solutions to the IGE problem, i.e., we try as much as possible to limit the necessary visibility range, the number of used colors, and the size of the team.

We first show that minimizing the visibility range and the number of used colors are two orthogonal issues: it is impossible to design an algorithm solving the IGE problem that does not use different colors and that assumes visibility range 1.

Hence, we address the optimality of these two criteria separately. Precisely, we provide two algorithms for the IGE problem:

1. The first algorithm uses only six M2C robots with visibility range 1 and three colors.
2. The second algorithm uses only eight M2C robots with visibility range 2, yet no color (in other word, it assumes *oblivious anonymous* robots).

1.2 Roadmap

In the next section, we define the model associated to M2C robots and Poleless. In Section 3, we present our impossibility result. In Section 4, we describe our two algorithms. Section 5 is dedicated to related work. We conclude in the last section.

2 Preliminaries

As justified in the introduction, we model the ground of Poleless by 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.

We assume a team $\mathcal{R}$ of $n > 0$ M2C robots evolving on (nodes of) the grid. Recall that M2C robots are melomaniac, i.e., they compute and move *synchronously* by continuously singing Heigh-ho. Precisely, at each beat (or *round*), they all perform an atomic *dance step* as follows. First, they look at their surroundings. Then, they compute a destination among their current position and the four neighboring ones. Finally, they move to the computed destination.

M2C robots are chameleons, i.e., they may change their color, which can be seen by other robots in their surroundings. Let Cl be the set of possible colors. Recall that robots cannot simultaneously occupy the same node nor traverse the same edge. 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 either the color of the robot located at this node, if it is occupied, or $\perp$ otherwise. When a robot looks around, it can only see the states of the node that are within distance $\Phi \in \mathbb{N}^*$ from its position. Φ is called the *visibility range* of the robots. The value of Φ depends on the quality (and so the price) of the robots' lenses. If the visibility range is one, a robot sees its own location and the four neighboring nodes of the grid. After looking around, a robot computes the next destination based only on what it sees and on its own color. During the compute phase, a robot may also decide to change its color.

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 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: when we say "the robot moves Up", it is according to the globally oriented view. However, since M2C robots are designed to explore Poleless, they do not have any compass and so, 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), $\pi/2$, 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.

Moreover, since robots may obstruct visibility, the function that removes the state of a node u if there is another robot between u and r is *systematically* applied to obtain the local view. 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).

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 Poleless Exploration

An algorithm $\mathcal{A}$ solves the *Poleless exploration* 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.

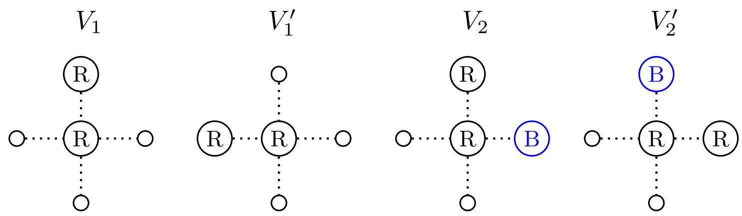

Figure 1 Example of four views. V_1 and V_1' are indistinguishable. Similarly, V_2 and V_2' are indistinguishable.

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 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 algorithms we will propose will be well-defined. However, to be as general as possible, we will not make such an assumption in our impossibility results. Finally, remark that a well-defined algorithm has a unique execution.

2.7 Notations

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

3 Impossibility Result

3.1 The Fence Crossing Lemma

In order to explore Poleless, M2C robots 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 stays 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); and (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 robots explore Poleless, then there is a fence that is *single-handed crossed* by a subset of robots; see Lemma 1. This latter result will be used to show that, if robots are anonymous and cannot change their color, the Poleless exploration is impossible under visibility range 1, whatever the number of robots is; see Theorem 2.

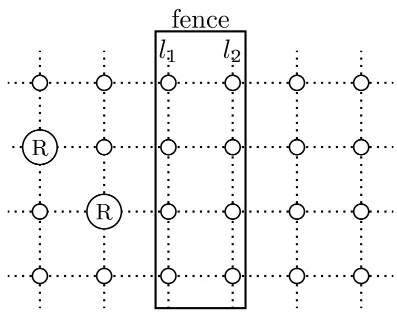

Figure 2 A team of robots in front of a fence.

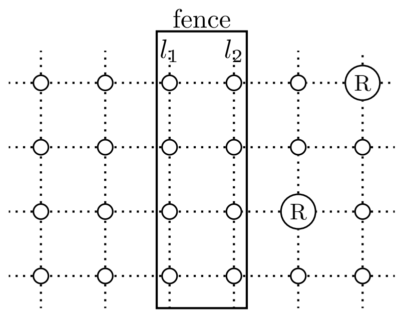

Figure 3 A team of robots has crossed a fence.

► **Lemma 1** (The test of the fence). *If n robots can explore Poleless, then in every execution there exists a fence L and a subset of robots S such that S single-handed crosses L within a finite number of rounds.*

Proof. If n robots successfully explore Poleless, 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. If there is an execution where no subset of robots single-handed crosses at least one of them, then this means that each time a fence is crossed in the execution, some robots are not crossing and are left behind. If i is large enough ($i > n$), then there is not enough robots to cross all the fences to reach u. Hence, a subset of robots single-handed crosses a fence in a finite number of rounds. ◀

3.2 The Impossibility Result

With only one color and under visibility one, at each round there is at most six possible (local) 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. 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 center robot, 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.

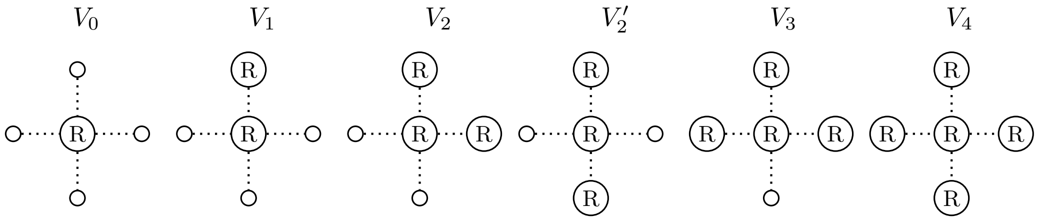

Figure 4 The possible views of a robot with visibility one and without colors.

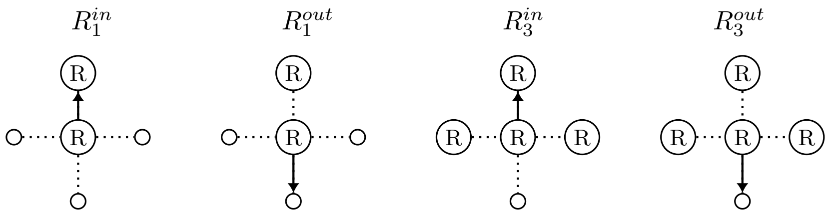

Figure 5 Non-ambiguous and non-idle rules, with visibility one and no color.

▶ **Theorem 2.** *There is no algorithm that solves the Poleless exploration problem with single-color robots and assuming visibility range one.*

Proof. Assume, by contradiction, that an algorithm $\mathcal{A}$ solves the Poleless exploration problem with single-color 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}.

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 then 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 or (c) r_d executes Rule R_3^{out}. We now show that in all theses cases, we either obtain a contradiction, or we show that there are two adjacent robots r_a and r'_a (of S) on l_2 at Round $t_1 + 1$.

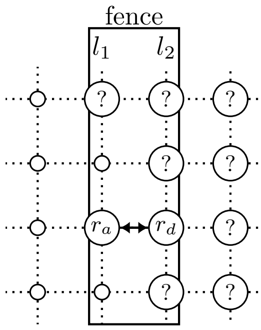

Figure 6 Case (a), reaching a contradiction.

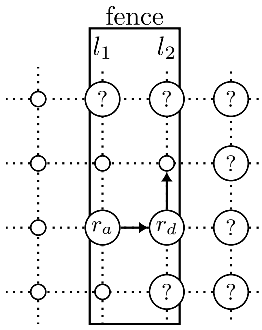

Figure 7 Case (b), r_a and r_d are neighbors at Round $t_1 + 1$.

- 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 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. So, 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 $t_1 + 1$.

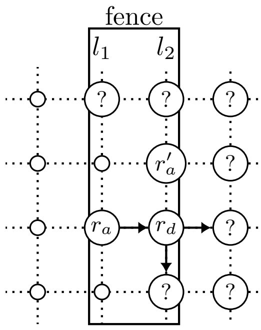

Figure 8 Case (c), r_a and r'_a are neighbors at Round $t_1 + 1$.

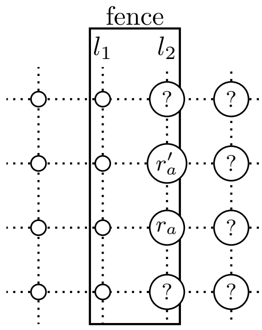

Figure 9 Robots r_a and r'_a are stuck on the fence.

- 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 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, or (iii) stays idle. If r_a executes R_1^{in} or an ambiguous rule towards an occupied destination, it moves UP towards r'_a. 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 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 Algorithms

In this section, we give two algorithms, respectively called $\mathcal{A}^1_{(6,3)}$ and $\mathcal{A}^2_{(8,1)}$, for solving the Poleless exploration. Algorithm $\mathcal{A}^1_{(6,3)}$ (presented in Subsection 4.1) assumes visibility one and uses six robots and three colors. Algorithm $\mathcal{A}^2_{(8,1)}$ requires visibility two and uses eight anonymous oblivious robots, i.e., indistinguishable robots. The animations of these two algorithms are available in our complementary material [6]. The fact that the rules of these algorithms are well-defined has been checked by the script that generated those animations. This has been done by making sure that (1) the view of any rule cannot be transformed into the view of another rule using mirroring, rotation, or a combination of the two, and (2) for each rule, the global destination does not depend on the applied local indistinguishable transformation.

4.1 Six Robots with Three Colors under Visibility Range One

The six robots are divided into two categories: the *beacon* robots and the *moving group*. There are four beacon robots, each of those being B-colored in the following. The *moving group* is made of two robots: one L-colored *leader* and one F-colored *follower*. However, some robots change their role (by changing their color) along the execution.

Initially, all the robots are close together and organized as shown in Figure 10. The beacons are used to delimit the area which has been already explored. The moving group aims at reaching the beacons one by one. Each time the moving group reaches a beacon,

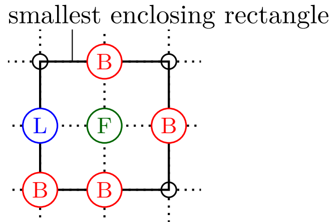

Figure 10 Initial configuration of Algorithm $\mathcal{A}^{1}_{(6,3)}$.

robots make an *adjustment*. At the end of the adjustment, the new beacon position is in the diagonal (two hops) of the previous one and the moving group has made a turn toward the next beacon. This adjustment, in particular, allows to take the newly explored nodes into account. The moving group then continues toward the next beacon, and so on. Each time the moving group comes back to the first beacon, a so-called *phase* terminates: the border of the area initially delimited by the four beacons is now fully visited, and the area newly delimited by the beacons is bigger; see Figure 11 to visualize the increasing area that is explored by the moving group.

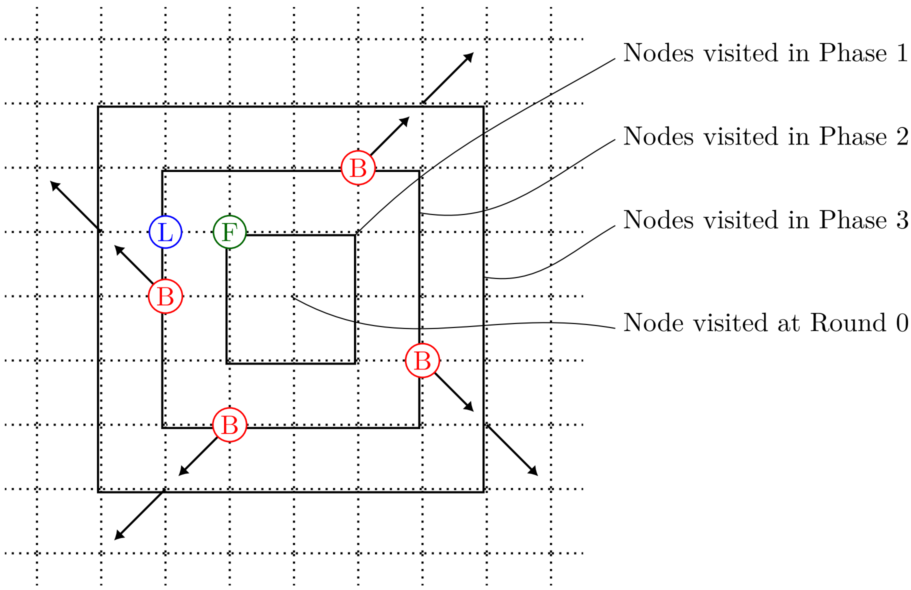

Figure 11 Visited area after the first three phases for $\mathcal{A}^{1}_{(6,3)}$. The positions of the robots at those at the beginning of the second phase.

During an *adjustment*, the leader becomes the beacon, the beacon becomes the leader, and after that, the new moving group travels toward the next beacon. Figure 15 shows the sequence of moves of an adjustments occurring at the top-right beacon.

The moving group successfully performs a phase independently of the distance between the beacons, so that infinitely many growing phases are achieved in sequence. The Poleless exploration problem is then solved as any node of the grid is eventually included in the area delimited by the beacons. Note that we use the same technique for the second algorithm.

The rules that allow the moving group to travel along a straight line are shown in Figure 12. The rules to make an adjustment are shown in Figure 13. In order for the two first rounds to work as expected, three more rules (Figure 14) are necessary. Those rules are similar to the previous rules, but consider cases where more robots appear in view. This

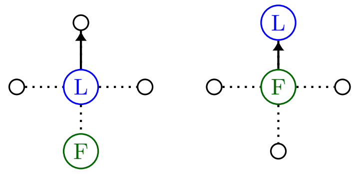

Figure 12 The two rules that make the moving group travel along a straight line.

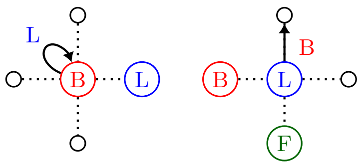

Figure 13 The two rules that perform an adjustment.

actually occurs at the beginning of the algorithm only, when all the robots are close together. For instance, the follower robot should move towards the leader, even if it sees beacon robots around it.

▶ **Theorem 3.** *Algorithm $\mathcal{A}^{1}_{(6,3)}$ solves the Poleless exploration problem using six robots, three 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 beacon robot is located in the initial configuration; see Figure 10. Recall that these global coordinates are used for the analysis only: robots cannot access those coordinates.

Using this coordinate system, the initial configuration is denoted C^0 and is decomposed as follow: $C^0 = \{M^0, C_0^0, C_1^0, C_2^0, C_3^0\}$, where $M^0 = \{((0,1), L), ((1,1), F)\}$, $C_0^0 = \{((0,0), B)\}$, $C_1^0 = \{((1,0), B)\}$, $C_2^0 = \{((2,1), B)\}$, and $C_3^0 = \{((1,2), B)\}$. 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 in the initial configuration. We now prove that starting from Configuration C^i, Configuration C^{i+1} is eventually reached. Since the initial 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 rectangles. The illustration of one cycle is presented in Figure 16.

Assume we reach the first configuration C^i of Phase i at time t. Recall that $C^i = \{((-i, i+1), L), ((1-i, i+1), F), C_0^i, C_1^i, C_2^i, C_3^i\}$. After one round, the configuration is $\{((-i, i), L), ((-i, i+1), F), C_0^{i+1}, C_1^i, C_2^i, C_3^i\}$.

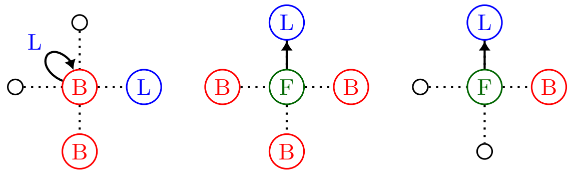

Figure 14 The three rules similar the the previous rules, but used at the beginning of the algorithm, when the robots close together.

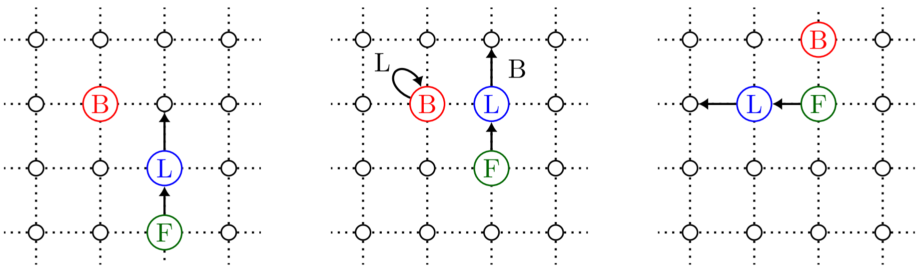

Figure 15 Sequence of moves for an adjustment at the top left beacon robot.

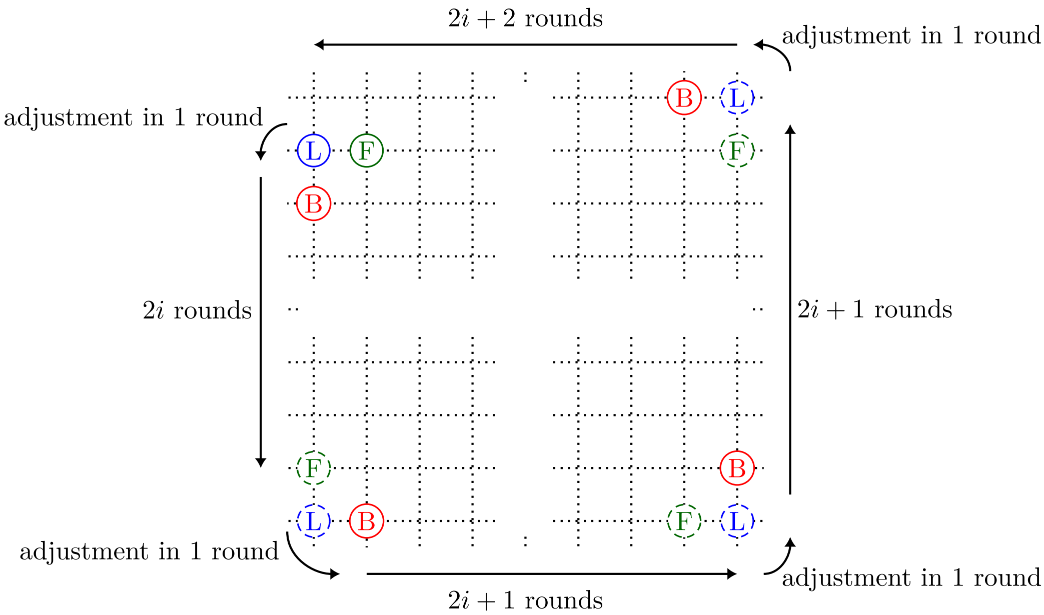

Figure 16 Visualization of one phase.

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

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

When the robot with color L sees the third beacon, the third adjustment occurs and the reached configuration is $\{((2+i,1-i),L),((2+i,-i),F),C_0^{i+1}, C_1^{i+1},C_2^{i+1},C_3^i\}$.

Then, the moving group travels during $2i+1$ rounds until the robot with Color L sees the fourth beacon robot. The last adjustment is performed to obtain the configuration $\{((1+i,2+i),L),((2+i,2+i),F),C_0^{i+1}, C_1^{i+1},C_2^{i+1},C_3^{i+1}\}$. 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 $t+8i+5$.

Inductively, the robots start from configuration C^0 and reach configuration C^i within finite time, for any $i \geq 0$. Also, Node $(1,1)$ is visited at Round 0, and the set V_i of nodes visited by the robot with Color L between Phase i and $i+1$ contains the edges of the rectangle $\{\vec{t}_{(-i,-i)}(0,0), \vec{t}_{(i,-i)}(2,0), \vec{t}_{(i,i)}(2,2), \vec{t}_{(-i,i)}(0,2)\}$; see Figure 11. Since $\{(1,1)\} \cup \bigcup_{i\geq 0} V_i = \mathbb{Z} \times \mathbb{Z}$, Algorithm $\mathcal{A}^{1}_{(6,3)}$ solves the Poleless exploration problem. ◀

4.2 Eight Anonymous Oblivious Robots under Visibility Two

Algorithm $\mathcal{A}^{2}_{(8,1)}$ is based on principles similar to those used in Algorithm $\mathcal{A}^{1}_{(6,3)}$: four *beacon* robots delimit the visited area, and there is a *moving group*, this time made of four robots, to travel from one beacon to another. The initial configuration of $\mathcal{A}^{2}_{(8,1)}$ is described in Figure 17. Since robots are anonymous, the only way to distinguish them is to use their relative locations. Notice that, this time, beacon robots are always the same. To maintain this property, we ensures that beacon robots are never adjacent to any other robot. Since the visibility range is two, a beacon can see other robots and move before becoming their neighbor.

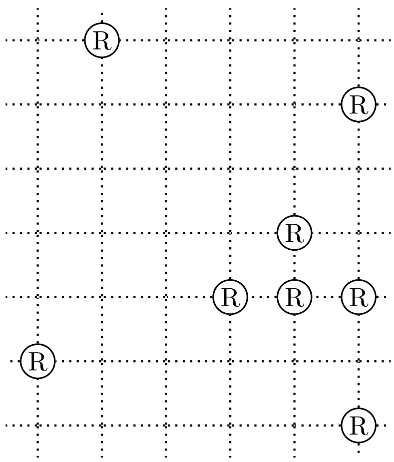

Figure 17 Initial configuration I of $\mathcal{A}^{2}_{(8,1)}$.

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 $\mathcal{A}^{2}_{(8,1)}$ 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 18.

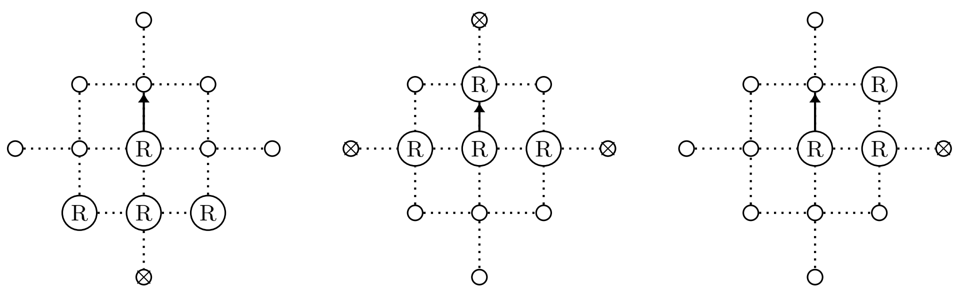

Figure 18 Rules to move along a straight line.

The first three rules (see Figure 18) 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.

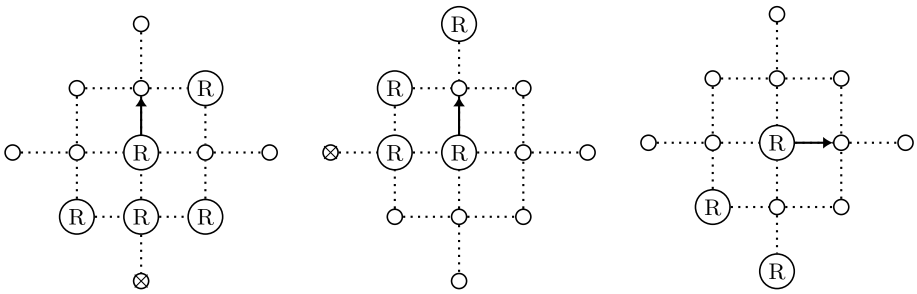

Figure 19 Rules to make the spaceship moving along a straight line when seeing the beacon, and to make the beacon move away.

Then, when the moving group meets a beacon robot, an adjustment is made in two rounds. The first round of the adjustment, robots execute the rules defined in Figure 19. The first two rules order 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 20 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. After the execution of those rules, the spaceship shape is preserved, but the bow robot has become a side robot, 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 19) 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 21 is necessary to make moving as expected the side robot that still sees the beacon robot right after an adjustment.

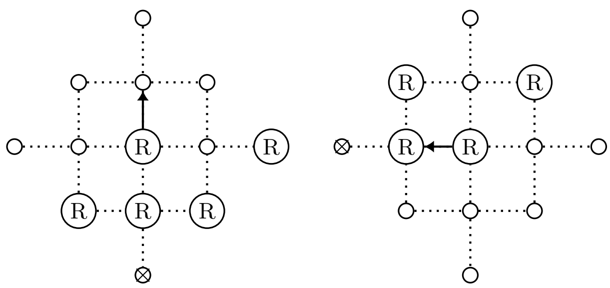

Figure 20 The leader moves in straight line again to become a side follower, the side follower that sees the beacon moves left (the beacon move away again using the same rule as before).

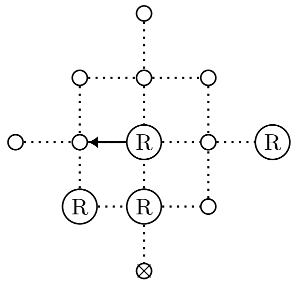

Figure 21 The moving group moves away from the beacon. The side follower that still see the beacon moves away from it.

▶ **Theorem 4.** *Algorithm $\mathcal{A}^2_{(8,1)}$ solves the exclusive Poleless exploration problem using eight robots without color and visibility range of two.*

Proof. The proof of this theorem is similar to the proof of Theorem 4: we decompose the execution into phases, show by induction that each phase is eventually reached, and finally a particular rectangle is visited during each phase.

We fix a global coordinate system, not accessible to the robots, where the initial configuration, denoted by C^0, is split as follows: $C^0 = \{M^0, C^0_0, C^0_1, C^0_2, C^0_3\}$, where $M^0 = \{(3,2),(4,2),(5,2),(4,3)\}$, $C^0_0 = \{(5,5)\}$, $C^0_1 = \{(1,6)\}$, $C^0_2 = \{(0,1)\}$, and $C^0_3 = \{(5,0)\}$. We define the configuration $C^i = \{M^i, C^i_0, C^i_1, C^i_2, C^i_3\}$ in Phase i, where $M^i = \vec{t}_{(i,i)}(M^0)$, $C^i_0 = \vec{t}_{(i,i)}(C^0_0)$, $C^i_1 = \vec{t}_{(-i,i)}(C^0_1)$, $C^i_2 = \vec{t}_{(-i,-i)}(C^0_2)$, and $C^i_3 = \vec{t}_{(i,-i)}(C^0_3)$.

Here, C^i_0 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 $C^i = \{(i+3,i+2),(i+4,i+2),(i+5,i+2),(i+4,i+3), C^i_0, C^i_1, C^i_2, C^i_3\}$. After three rounds, the configuration is $\{(i+3,i+5),(i+4,i+4),(i+4,i+5),(i+4,i+6), C^{i+1}_0, C^i_1, C^i_2, C^i_3\}$.

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 $\{(1-i,i+4),(-i,i+5),(1-i,i+5),(2-i,i+5), C^{i+1}_0, C^{i+1}_1, C^i_2, C^i_3\}$. 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 $\{\vec{t}_{(-i,-i)}(0,1), \vec{t}_{(i,-i)}(5,1), \vec{t}_{(i,i)}(5,5), \vec{t}_{(-i,i)}(0,5)\}$. 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 Poleless exploration problem. ◀

5 Related Work

The robots we have considered are known as *luminous robots* in the literature. They have been introduced by Peleg in [14]. In [8], the authors compare the computational power of luminous robots with respect to the three main execution models: fully-synchronous, semi-synchronous, and asynchronous. Solutions for dedicated problems such as *weak gathering* or *mutual visibility* have been respectively investigated in [12] and [13].

Exploration tasks have been first considered in the context of finite graphs. In this setting, 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 [9], authors solve terminating exploration of any finite grid using few asynchronous anonymous oblivious robots, yet assuming unbounded visibility range. The exclusive perpetual exploration of a finite grid is considered in the same model in [3].

Various terminating problems have been investigated in infinite grids such as *arbitrary pattern formation* [4], *mutual visibility* [1], and *gathering* [15, 10].

Emek et al. [11] have investigated the treasure search problem in an unbounded size grid [7]. They consider robots operating in two models: the semi-synchronous and 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 synchronous and four semi-synchronous robots. Moreover, they exclude solutions for two robots.

In a followup paper [7], 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 [5], we have investigated the IGE problem by a swarm of autonomous mobile luminous synchronous 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 non-modifiable 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, with 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 identical robots without light.

6 Conclusion

Thanks to our impossibility results, NASA has been convinced that our two algorithms with Melomaniac Myopic Chameleon Robots are relevant to explore Poleless. Due to some restrictions on the budget, they finally decided to use robots with only a visibility range of one. Then, in 2048, the mission “Finding Water on Poleless” was a great success, and fortunately thanks to our algorithm water was found on Poleless. Once this great news has been known, NASA realizes that the time to send some humans to Poleless the water might have change the place. Hence, they ask a different challenge to the community with the perpetual exploration of Poleless, in order to regularly update information about water localizations With this new challenge our roadmap is clear for the next months.

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 Roberto Baldoni, François Bonnet, Alessia Milani, and Michel Raynal. Anonymous graph exploration without collision by mobile robots. *Inf. Process. Lett.*, 109(2):98–103, 2008. doi:10.1016/j.ipl.2008.08.011.

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 asynchronous 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, and Pascal Lafourcade. Infinite grid exploration by disoriented robots. In Michele Flammini Keren Censor-Hillel, editor, *Structural Information and Communication Complexity - 26th International Colloquium, SIROCCO 2019, L'Aquila, Italy, July 1-4, 2019, Proceedings*, volume 11639 of *Lecture Notes in Computer Science*, pages 340–344. Springer, 2019. doi:10.1007/978-3-030-24922-9_25.

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

7 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.

8 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.

9 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*, March 2020. doi:10.1093/comjnl/bxz166.

10 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.

11 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.

12 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.

13 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.

14 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. `doi:10.1007/11603771_1`.

15 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`.