Robot Searching and Gathering on Rings under Minimal Assumptions

Gianlorenzo d'Angelo, Alfredo Navarra, Nicolas Nisse

Robot Searching and Gathering on Rings under Minimal Assumptions

Gianlorenzo D'Angelo, Alfredo Navarra, Nicolas Nisse

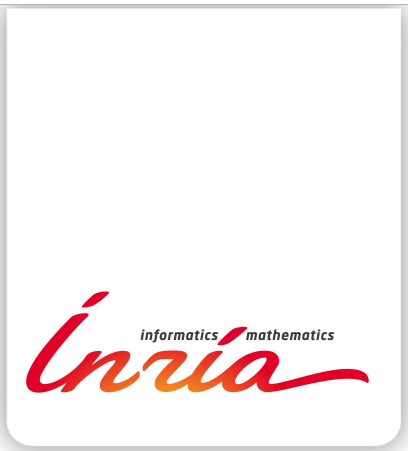
RESEARCH REPORT

N° 8250

February 2013

Project-Teams COATI

ISSN 0249-6399 ISRN INRIA/RR--8250--FR+ENG



Robot Searching and Gathering on Rings under Minimal Assumptions*

Gianlorenzo D'Angelo, Alfredo Navarra, Nicolas Nisse

Project-Teams COATI

Research Report n° 8250 — February 2013 — 30 pages

Abstract: Consider a set of mobile robots with minimal capabilities placed over distinct nodes of a discrete anonymous ring. They operate on the basis of the so called *Look-Compute-Move* cycle. Asynchronously, each robot takes a snapshot of the ring, determining which nodes are either occupied by robots or empty. Based on the observed configuration, it decides whether to move to one of its adjacent nodes or to stay idle. In the first case, it performs the computed move, eventually. The computation also depends on the required task. In this paper, we solve both the well-known *Searching* and *Gathering* tasks. In the literature, most contributions are restricted to a subset of initial configurations. Here, we design two different algorithms and provide a full characterization of the initial configurations that permit the resolution of the problems under minimal assumptions.

Key-words: Distributed Computing; CORDA model; Gathering; Graph Searching

*

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Réunion et Nettoyage par des Robots dans un anneau

Résumé : Nous considérons un ensemble de robots mobiles qui sont placés sur distincts sommets d'un réseau en anneau. Le réseau est anonyme et les robots ont des aptitudes minimales. Ils opèrent par des cycles *Observer-Calculer-Bouger*. Nous résolvons les problèmes de la réunion et du nettoyage de graphe dans ce modèle.

Mots-clés : Cacul distribué; Modèle CORDA; Réunion; Nettoyage de graphe

1 Introduction

In the field of robot-based computing systems, the study of the minimal settings required to accomplish specific tasks represents a challenging issue. We consider k robots initially placed on n distinct nodes of a discrete ring, and we solve two fundamental problems: the *Searching* (see, e.g., [**?**, **?**, **?**]) and the *Gathering* (see, e.g., [**?**, **?**, **?**, **?**, **?**]). Robots are assumed to be oblivious (without memory of the past), uniform (running the same deterministic algorithm), autonomous (without a common coordinate system, identities or chirality), asynchronous (without central coordination), without the capability to communicate. Neither nodes nor edges are labeled and no local memory is available on nodes.

The relevance of the ring topology is motivated by its completely symmetric structure. It means that algorithms for rings are more difficult to devise as they cannot exploit any topological structure, assuming that all nodes look the same. In fact, our algorithms are only based on robots' disposal and not on topology. Robots are equipped with visibility sensors and motion actuators, and operate in *Look-Compute-Move* cycles in order to achieve a common task (see [**?**]). The Look-Compute-Move model considers that in each cycle a robot takes a snapshot of the current global configuration (Look), then, based on the perceived configuration, takes a decision to stay idle or to move to one of its adjacent nodes (Compute), and in the latter case it moves to this neighbor (Move). Cycles are performed asynchronously, i.e., the time between Look, Compute, and Move operations is finite but unbounded, and it is decided by an adversary for each robot. Hence, robots that cannot communicate may move based on outdated perceptions. The scheduler that determines the timing of the Look-Compute-Move cycles is assumed to be fair, that is, each robot performs its cycle within finite time and infinitely often. The inaccuracy of the sensors used by robots to scan the surrounding environment motivates its discretization. Moreover, robots can model software agents moving on a computer network.

Initial configurations are assumed to be *exclusive*, that is, any node is occupied by at most one robot. On rings, different types of exclusive configurations may require different approaches. In particular, periodicity and symmetry arguments must be carefully handled. An exclusive configuration is called *periodic* if it is invariable under non-complete rotations. It is called *symmetric* if the ring has an *axis of symmetry* that reflects single robots into single robots, and empty nodes into empty nodes. It is called *rigid* if it is aperiodic and asymmetric.

Searching: Graph searching has been widely studied in centralized [**?**] and distributed settings (e.g., [**?**]). The aim is to make the robots clear all the edges of a contaminated graph. An edge is cleared if it is traversed by a robot or if both its endpoints are occupied. However, a clear edge can be recontaminated if there is a path without robots from a contaminated edge to it. A graph is *searched* if all its edges are cleared. The searching is *perpetual* if it is accomplished infinitely many time. In this paper (and following [**?**, **?**]), we also consider an additional constraint: the so called *exclusivity property*, that is, no two robots can be concurrently on the same node or cross the same edge. The study of perpetual graph searching in the discrete model provided here when the exclusivity property must be always satisfied has been introduced in [**?**] for tree topologies. Concerning rings, in [**?**] the case of rigid configurations has been tackled.

Gathering: The gathering problem consists in moving all the robots towards the same node and remain there, that is a sort of complementary requirement with respect to the exclusivity property. On rings, under the Look-Compute-Move model, the gathering is unsolvable if the robots are not empowered by the so-called *multiplicity detection* capability [**?**], either in its *global* or *local* version. In the former type, a robot is able to perceive whether any node of the graph is occupied by a single robot or more than one (i.e., a *multiplicity* occurs) without perceiving the exact number. In the latter type, a robot is able to perceive the multiplicity only if it is part of it. Using the global multiplicity detection capability, in [**?**] some impossibility results

have been proven. In detail, if the initial disposal of the robots is periodic, or it has an axis of symmetry passing through two edges, or there are only two robots, the gathering is unsolvable. Then, several algorithms have been proposed for different kinds of exclusive initial configurations in [**?**, **?**, **?**]. These papers left open some cases which have been closed in [**?**] where a unified strategy has been provided. With local multiplicity detection capability, an algorithm starting from rigid configurations where the number of robots k is strictly smaller than $\lfloor \frac{n}{2} \rfloor$ has been designed in [**?**]. In [**?**], the case where k is odd and strictly smaller than $n-3$ has been solved. In [**?**], the authors provide an algorithm for the case where n is odd, k is even, and $10 \leq k \leq n-5$. Recently, the case of rigid configurations has been solved in [**?**]. The remaining cases are left open and the design of a unified algorithm is still required.

Contribution: We provide two distinct and complementary algorithms based on an initial common phase for solving both the perpetual searching and the gathering tasks. The perpetual searching is solved starting from any exclusive and aperiodic configuration with $4 < k \leq n-3$ robots but for the cases with $n = 10$ and $k \in \{5, 6\}$, or with k even and an axis of symmetry passing through an empty node. We prove that in this latter case the problem is unsolvable. Moreover, in [**?**] it has been shown that it is unsolvable also for $k \leq 3$ and $k \geq n-2$. Therefore, we provide a full characterization of any aperiodic initial configuration with $k \neq 4$, and $(n, k) \notin \{(10, 5), (10, 6)\}$. For the gathering, our algorithm solves the problem starting from any exclusive configuration with $3 \leq k < n-4$, $k \neq 4$, robots empowered by the local multiplicity detection, but for the unsolvable configurations provided in [**?**].

Due to space constraints, most of the proofs of the lemmata and theorems and the pseudo-codes of the proposed algorithms are reported in Appendix.

2 Notation and preliminary

Consider a ring with n nodes $\{v_0, \cdots, v_{n-1}\}$ and $3 \leq k < n-2$, $k \neq 4$, robots standing at distinct nodes. A *configuration* $\mathcal{C}$ is defined by the k nodes occupied by robots. For any $i \leq n$, let $\mathcal{S}_i = (r_1^i, \cdots, r_n^i) \in \{0,1\}^n$ be the sequence such that $r_j^i = 0$ if $v_{i+j \bmod n}$ is occupied and $r_j^i = 1$ otherwise, $1 \leq j \leq n$. Intuitively, $\mathcal{S}_i$ represents the positions of robots, starting at v_i. For any sequence $X = (x_0, \cdots, x_r)$, let us denote $\overline{X} = (x_r, \cdots, x_0)$ and $X_i = (x_{i \bmod r+1}, \ldots, x_{r+i \bmod r+1})$. A *representation* of $\mathcal{C}$ is any sequence in $\mathcal{S}_\mathcal{C} = \{\mathcal{S}_i, \overline{(\mathcal{S}_i)}\}_{i \leq n}$. Abusing the notation, we say $\mathcal{C} = S$ for any $S \in \mathcal{S}_\mathcal{C}$. Note that, for any $S = (s_1, \cdots, s_n) \in \mathcal{S}_\mathcal{C}$, $\sum_{i \leq n} s_i = n - k$. A *supermin* of $\mathcal{C}$ is any representation of $\mathcal{C}$ that is minimum in the lexicographical order. We denote the supermin of $\mathcal{C}$ as $\mathcal{C}^{\min}$. Note that, in any supermin $(s_1, \cdots, s_n)$, $s_n = 1$ since $k < n$. We denote by x^h a sequence of $h \geq 0$ consecutive x, $x \in \{0,1\}$. We say that a sequence X is *palindrome* if $X = \overline{X}$ and it is *symmetric* if X_i is palindrome for some i. A configuration is symmetric if at least one of its representations is symmetric. Let us define the configurations $\mathcal{C}^a$, $\mathcal{C}^b$, and $\mathcal{C}^c$ as those whose supermin is $(0^{k-1}, 1, 0, 1^{n-k-1})$, $(0^k, 1^{n-k})$, or $(0^{\frac{k}{2}}, 1^j, 0^{\frac{k}{2}}, 1^{n-k-j})$, with $j < \frac{n-k}{2}$, respectively. Given a configuration $\mathcal{C}$, a move M permits some robots to change their position by occupying one of their adjacent nodes. If $\mathcal{C}$ is symmetric, and M permits a robot r to move, then, if r is reflected by the axis of symmetry to another robot r' (indistinguishable from r), also r' is permitted to move in a symmetric way. That is, if r and r' execute the same algorithm, they will perform the same (symmetric) move. However, since the system is asynchronous, it may happen that only one of them performs the entire Look-Compute-Move cycle. In this case the generated configuration has a potential so-called *pending* move. An asymmetric configuration $\mathcal{C}$ is *adjacent* to a symmetric configuration $\mathcal{C}'$ with respect to a move M if $\mathcal{C}$ can be obtained from $\mathcal{C}'$ by applying M to only one of the robots permitted to move by M.

It has been proven that the searching and gathering problems cannot be solved in some symmetric configurations. For instance, it is impossible to solve the searching in a configuration

with an axis of symmetry passing through two nodes, while it is impossible to solve the gathering if the axis of symmetry passes through two edges. We say that a configuration is *allowed* for problem P if it is not periodic and it is not symmetric having one of the symmetries which are unsolvable for P. In particular, all rigid configurations are allowed.

3 ALIGN algorithm

In this section, we devise algorithm ALIGN that, starting from any allowed configuration, reaches one of the configurations $\mathcal{C}^a$, $\mathcal{C}^b$, and $\mathcal{C}^c$ previously defined. Algorithm ALIGN will be exploited in next sections to achieve the configurations suitable for the graph searching, or gathering problems.

In detail, if the initial allowed configuration is symmetric and k is even, ALIGN achieves either configuration $\mathcal{C}^b$ or $\mathcal{C}^c$, and the original type of symmetry is preserved, hence the obtained configuration is still allowed. If the configuration is asymmetric and k is even, then any of $\mathcal{C}^a$, $\mathcal{C}^b$, and $\mathcal{C}^c$ can be achieved, if they are allowed. If k is odd, then the configuration achieved is either $\mathcal{C}^a$ or $\mathcal{C}^b$, if this latter is allowed. The general strategy of the algorithm is the following.

- If the configuration is symmetric, then ALIGN preserves the symmetry by performing two symmetric moves in a way that, if only one of such moves is performed, then the obtained configuration is guaranteed to be asymmetric and not adjacent to another symmetric configuration with respect to any other move that can be possibly performed by ALIGN. When k is odd, the symmetry is preserved until it can be safely broken by moving in an arbitrary direction the unique robot lying on the axis of symmetry.
- If the configuration is asymmetric, then always only one robot is permitted to move by ALIGN. First, the algorithm checks whether the asymmetric configuration is adjacent to some allowed symmetric configuration with respect to some move permitted by ALIGN. In this case, ALIGN forces the only possible pending move. We recall that the moves performed from a symmetric configuration are designed in a way that the configuration obtained is not adjacent to any other symmetric configuration different from the correct one. Therefore, from an asymmetric configuration adjacent to an allowed symmetric one with respect to the permitted move of ALIGN, the robot that has to move can be univocally determined and the original symmetry preserved. Note that, such behavior is performed even if the initial configuration is asymmetric. In this case, the configuration obtained after the move is symmetric and allowed, and the algorithm proceeds like in the case that the initial configuration was symmetric. In fact, as the robots are oblivious, they cannot distinguish the two cases.
- If an asymmetric configuration is not adjacent to any symmetric configuration with respect to any move of ALIGN, then the algorithm in [**?**] is performed. Such algorithm, ensures that only one move is performed and the obtained configuration is always rigid, thus it is allowed.

In detail, algorithm ALIGN is based on four moves described below. Let $\mathcal{C}$ be any allowed configuration and let $\mathcal{C}^{\min} = (v_0, v_1, \ldots, v_{n-1})$ be its supermin. Let ℓ_1 be the smallest integer such that $\ell_1 > 0$, $v_{\ell_1} = 0$ and $v_{\ell_1-1} = 1$; let ℓ_2 be the smallest integer such that $\ell_2 > \ell_1$, $v_{\ell_2} = 0$ and $v_{\ell_2-1} = 1$; let ℓ_{-1} be the greatest integer such that $\ell_{-1} < n$ and $v_{\ell_{-1}} = 0$. The four moves permitted by ALIGN are the following:

- **REDUCE$_0(\mathcal{C})$:** The robot at node v_0 moves to node v_1;
- **REDUCE$_1(\mathcal{C})$:** The robot at node v_{ℓ_1} moves to node v_{ℓ_1-1};
- **REDUCE$_2(\mathcal{C})$:** The robot at node v_{ℓ_2} moves to node v_{ℓ_2-1};
- **REDUCE$_{-1}(\mathcal{C})$:** The robot at node $v_{\ell_{-1}}$ moves to node $v_{\ell_{-1}+1}$.

ALIGN, works in two phases: the first phase (Procedure ALIGN-ONE) copes with configura-

tions without any consecutive occupied nodes (i.e. $v_1 = 1$) while the second phase copes with configurations having at least two consecutive occupied nodes (Procedure ALIGN-TWO-SYM, if the configuration is symmetric, and ALIGN-TWO-ASYM otherwise).

Phase ALIGN-ONE. If $v_1 = 1$ and the configuration $\mathcal{C}$ is symmetric, the general strategy is to reduce the supermin by performing REDUCE$_0$. If the two symmetric robots that should move perform their Look-Compute-Move cycles synchronously, then a new symmetric configuration $\mathcal{C}'$ is obtained where the supermin is reduced and the axis of symmetry of $\mathcal{C}$ is preserved. Hence, $\mathcal{C}'$ is allowed.

If only one of the two symmetric robots that should move actually performs the move (due to the asynchronous execution of their respective Look-Compute-Move cycles), then the following lemma, from [**?**], ensures that the configuration $\mathcal{C}'$ obtained is asymmetric and not adjacent to any symmetric configuration with respect to a possible move affecting at most two robots.

Lemma 1 ([?]) *Let $\mathcal{C}$ be an allowed configuration and let $\mathcal{C}'$ be the one obtained from $\mathcal{C}$ after a* REDUCE$_0$ *performed by a single robot. Then, $\mathcal{C}'$ is asymmetric and at least two robots have to move to obtain $\mathcal{C}'$ from an aperiodic symmetric configuration with an axis of symmetry different from that of $\mathcal{C}$.*

It follows that robots can recognize whether $\mathcal{C}'$ has been obtained by performing REDUCE$_0$ from $\mathcal{C}$. This is done by performing such a move on $\mathcal{C}'$ backwards. In the affirmative case, ALIGN forces to perform the possible pending move.

However, it is not always possible to perform REDUCE$_0$ on a symmetric configuration $\mathcal{C}$. Indeed, in case that $\mathcal{C}^{\min} = (0, 1, 0, R)$, for some $R = \overline{R}$, then performing REDUCE$_0$ would imply that two robots occupy the same node (a *collision* occurs). In fact, note that in this case the node symmetric to v_0 is v_2 and performing REDUCE$_0$ consists in moving both robots from v_0 and v_2 to v_1. In this case, we perform REDUCE$_{-1}$. In Lemma 5 (for $j = 1$) we show that such a move performed by only one robot from a configuration $\mathcal{C}$ such that $\mathcal{C}^{\min} = (0, 1, 0, R)$, with $R = \overline{R}$, does not create a symmetric configuration and the configuration obtained is not adjacent to any possible move performed by ALIGN. Therefore, we can again preserve the symmetry by forcing to perform the symmetric move. Note that also in this case, performing REDUCE$_{-1}$ results in reducing the supermin.

If the configuration is asymmetric and it cannot be obtained by performing REDUCE$_0$ or REDUCE$_{-1}$ from any possible allowed symmetric configuration, then we execute the algorithm in [**?**] (ProcedureASYM). The following lemma shows that such algorithm always leads to rigid configurations.

Lemma 2 ([?]) *Let $3 \leq k < n - 2$ robots standing in an n-node ring and forming a rigid exclusive configuration, Algorithm* ASYM *eventually terminates achieving configuration $\mathcal{C}^c$ and all intermediate configurations obtained are exclusive and rigid.*

Algorithm ASYM ensures that each move permits only one robot to change its position, and then no pending moves are possible. If by applying ASYM, we produce an asymmetric configuration which is adjacent to a symmetric configuration with respect to some of the moves permitted by ALIGN, then we force to perform the possible pending move.

Note that, in some symmetric configurations there exists a robot r that occupies a node lying on the axis of symmetry. In these cases, REDUCE$_0$ or REDUCE$_{-1}$ may consists in moving r (in any arbitrary direction). The obtained configuration is asymmetric and not adjacent to any other symmetric configuration with respect to the moves of ALIGN. However, we cannot move the robot symmetric to r as it does not exist, but in this case, we can safely perform ASYM as there are no pending moves.

By the above discussion, it follows that ALIGN-ONE leads to a configuration with two consecutive occupied nodes. In detail, we can obtain: (i) an asymmetric configuration with two consecutive nodes occupied which is not adjacent to any symmetric configuration with respect to

a move permitted by ALIGN-ONE; (ii) an asymmetric configuration with two consecutive occupied nodes which is adjacent to a symmetric configuration with respect to some move permitted by ALIGN-ONE; (iii) a symmetric configuration with two or three consecutive occupied nodes with the axis of symmetry passing in their middle; (iv) a symmetric configuration with two symmetric pairs of consecutive occupied nodes.

Phase ALIGN-TWO-SYM. Once that a configuration with two consecutive occupied nodes is achieved, the second phase of algorithm ALIGN starts. In such configurations, it is not possible to perform REDUCE$_0$ as it would cause a collision (which we want to avoid in this phase). Hence, one move among REDUCE$_1$, REDUCE$_2$ or REDUCE$_{-1}$ is performed.

In symmetric configurations, we perform REDUCE$_1$ every time it is possible. This occurs when the asynchronous execution of the two symmetric robots that should perform the move does not generate a symmetric configuration with a different axis of symmetry or a configuration which is adjacent to a different symmetric configuration with respect to any move permitted by ALIGN.

If it is not possible to perform REDUCE$_1$, we perform REDUCE$_2$. It can be proven that asynchronous executions cannot generate other symmetries or configurations adjacent to symmetric ones potentially reachable.

There are cases when we cannot perform REDUCE$_1$ and REDUCE$_2$. For instance this can happen if $\mathcal{C}^{\min} = (0^i, 1^j, 0^i, R)$, with $R = \overline{R}$. In fact, in this case, $\mathcal{C}^{\min} = \overline{(\mathcal{C}^{\min}_{2i+j})}$ and performing REDUCE$_1$ corresponds to move the robot at v_{i+j} which is symmetric to that at v_{i-1}. Similar instances where it is not possible to perform REDUCE$_2$ can occur. In such cases, we perform REDUCE$_{-1}$ and show that this cannot create any different symmetry or configuration adjacent to symmetric ones with respect to any move permitted by ALIGN.

To give more detail on the behavior of the algorithm in the case of symmetric configurations, we define the following three sets. Let S_1 be the set of symmetric configurations with supermin $(0^i, 1, 0, R)$, where $i \geq 2$ and R contains a sequence 0^i. Let S_2 be the set of configurations $\mathcal{C} \in S_1$ such that $\mathcal{C}^{\min} = (0^i, 1^j, 0^i, Z)$ for some $Z = \overline{Z}$ and $j \geq 1$. Finally, let S_3 be the set of configurations $\mathcal{C} \in S_1$ such that $\mathcal{C}^{\min} = (0^i, 1^{j'}, 0^x, 1^j, 0^x, 1^{j'}, 0^i, Z)$ for some $Z = \overline{Z}$, $j, j' > 0$ and $1 \leq x \leq i$ or configurations $\mathcal{C} \in S_1$ such that $\mathcal{C}^{\min} = (0^i, 1^j, 0^{i-1}, 1, 0, R, 1)$, $R = \overline{R}$, $j > 0$.

The sets S_2 and S_3 contain the configurations where it is not possible to perform REDUCE$_1$ or REDUCE$_2$, respectively. In a series of lemmata provided in Appendix, we show that given a symmetric configuration $\mathcal{C}$, if $\mathcal{C}^1$, $\mathcal{C}^2$, and $\mathcal{C}^{-1}$ are the configuration obtained from $\mathcal{C}$ by applying REDUCE$_1$, REDUCE$_2$, and REDUCE$_{-1}$, respectively, on only one robot, then

- (Lemma 6) $\mathcal{C}^1$ is symmetric only if $\mathcal{C} \in S_1 \setminus S_3$, $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, or $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$, for some $i > 1$, $\ell > 0$;
- (Lemma 7) $\mathcal{C}^1$ is adjacent to a symmetric configuration $\mathcal{C}'$ different from $\mathcal{C}$ with respect to REDUCE$_1$ only if $\mathcal{C} \in S_1 \setminus S_3$, $\mathcal{C}' \in S_1 \setminus S_3$ or $(\mathcal{C}^1)^{\min} = (0^i, 1, 0, 1, 0^w, 1, 1)$, for some $i > 1$, $w < x$;
- (Lemmata 8 and 10) if $\mathcal{C} \in S_1 \setminus S_3$ or $\mathcal{C}^{\min} = (0^i, 1, 0, 1, 0^x, 1, 0, 1)$, where $i > 1$ and $x < i$, then $\mathcal{C}^2$ is asymmetric and not adjacent to any symmetric configuration different from $\mathcal{C}$ with respect to moves REDUCE$_1$ and REDUCE$_2$;
- (Lemma 9) if $\mathcal{C} \in S_2$, or $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, or $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1,$ for some $i > 1$, $\ell > 0$, then $\mathcal{C}^{-1}$ is asymmetric and it is not adjacent to any symmetric configuration different from $\mathcal{C}$ with respect to moves REDUCE$_1$, REDUCE$_2$, and REDUCE$_{-1}$;

Therefore, ALIGN-TWO-SYM works as follows. If $\mathcal{C}$ is in S_2, then REDUCE$_1$ cannot be performed. However, we can safely perform REDUCE$_{-1}$ (Lemma 9). If $\mathcal{C} \notin S_2$, then ALIGN-TWO-SYM first computes the configuration $\mathcal{C}'$ that would be obtained from $\mathcal{C}$ by applying REDUCE$_1$ on only one robot. If $\mathcal{C}'$ is symmetric, then we know that $\mathcal{C} \in S_1 \setminus S_3$, $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, or $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$, for some $\ell > 0$ (Lemma 6). In the former case, we can safely perform REDUCE$_2$ as the ob-

tained configuration is neither symmetric nor adjacent to any other symmetric configuration (Lemma 8). In the latter two cases, we cannot perform REDUCE$_2$ but, similarly, we can safely perform REDUCE$_{-1}$ (Lemma 9).

If $\mathcal{C}'$ is asymmetric, then ALIGN-TWO-SYM checks whether it can be obtained by applying REDUCE$_1$ from a symmetric configuration $\mathcal{C}''$ different from $\mathcal{C}$. To this aim, it computes all the configurations that can possibly generate $\mathcal{C}'$. As REDUCE$_1$ reduces the supermin, then by performing it, the starting node of the supermin in the obtained configuration is either the same of the previous one or it is one of the endpoints of a sequence of consecutive occupied nodes which is generated by the move itself. It follows that $\mathcal{C}''$ can be computed by increasing the supermin of $\mathcal{C}'$ by moving one of the robots in the endpoints of the sequence of consecutive occupied nodes at the beginning of the supermin sequence or the possible robot in position v_{ℓ_1}. In other words, if $\mathcal{C}' = (0^i, 1^j, 0, R, 1)$ for $i \geq 2$ and $j \geq 1$, then $\mathcal{C}''$ can be only among the following configurations: $\mathcal{C}^\alpha := (0^{i-1}, 1, 0, 1^{j-1}, R, 1)$, $\mathcal{C}^\beta := (0^{i-1}, 1^j, R, 0, 1)$, and, if $R = (1, R')$, $\mathcal{C}^\gamma := (0^i, 1^{j+1}, 0, R', 1)$. If at least two among $\mathcal{C}^\alpha$, $\mathcal{C}^\beta$, and $\mathcal{C}^\gamma$ are symmetric and the move from both of them to $\mathcal{C}'$ corresponds to REDUCE$_1$ (i.e. two symmetric configurations are adjacent to $\mathcal{C}'$ with respect to REDUCE$_1$), then at least one of them must belong to $S_1 \setminus S_3$ (Lemma 7). Therefore, we can safely perform REDUCE$_2$ on such configuration and REDUCE$_1$ on the other one (see the first example in Appendix A.3).

In Appendix, we provide special arguments for the symmetric configurations $\mathcal{C}^{s_1} = (0^{i_1}, 1, 0, 1, 0^x, 1, 0, 1)$ and $\mathcal{C}^{s_2} = (0^{i_2}, 1, 1, 0^y, 1, 1)$ with $x < i_1$ and $y < i_2$.

In any other symmetric configuration, ALIGN-TWO-SYM performs REDUCE$_1$.

Phase ALIGN-TWO-ASYM. Procedure ALIGN-TWO-ASYM works similarly to ALIGN-ONE when the configuration is asymmetric. First, it checks whether the given configuration $\mathcal{C}$ has been obtained from a symmetric and allowed configuration $\mathcal{C}'$ by performing only one of the two symmetric moves. In the affirmative case, it performs the possible pending move, otherwise it performs algorithm ASYM. Given the moves performed by procedures ALIGN-ONE and ALIGN-TWO-SYM, a configuration $\mathcal{C}$ with $\mathcal{C}^{\min} = (0^i, 1^j, 0^x, 1^{j'}, R, 1)$, $j \geq 1$, $x \geq 1$, and $j' \geq 0$ can be adjacent to a symmetric configuration $\mathcal{C}'$ with respect to one of such moves only if this latter is one of the following configurations: $\mathcal{C}^\alpha := (0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'}, R, 1)$, $\mathcal{C}^\beta := (0^{i-1}, 1^j, R, 0, 1)$, if $j' > 0$, $\mathcal{C}^\gamma := (0^i, 1^j, 0^{x-1}, 1, 0, 1^{j'-1}, R, 1)$, or, if $R = (0, 1, R')$, $\mathcal{C}^\delta := (0^i, 1^j, 0^x, 1^{j'+1}, 0, R', 1)$. We remark that at most one of the above configurations can be symmetric. Let $\mathcal{C}^i$ be such symmetric configuration, if by applying ALIGN-TWO-SYM (or ALIGN-ONE if $\mathcal{C}^i$ has no consecutive occupied nodes) on a single robot of $\mathcal{C}^i$ we obtain $\mathcal{C}$, then $\mathcal{C}$ has been possibly obtained from $\mathcal{C}^i$ and then ALIGN-TWO-ASYM performs the possible pending move (see the second example in Appendix A.3). If none of $\mathcal{C}^i$, $i = \alpha, \beta, \gamma, \delta$, is symmetric, then $\mathcal{C}$ has not been obtained from any symmetric configurations and then ALIGN-TWO-ASYM performs ASYM. As in the case of ALIGN-ONE, if the robot that moved from $\mathcal{C}^i$ to $\mathcal{C}$ is the one on the axis of symmetry of $\mathcal{C}^i$, then algorithm ASYM is performed.

The next theorem shows the correctness of ALIGN. The proof relies on Lemmata 5–10, proven in Appendix.

Theorem 1 *Let $3 \leq k < n - 2$, $k \neq 4$, robots standing in an n-node ring forming an exclusive allowed configuration, Algorithm* ALIGN *eventually terminates achieving one exclusive allowed configuration among $\mathcal{C}^a$, $\mathcal{C}^b$, or $\mathcal{C}^c$.*

Proof Let us model all the possible executions of ALIGN as a directed graph where each configuration is represented as a node and there exists an arc (u, v) if there exist a move and a time schedule of the algorithm that starting from the configuration represented by u lead to that represented by v, even with possible pending moves. An execution of ALIGN is represented by a path in this graph. In what follows, we show that such paths are acyclic, are made of nodes representing allowed configurations, and they always lead to a node representing one of the configurations $\mathcal{C}^a$, $\mathcal{C}^b$, or $\mathcal{C}^c$.

We can partition the nodes into three sets: those representing symmetric configurations, those representing asymmetric configurations which are adjacent to some symmetric configurations with respect to one of the moves permitted by ALIGN, and those representing the remaining asymmetric configurations. We denote such sets as S, $AS1$ and $AS2$, respectively. Lemmata 1, 2 and 5–10 imply the following properties.

- A node in S representing a configuration $\mathcal{C}$ has either one or two outgoing arcs. If it has exactly two outgoing arcs, then one of them is directed to the node v' representing the configuration $\mathcal{C}'$ obtained if both the symmetric robots permitted to move by ALIGN perform their moves synchronously. The other arc is directed to the node v'' representing the configuration $\mathcal{C}''$ obtained if only one of the two symmetric robots permitted to move by ALIGN performs its move. In other words, the former arc models the case where both the two symmetric robots permitted to move perform the entire cycle Look-Compute-Move, while the latter arc models the case where only one of them performs entirely such cycle. Note that, v' belongs to S, while v'' belongs to $AS1$. Moreover, if $\mathcal{C}$ is allowed, then also $\mathcal{C}'$ is. If the node has exactly one outgoing arc then the robot r moved by ALIGN lies on the axis of symmetry. In this case, any move performed by ALIGN moves r in an arbitrary direction. Then, the arc is directed to a node in $AS1$.
- A node in $AS1$ representing a configuration $\mathcal{C}''$ has exactly one incoming arc from a node in S, it can have some incoming arcs from nodes in $AS2$, and it has exactly one outgoing arc, directed to a node in S or in $AS2$. If the outgoing arc is directed to a node in S, then one of the incoming arcs comes from a node u in S and models the case when only one of the two symmetric robots permitted to move by ALIGN from the configuration $\mathcal{C}$ represented by u performs its move. From Lemmata 5–10, there exists only one of such nodes. The outgoing arc leads to the node in S representing configuration $\mathcal{C}'$ which can be obtained by moving synchronously both the symmetric robots permitted to move by ALIGN from $\mathcal{C}$. Note that both C and C'' are allowed configurations. If the outgoing arc is directed to a node in $AS2$, then $\mathcal{C}''$ has been obtained by a configuration $\mathcal{C}$ corresponding to a node in S such that the robot r moved by ALIGN lies on the axis of symmetry. In this case, ALIGN performs ASYM from $\mathcal{C}''$ and the obtained configuration is in $AS2$.
- A node in $AS2$ has exactly one outgoing arc, directed either to another node in $AS2$ or to a node in $AS1$ but it cannot be directed to a node in S (by Lemma 2). It can have some arcs coming from nodes in $AS1$ or $AS2$.

It follows that any execution path performed by the algorithm is made of nodes representing allowed configurations. Moreover, each allowed configuration has an outgoing arc that is traversed by the execution path of the algorithm. Note also that any move performed by the algorithm reduces the supermin of a configuration. This implies that the graph is acyclic, as we can define a topological ordering of the nodes on the basis of the ordering given by the supermin of the corresponding configurations. The statement is then proven by observing that configurations in $\mathcal{C}^a$, $\mathcal{C}^b$, or $\mathcal{C}^c$ are those with the minimum possible supermin and hence are the only possible sinks of the graph. ■ ■

4 Searching in a Ring

In this section, we present an algorithm that allows a team of robots to perpetually search a ring. In [**?**], it is shown that, for any starting configuration, it is not possible to search an n-node ring using k robots if $n \leq 9$, $k \leq 3$, or $k \geq n-2$. However, there is an algorithm allowing $5 \leq k \leq n-3$ robots to search a ring with $n \geq 10$ nodes (but for $k = 5$ and $n = 10$), for rigid initial configurations.

If k is even and the axis does not pass through an empty node, the searching is clearly unsolvable because a synchronous execution of any algorithm either cause a collision in the node lying on the axis or does not allow to search the edges incident to such node. It follows that in the searching problem, the symmetric allowed configurations are all those with k odd and those with k even where the axis does not pass through an empty node, provided that $3 < k < n-2$ and $n > 9$. Moreover, allowed configurations must be exclusive.

Here, we improve over the algorithm in [**?**] by addressing also aperiodic symmetric configurations. We will use two sub-procedures (defined in the Appendix): Algorithm Compact-Align is used after Align to achieve configuration $\mathcal{C}^b$, in the case Align reaches configuration $\mathcal{C}^c$, and Algorithm Break-Symmetry is used to achieve $\mathcal{C}^a$ in the case that k is odd.

Algorithm Search-Ring. The algorithm first checks whether $k = n-3$ or if n is odd and k is even. In the affirmative case, any allowed configuration must be asymmetric, and therefore the algorithm of [**?**] can be applied and the ring is searched. If k is odd, we first use Algorithm Break-Symmetry to break the potential symmetry and then use the algorithm of [**?**]. Each of these configurations used during the searching phase of the algorithm of [**?**] are asymmetric and are not adjacent to any symmetric configuration reached by Algorithm Break-Symmetry. Therefore, there is no ambiguity (no pending move) when a robot recognizes such a configuration.

If n and k are even, we may be in allowed symmetric configurations and therefore the Search-Ring proceeds in two phases. Algorithm Compact-Align is first applied until one of the configurations in $\mathcal{A}$ (described in Appendix) is achieved. This is guaranteed by the fact that both $\mathcal{C}^a$ and $\mathcal{C}^b$ belong to $\mathcal{A}$. Then, the algorithm proceeds to Phase 2 which actually performs the searching.

The intuitive explication of the Searching algorithm (Phase 2) is as follows. All robots are aligned on consecutive nodes. Then, both robots r and r' at the ends of this segment move (one clockwise and the other anti-clockwise) to reach the two adjacent nodes opposite to the occupied segment. Then, the two robots q and q' occupying the ends of the "long" occupied segment move to their empty neighbors. These moves indicate to r and r' that it is time to go back toward the "long" segment, and that is what happens. Finally, when r is adjacent to q' and r' is adjacent to q', then q and q' move to their empty neighbors in order to re-build the original segment. Then, the process is repeated perpetually.

Such a sequence of moves actually searches the ring. Moreover, by definition of the configurations met during the process (configurations in $\mathcal{A}$), there is no ambiguity in the choice of the robot(s) that must move. Finally, there are no conflicts between the different phases of our procedure because any configuration in $\mathcal{A}$ is not adjacent to any symmetric configuration not in $\mathcal{A}$. This allows to state the following theorem:

Theorem 2 *Let $4 < k \leq n-3$ robots, standing in an n-node ring forming an allowed configuration, Algorithm* Search-Ring *perpetually searches the ring, but for $(n,k) \in \{(10,5);(10,6)\}$ when the configuration is symmetric.*

5 Gathering in a ring

The gathering asks for a deterministic strategy that let robots meet at a common node and remain in there. This means that more than one robot must be allowed to occupy a single node, i.e. a collision occurs. Actually, collisions are referred as *multiplicities* (or *towers*) in the gathering literature. We assume that the robots have the *local* multiplicity detection capability. That is, a robot perceives whether a node x is occupied by more than one robot only if it resides on x as well. In [**?**], it has been proven that the multiplicity detection capability is necessary to allow robots to accomplish the gathering task. Moreover, the problem requires that an initial

configuration is not periodic, does not admit an axis of symmetry passing through two edges, and contains more than two robots [**?**]. Starting from ALIGN, in Appendix is described a strategy to reach the following result:

Theorem 3 *Let $3 \leq k < n-4$, $k \neq 4$, robots, empowered with the local multiplicity detection, standing in an n-node ring forming an allowed configuration, Algorithm* GATHERING *eventually terminates with only one node occupied.*

The idea is to first make use of Procedure ALIGN to reach one of the following configuration types: $\mathcal{C}^a = (0^{k-1}, 1, 0, 1^{n-k-1})$, $\mathcal{C}^b = (0^k, 1^{n-k})$, with k or n odd, $\mathcal{C}^c = (0^{\frac{k}{2}}, 1^j, 0^{\frac{k}{2}}, 1^{n-k-j})$, with j or n odd.

Then, configurations in $\mathcal{C}^a$ are moved either to $\mathcal{C}^b$ (if k is odd) or to $\mathcal{C}^c$ (if k is even and n is odd). Configurations in $\mathcal{C}^b$ move from $(0^k, 1^{n-k})$ to $(0^{k-2}, 1^{n-k+2})$ (if k is odd) and configurations in $\mathcal{C}^c$ move to $(0^{k-3}, 1^{n-k+3})$ with a multiplicity in the middle. By continuing with this process, the gathering is accomplished, eventually.

6 Conclusion

We have proposed two algorithms to solve the perpetual searching and the gathering tasks under minimal assumptions. The results provided here constitute a complete characterization of the two problems but for few marginal cases. For the perpetual searching, our algorithm handles the missing cases of our previous work on the same subject. For the gathering, this paper closes a long standing open problem. In fact, the results in the literature left some open cases which are now closed. Moreover, we have also addressed the lack of a unified algorithm that works for any gatherable configuration.

References

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):58–82, 2006.

[2] L. Blin, J. Burman, and N. Nisse. Distributed exclusive and perpetual tree searching. In *Proc of 26th DISC*, volume 7611 of *LNCS*, pages 403–404, 2012.

[3] J. Chalopin and S. Das. Rendezvous of mobile agents without agreement on local orientation. In *Proc of 37th ICALP*, volume 6199, pages 515–526, 2010.

[4] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Proc. of 30th ICALP*, volume 2719, pages 1181–1196, 2003.

[5] J. Czyzowicz, L. Gasieniec, and A. Pelc. Gathering few fat mobile robots in the plane. *Theor. Comp. Sci.*, 410(6–7):481 – 499, 2009.

[6] G. D'Angelo, G. Di Stefano, and A. Navarra. Gathering of six robots on anonymous symmetric rings. In *18th SIROCCO*, volume 6796 of *LNCS*, pages 174–185, 2011.

[7] G. D'Angelo, G. Di Stefano, and A. Navarra. How to gather asynchronous oblivious robots on anonymous rings. In *Proc. of 26th DISC*, volume 7611 of *LNCS*, pages 330–344, 2012.

[8] G. D'Angelo, G. Di Stefano, A. Navarra, N. Nisse, and K. Suchan. A unified approach for different tasks on rings in robot-based computing systems. In *Proc. of 15th IEEE IPDPS APDCM*, 2013. to appear.

[9] Y. Dieudonne, A. Pelc, and D. Peleg. Gathering despite mischief. In *Proc. of 23rd SODA*, pages 527–540, 2012.

[10] P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by oblivious mobile robots.* Morgan and Claypool, 2012.

[11] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.

[12] D. Ilcinkas, N. Nisse, and D. Soguet. The cost of monotonicity in distributed graph searching. *Distributed Computing*, 22(2):117–127, 2009.

[13] T. Izumi, T. Izumi, S. Kamei, and F. Ooshita. Mobile robots gathering algorithm with local weak multiplicity in rings. In *Proc. of 17th SIROCCO*, volume 6058 of *LNCS*, pages 101–113, 2010.

[14] S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil. Asynchronous mobile robot gathering from symmetric configurations. In *Proc. of 18th SIROCCO*, volume 6796 of *LNCS*, pages 150–161, 2011.

[15] S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil. Gathering an even number of robots in an odd ring without global multiplicity detection. In *Proc. of 37th MFCS*, volume 7464 of *LNCS*, pages 542–553, 2012.

[16] R. Klasing, A. Kosowski, and A. Navarra. Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring. *Theor. Comput. Sci.*, 411:3235–3246, 2010.

[17] R. Klasing, E. Markou, and A. Pelc. Gathering asynchronous oblivious mobile robots in a ring. *Theor. Comput. Sci.*, 390:27–39, 2008.

Appendix

A Align Algorithm

A.1 Pseudo-code

Algorithm: Align
Input: Allowed configuration $\mathcal{C}$ with $\mathcal{C}^{\min} = (v_0, v_1, \ldots, v_{n-1})$

```
1  if v_1 = 1 then
2  |  Align-One(C);
3  else
4  |  if C is symmetric then
5  |  |  Align-Two-Sym(C);
6  |  else
7  |  |  Align-Two-Asym(C);
```

Figure 1: Algorithm Align.

Algorithm: Align-One
Input: Allowed configuration $\mathcal{C}$ with $\mathcal{C}^{\min} = (v_0, v_1, \ldots, v_{n-1})$

```
1  if C is symmetric then
2  |  if C^min = (0,1,0,R), with R = R̄ then
3  |  |  reduce_{-1}(C);
4  |  else
5  |  |  reduce_0(C);
6  else
7  |  Let C^min = (0,1,X,1);
8  |  Let C' = (1,1,X,0);
9  |  if C' is symmetric and allowed and C'^min = (0,1,1,X) then
10 |  |  if The robot that moved from C' to C is not the one on the axis of symmetry of C' then
11 |  |  |  Perform the reduce_0 move symmetrical to the one from C' to C and exit;
12 |  else
13 |  |  if C' = (1,1,0,1,0,1,1,0,X',0), C'^min = (0,1,0,1,1,0,X',0,1,1), and X' = X̄' then
14 |  |  |  if The robot that moved from C' to C is not the one on the axis of symmetry of C' then
15 |  |  |  |  Perform the reduce_0 move symmetrical to the one from C' to C and exit;
16 |  |  else
17 |  |  |  if C^min = (0,1,0,1^j,0,1,X'') then
18 |  |  |  |  Let C'' = (0,1,0,1^{j+1},0,X'');
19 |  |  |  |  if C'' is symmetric and allowed and C''^min = (0,1,0,1^{j+1},0,X'') then
20 |  |  |  |  |  if The robot that moved from C' to C is not the one on the axis of symmetry of
   |  |  |  |  |  C' then
21 |  |  |  |  |  |  Perform the reduce_0 move symmetrical to the one from C' to C and exit;
22 |  Asym(C);
```

Figure 2: First phase of algorithm Align.

Algorithm: ALIGN-TWO-SYM
Input: Allowed configuration $\mathcal{C}$ with $\mathcal{C}^{\min} = (v_0, v_1, \ldots, v_{n-1})$

if $\mathcal{C} \in S_2$ *or* $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$*, or* $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$ **then**
  REDUCE$_{-1}(\mathcal{C})$;
else
  Let $\mathcal{C}'$ be the configuration obtained by applying REDUCE$_1(\mathcal{C})$ on $\mathcal{C}$;
  if $\mathcal{C}'$ *is symmetric* **then**
    REDUCE$_2(\mathcal{C})$;
  else
    Let $\mathcal{C}' = (0^i, 1^j, 0, R, 1)$;
    $\mathcal{C}^\alpha := (0^{i-1}, 1, 0, 1^{j-1}, R, 1)$;
    $\mathcal{C}^\beta := (0^{i-1}, 1^j, R, 0, 1)$;
    if $R = (1, R')$ **then** $\mathcal{C}^\gamma := (0^i, 1^{j+1}, 0, R', 1)$;
    if *At least two among* $\mathcal{C}^\alpha$*,* $\mathcal{C}^\beta$*, and* $\mathcal{C}^\gamma$ *are symmetric, the move from both of them corresponds to* REDUCE$_1$*, and (*$\mathcal{C} \in S_1 \setminus S_3$ *or* $\mathcal{C} = (0^i, 1, 0, 1, 0^x, 1, 0, 1)$*)* **then**
      REDUCE$_2(\mathcal{C})$;
    else
      REDUCE$_1(\mathcal{C})$;

Figure 3: Second phase of algorithm ALIGN for symmetric configurations.

Algorithm: ALIGN-TWO-ASYM
Input: Allowed configuration $\mathcal{C}$ with $\mathcal{C}^{\min} = (v_0, v_1, \ldots, v_{n-1})$

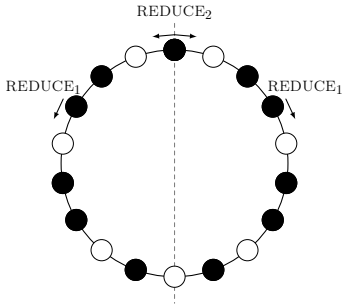
Let $\mathcal{C} = (0^i, 1^j, 0^x, 1^{j'}, R, 1)$ with $j \geq 1$, $x \geq 1$, and $j' \geq 0$;
$\mathcal{C}^\alpha := (0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'}, R, 1)$;
$\mathcal{C}^\beta := (0^{i-1}, 1^j, R, 0, 1)$;
if $j' > 0$ **then** $\mathcal{C}^\gamma := (0^i, 1^j, 0^{x-1}, 1, 0, 1^{j'-1}, R, 1)$;
if $R = (0, 1, R')$ **then** $\mathcal{C}^\delta := (0^i, 1^j, 0^x, 1^{j'+1}, 0, R', 1)$;
for $i \in \{\alpha, \beta, \gamma, \delta\}$ **do**
  if $\mathcal{C}^i$ *is symmetric and allowed* **then**
    if $\mathcal{C}^i$ *has no consecutive occupied nodes* **then**
      Let $\mathcal{C}'$ be the configuration obtained by executing ALIGN-ONE on $\mathcal{C}^i$;
    else
      Let $\mathcal{C}'$ be the configuration obtained by executing ALIGN-TWO-SYM on $\mathcal{C}^i$;
    if $\mathcal{C} = \mathcal{C}'$ **then**
      if *The robot that moved from* $\mathcal{C}^i$ *to* $\mathcal{C}$ *is not the one on the axis of symmetry of* $\mathcal{C}^i$ **then**
        Perform the REDUCE$_0$ move symmetrical to the one from $\mathcal{C}'$ to $\mathcal{C}$ and exit;
ASYM($\mathcal{C}$);

Figure 4: Second phase of algorithm ALIGN for symmetric configurations.

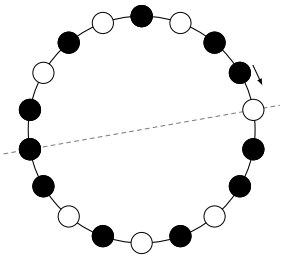
A.2 Further details

Here, we give some special arguments for the symmetric configurations $\mathcal{C}^{s_1} = (0^{i_1}, 1, 0, 1, 0^x, 1, 0, 1)$ and $\mathcal{C}^{s_2} = (0^{i_2}, 1, 1, 0^y, 1, 1)$ with $x < i_1$ and $y < i_2$.

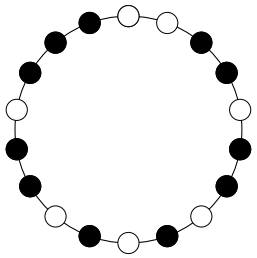
Note that in these cases REDUCE$_1$ and REDUCE$_{-1}$ coincide. Moreover, in $\mathcal{C}^{s_2}$ the only move that reduces the supermin and does not create a collision is REDUCE$_1$. However, performing REDUCE$_1$ from $\mathcal{C}^{s_1}$ and $\mathcal{C}^{s_2}$ leads to the same configuration $\mathcal{C}' = (0^i, 1, 0, 1, 0^w, 1, 1)$ if $i = i_2 = i_1 + 1$ and $w = x = y - 1$. For these reasons ALIGN-TWO-SYM performs REDUCE$_1$ from $\mathcal{C}^{s_2}$ and REDUCE$_2$ from $\mathcal{C}^{s_1}$. This latter move cannot create a symmetric configuration nor a configuration adjacent to any symmetric configuration different from $\mathcal{C}^{s_1}$ with respect to any move permitted by ALIGN (Lemma 10).



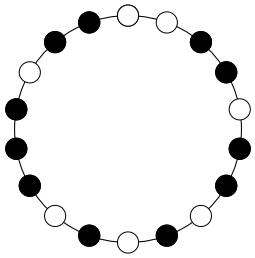
(a) Configuration $\mathcal{C}$.



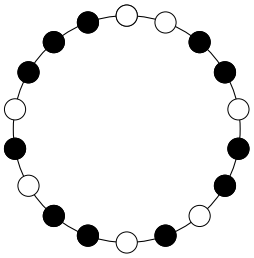
(b) Configuration $\mathcal{C}'$ computed from $\mathcal{C}$ by applying REDUCE$_1$.



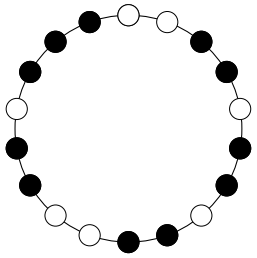
(c) Configuration $\mathcal{C}''$ computed from $\mathcal{C}$ by applying REDUCE$_2$.



(d) Configuration $\mathcal{C}^\alpha$ computed from $\mathcal{C}''$ in Procedure ALIGN-TWO-ASYM.



(e) Configuration $\mathcal{C}^\gamma$ computed from $\mathcal{C}''$ in Procedure ALIGN-TWO-ASYM.



(f) Configuration $\mathcal{C}^\delta$ computed from $\mathcal{C}''$ in Procedure ALIGN-TWO-ASYM.

Figure 5: First example

A.3 Examples of execution

The example in Figure 5 shows a case where applying REDUCE$_1$ on a symmetric configuration results in a symmetric configuration with a different axis of symmetry. Let us consider the configuration $\mathcal{C}$ in Figure 5a such that $\mathcal{C}^{\min} = (0,0,1,0,0,1,0,1,0,0,1,0,0,1,0,1,0,1)$. As $v_1 = 0$ and $\mathcal{C}$ is symmetric, then Procedure ALIGN-TWO-SYM is performed. It first compute configuration $\mathcal{C}'$ in Figure 5b which is the one that would be obtained from $\mathcal{C}$ by applying REDUCE$_1$ on only one robot. As such a configuration is symmetric, REDUCE$_2$ is applied. If only one robot moves, then the configuration $\mathcal{C}''$ in Figure 5c is obtained. Such configuration is asymmetric and it has a possible pending move.

From configuration $\mathcal{C}''$, Procedure ALIGN-TWO-ASYM is applied. Such a procedure computes the unique symmetric configuration which $\mathcal{C}''$ is adjacent to. To this aim, it computes the four possible configurations that can generate $\mathcal{C}''$ by applying ALIGN. Such configurations are:

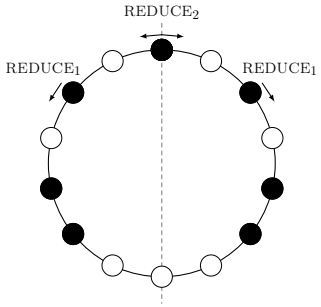
- $\mathcal{C}^\alpha$ given in Figure 5d;
- $\mathcal{C}^\beta$ which is equivalent to $\mathcal{C}$;
- $\mathcal{C}^\gamma$ given in Figure 5e;
- $\mathcal{C}^\delta$ given in Figure 5f.

Among such configurations, only one is symmetric which is $\mathcal{C}^\beta = \mathcal{C}$. Therefore, ALIGN-TWO-ASYM is able to identify the robot that has to move in order to perform the possible pending move. In this specific case, The robot that moved from $\mathcal{C}$ to $\mathcal{C}''$, is the one on the axis of symmetry. It follows that there are no pending moves and ALIGN-TWO-ASYM proceeds by applying ASYM.

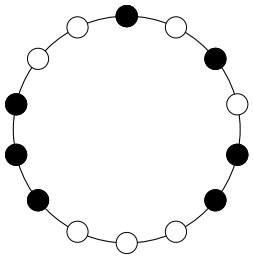
The example in Figure 6 shows a case where applying REDUCE$_1$ on a symmetric configuration results in an asymmetric configuration which is adjacent to another symmetric configuration, different from the original one, with respect to REDUCE$_1$. Let us consider the configuration of Figure 6a. As $v_1 = 0$ and $\mathcal{C}$ is symmetric, then Procedure ALIGN-TWO-SYM is performed. It first compute configuration $\mathcal{C}'$ in Figure 6b which is the one that would be obtained from $\mathcal{C}$ by applying REDUCE$_1$ on only one robot. As such configuration is asymmetric, the procedure checks whether it can be obtained by applying REDUCE$_1$ from a symmetric configuration different from $\mathcal{C}$. To this aim it computes:

- $\mathcal{C}^\alpha$ which is equivalent to $\mathcal{C}$;
- $\mathcal{C}^\beta$ given in Figure 6d;
- $\mathcal{C}^\gamma$ given in Figure 6e.

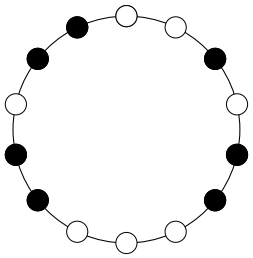
Both configurations $\mathcal{C}^\alpha$ and $\mathcal{C}^\gamma$ are symmetric, and configuration $\mathcal{C}'$ can be obtained from both of them by



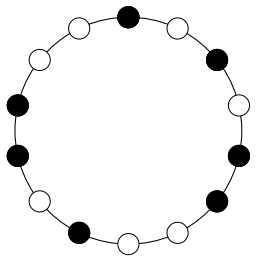
(a) Configuration $\mathcal{C}$.



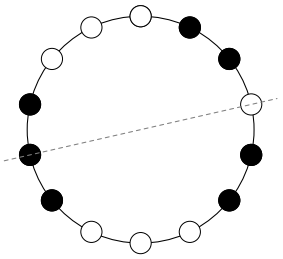
(b) Configuration $\mathcal{C}'$ computed from $\mathcal{C}$ by applying REDUCE$_1$.



(c) Configuration $\mathcal{C}''$ computed from $\mathcal{C}$ by applying REDUCE$_2$.



(d) Configuration $\mathcal{C}^\beta$ computed from $\mathcal{C}'$ in Procedure ALIGN-TWO-SYM.



(e) Configuration $\mathcal{C}^\gamma$ computed from $\mathcal{C}'$ in Procedure ALIGN-TWO-SYM.

Figure 6: Second example

applying REDUCE$_1$. By Lemma 7, follows that one of them must belong to $S_1 \setminus S_3$. In fact, $\mathcal{C}^\alpha \in S_1 \setminus S_3$. Therefore, Procedure ALIGN-TWO-SYM exploits Lemma 8 and applies REDUCE$_2$ on $\mathcal{C}$. The obtained configuration $\mathcal{C}''$ is given in Figure 6c. It is easy to see that such configuration is asymmetric and it is not adjacent to any symmetric configuration different from $\mathcal{C}$ with respect to any move permitted by ALIGN.

A.4 Correctness

In the following, we distinguish between the following two types of symmetry.

Definition 1 *Let $\mathcal{C}$ be a configuration such that $\mathcal{C}^{\min} = (0^i, 1, R, 1)$, then $\mathcal{C}$ is said symmetric if* **exactly** *one of the following two conditions is satisfied:*

Symmetry 1. *$R = \overline{R}$, or*

Symmetry 2. *$(1, R, 1) = (Y, 0^i, Z)$ with $Y = \overline{Y}$ and $Z = \overline{Z}$*

In other words, if $\mathcal{C}$ has Symmetry 1, then the sequence of consecutive occupied nodes which starts with the supermin, is cut in the middle by the axis of symmetry. In this case, there are two symmetrical supermins at the two sides of such sequence. If $\mathcal{C}$ has Symmetry 2 then the axis of symmetry passes through the middle of sequences Y and Z and there are two symmetrical supermin corresponding to the two sequences of i consecutive occupied nodes.

Lemma 3 *If $(0, \overline{R}) = (R, 0)$ and $(0, R) = (\overline{R}, 0)$, then $R \in (0^k)$, $k \in \mathbb{N}$.*

Proof. R may be $\emptyset$. Clearly, $R = (0)$ if $|R| = 1$. Assume $|R| > 1$.

By induction on $0 \leq j \leq \lfloor \frac{n}{2} \rfloor$, we show that $R = (0^j, X, 0^j)$.Assume that $R = (0^j, a, X, b, 0^j)$, then, by symmetries, $(0, 0^j, a, X, b, 0^j) = (0^j, b, \overline{X}, a, 0^j, 0)$ and $(0, 0^j, b, \overline{X}, a, 0^j) = (0^j, a, X, b, 0^j, 0)$, hence $a = b = 0$ and thus, $R = (0^{j+1}, X, 0^{j+1})$. Then, $X = \emptyset$, or $|X| = 1$, in which case the only possibility is $X = (0)$, or $X = (0^{j+1}, a', X', b', 0^{j+1})$ and the result holds by induction. ■ ■

Lemma 4 *Let Y be any sequence and let X satisfy $(Y, X) = (X, \overline{Y})$ and $X = \overline{X}$. Then, the set of solutions for consists of the sequences X that satisfy: $X = (Y^i, U)$, with $Y = (U, V)$ and $U = \overline{U}$ and $V = \overline{V}$, and $i \geq 0$.*

Proof. Clearly, any sequence X satisfying the properties of the lemma is a valid solution. We prove that any solution has the desired form by induction on the length of the solution.

Assume first that there is a solution X with $|X| \leq |Y|$. Then, X is a prefix of Y because $(Y, X) = (X, \overline{Y})$. Therefore, $Y = (X, V)$. Plugging it into the equation, we get $(X, V, X) = (X, \overline{V}, X)$. Therefore, $V = \overline{V}$. Hence, X and Y have the desired form.

Now, consider a solution X such that $|X| > |Y|$. Then, Y is a prefix of X because $(Y, X) = (X, \overline{Y})$. Therefore, $X = (Y, X')$. Plugging it into the equation, we get $(Y, Y, X') = (Y, X', \overline{Y})$. Moreover, because $X = \overline{X}$, we get that $(Y, \overline{X'}, Y) = (Y, X', \overline{Y})$. All together, we get that $X = (Y, X')$ with $(Y, X') = (X', \overline{Y})$ and $X' = \overline{X'}$.

Therefore, by induction, we get that, there is $i \geq 0$ such that $X' = (Y^i, U)$ with $Y = (U, V)$ and $U = \overline{U}$ and $V = \overline{V}$. Hence, $X = (Y^{i+1}, U)$ and the lemma holds. ■ ■

Lemma 5 *Let $\mathcal{C}$ be a symmetric and allowed configuration with supermin $\mathcal{C}^{\min} = (0, 1^j, 0, R)$, for $R = \overline{R}$ and $j \geq 1$, and let $\mathcal{C}'$ the configuration obtained by applying* REDUCE$_{-1}$ *on only one robot on $\mathcal{C}$. Then $\mathcal{C}'$ is asymmetric and it is not adjacent with respect to* REDUCE$_0$ *and* REDUCE$_{-1}$ *to any symmetric configuration different from $\mathcal{C}$.*

Proof. Since R is palindrome, then $\mathcal{C}^{\min} = (0, 1^j, 0, R)$ is the only supermin as otherwise $\mathcal{C}$ is periodic. Therefore, $R = (1^{j'}, 0, R', 0, 1^{j'})$, for some $j' > j$ and $R' = \overline{R'}$. The configuration obtained by applying REDUCE$_{-1}$ on only one robot on $\mathcal{C}$ is $\mathcal{C}' = (0, 1^j, 0, 1^{j'}, 0, R', 1, 0, 1^{j'-1})$. We distinguish the following two cases.

- $j' - 1 > j$. In this case, $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^{j'-1}, 0, 1, R', 0, 1^{j'})$. The configuration is asymmetric as if there exist another sequence starting with $(0, 1^j, 0)$ then we obtain a contradiction with the superminimality of $\mathcal{C}^{\min}$.
  Configuration $\mathcal{C}'$ can be obtained by applying REDUCE$_0$ or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}'$ only if (i) $\mathcal{C}'' = (0, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R', 0, 1^{j'-1})$, or (ii) $\mathcal{C}'' = \mathcal{C}$, or (iii) $\mathcal{C}'' = (0, 1^{j+1}, 0, 1^{j'-2}, 0, 1, R', 0, 1^{j'})$. In fact, in case (i) $\mathcal{C}'$ is obtained by performing REDUCE$_0$ on $\mathcal{C}''$ and $\mathcal{C}''^{\min} = (0, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R', 0, 1^{j'-1})$, or by performing REDUCE$_{-1}$ on $\mathcal{C}''$ and $\mathcal{C}''^{\min} = (0, 1^{j'-1}, 0, 1^{j+1}, 0, 1^{j'-1}, 0, R', 1)$; in case (ii) $\mathcal{C}'$ is obtained by performing REDUCE$_0$ on $\mathcal{C}''$ and $\mathcal{C}''^{\min} = \mathcal{C}^{\min}$; and in case (iii), $\mathcal{C}'$ is obtained by performing REDUCE$_{-1}$ on $\mathcal{C}''$ and $\mathcal{C}''^{\min} = (0, 1^{j'}, 0, 1^{j+1}, 0, 1^{j'-2}, 0, 1, R')$.
  In the first case of (i), as the supermin is $(0, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R', 0, 1^{j'-1})$ we have that $(0, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R', 0, 1^{j'}$ $(0, 1^{j+1}, 0, 1^{j'-1}, 0, R', 1, 0, 1^{j'-1})$ which implies that $(1, R') < (R', 1)$. This last inequality can be satisfied only if $R = (1^{j''})$ for some $j'' > j$, which implies that $k = 4$, a contradiction. In the second case of (i), we must have that the axis passes through the middle of the first sequence of $j' - 1$ consecutive empty nodes and that $(1^{j+1}, 0, 1^{j'-1}, 0, R', 1)$ is palindrome, that is $(1^j, 0, 1^{j'-1}, 0, R') = (R', 0, 1^{j'-1}, 0, 1^j)$. By Lemma 4, it follows that $R' = ((1^j, 0, 1^{j'-1}, 0)^\ell, 1^j)$ for some $\ell \geq 0$. However, in this case we have that $\mathcal{C}^{\min} = (0, 1^j, 0, 1^{j'}, 0, (1^j, 0, 1^{j'-1}, 0)^\ell, 1^j, 0, 1^{j'})$ which is a contradiction as for $\ell > 0$, $\mathcal{C}^{\min}_{j+j'+2} < \mathcal{C}^{\min}$ and for $\ell = 0$ $\mathcal{C}$ is periodic.
  The case (ii) corresponds to the move that has been actually performed.
  In case (iii), we have that hat the axis passes through the middle of the first sequence of j' consecutive empty nodes and that the sequence $(1^{j+1}, 0, 1^{j'-2}, 0, 1, R')$ is palindrome. By Lemma 4, this can occur only if $j = 0$ as otherwise the sequence $(1^{j+1}, 0, 1^{j'-2}, 0, 1,)$ cannot be split into two palindrome sub-sequences. We obtained a contradiction as $j \geq 1$.
- $j'-1 = j$. In this case $\mathcal{C}' = (0, 1^j, 0, 1^{j+1}, 0, R', 1, 0, 1^j)$ and either $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^j, 0, 1^{j+1}, 0, R', 1)$ or $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^j, 0, 1, R', 0, 1^{j+1})$. In any case R' cannot contain a sequence $(0, 1^j, 0, 1^j)$. Therefore, if $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^j, 0, 1^{j+1}, 0, R', 1)$, the axis of symmetry can only pass through the first sequence of j consecutive empty nodes or in the robot separating the two sequences of j consecutive empty nodes. In the first case, R' must start with a $(1^{j-1}, 0)$ but this is a contradiction to the superminimality of $\mathcal{C}$. In the second case, we must have that $(1^{j+1}, 0, R', 1) = (1, R', 0, 1^{j+1})$ that is $(1^j, 0, R') = (R', 0, 1^j)$. By Lemma 4, this implies that $R' = ((1^j, 0)^\ell, 1^j)$ for some $\ell \geq 0$. It follows that $\mathcal{C}^{\min} = (0, 1^j, 0, 1^{j+1}, 0, (1^j, 0)^\ell, 1^j, 0, 1^{j+1})$. However, this implies that: if $\ell = 0$, then $\mathcal{C}$ is periodic, and if $\ell > 0$, then $\mathcal{C}^{\min}_{2j+2} < \mathcal{C}^{\min}$. In any case, we obtain a contradiction.
  If $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^j, 0, 1, R', 0, 1^{j+1})$, the axis of symmetry can only pass through the robot separating the two sequences of j consecutive empty nodes. Note that, in this case, $\mathcal{C}'^{\min} = (0, 1^j, 0, 1^j, 0, 1, R', 0, 1^{j+1}) = (0, 1^j, 0, 1^j, 0, 1^{j+1}, 0, R', 1)$ and hence the same arguments as before can be applied.

Configuration $\mathcal{C}'$ can be obtained by applying REDUCE$_0$ or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}'$ only if (i) $\mathcal{C}'' = (0, 1^{j+1}, 0, 1^j, 0, 1, R', 0, 1^j)$. In fact, $\mathcal{C}'$ can be obtained by applying REDUCE$_0$ or REDUCE$_{-1}$ to $\mathcal{C}''$ if $\mathcal{C}''^{\min} = (0, 1^{j+1}, 0, 1^j, 0, 1, R', 0, 1^j)$ or $\mathcal{C}''^{\min} = (0, 1^j, 0, 1^{j+1}, 0, 1^j, 0, R', 1)$, respectively. The first case is impossible as $\mathcal{C}''^{\min}$ is not minimum. In the second case, we must have that $(1^{j+1}, 0, 1^j, 0, R', 1) = (1, R', 0, 1^j, 0, 1^{j+1})$ and hence $(1^j, 0, 1^j, 0, R') = (R', 0, 1^j, 0, 1^j)$. By Lemma 4, this implies that $R' = ((1^j, 0, 1^j, 0,)^\ell, 1^j)$ or $R' = ((1^j, 0, 1^j, 0,)^\ell, 1^j, 0, 1^j)$. In any case, we obtain that either $\mathcal{C}$ is periodic or that $\mathcal{C}'^{\min}$ is not minimum. ■

■

Lemma 6 *Let $\mathcal{C}$ be a symmetric and allowed configuration with supermin $\mathcal{C}^{\min} = (0^i, 1, R)$, $i > 1$, and let $\mathcal{C}'$ the configuration obtained by applying* REDUCE$_1$ *on only one robot on $\mathcal{C}$. If $\mathcal{C}'$ is symmetric, then $\mathcal{C} \in S_1 \backslash S_3$, $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, or $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$.*

Proof. We first show that $\mathcal{C} \in S_1$ and then that the only configuration in S_3 such that C' is symmetric is such that $\mathcal{C}^{\min} = (0^{i-1}, 1, 0, 0, 1, 0^{i-1}, (1, 0, 1, 0^{i-1})^\ell, 1, 0, 1)$.

Let $\mathcal{C}^{\min} = (0^i, 1^j, 0, R')$ for $j \geq 1$, then $S = (0^i, 1^{j-1}, 0, 1, R')$ is a representation of $\mathcal{C}'$. We show that if $j > 1$, then $\mathcal{C}'^{\min} = (0^i, 1^{j-1}, 0, 1, R')$ is the unique supermin of $\mathcal{C}'$, i.e. the configuration is asymmetric. Note that $S < \mathcal{C}^{\min}$ and that $\overline{S_h} > \overline{\mathcal{C}_h^{\min}} \geq \mathcal{C}^{\min}$ for each $h \neq \ell_1 - 1$. Moreover, $\overline{S_{j-1}} > S$. Therefore, $\overline{S_h}$ cannot be a supermin of C' for each h. By contradiction, let j' be an integer such that $S_{j'} < S$. Then $\mathcal{C}_{j'}^{\min} < \mathcal{C}^{\min}$, a contradiction.

It follows that if C' is symmetric, then $\mathcal{C}^{\min} = (0^i, 1, 0, R')$ and the supermin of C' is $(0^{i+1}, 1, R')$, $(0^{i+1}, \overline{R'}, 1)$ or both. In this case, C' can only have symmetry 1 as there is only one sequence of $i+1$ consecutive occupied nodes. This implies that $(0^{i+1}, 1, R') = (0^{i+1}, \overline{R'}, 1)$. Let us assume that $\mathcal{C}$ has symmetry 1, then the sequence $(1, 0, R')$ is palindrome and then $R' = (R'', 0, 1)$ with $R'' = \overline{R''}$. Since, by the symmetry of $\mathcal{C}'$, $(1, R') = (\overline{R'}, 1)$, then $(1, R'', 0, 1) = (1, 0, R'', 1)$ and therefore $(R'', 0) = (0, R'')$. By Lemma 3, $R'' = (0^{j'})$ for some j' and then $W_{min}^{\mathcal{C}} = (0^i, 1, 0^{j'+2}, 1)$ which is a contradiction as it implies that $k = n - 2$. Therefore, C has symmetry 2 and then $C \in S_1$.

Since C' is asymmetric if $\mathcal{C}^{\min} = (0^i, 1^j, 0, R')$ for any R', it remains to show that it is asymmetric if $\mathcal{C}^{\min} \neq (0^i, 1, 0^{i-1}, 1, 0, R, 1)$, for some $R = \overline{R}$. Let us assume that $\mathcal{C}^{\min} = (0^i, 1, 0^{i-1}, 1, 0, R, 1)$ with $R = \overline{R}$, then $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^{i-2}, 1, 0, R, 1)$. Therefore, C' can only have symmetry 1 and hence the sequence $(1, 0^{i-2}, 1, 0, R, 1)$ must be palindrome. By Lemma 4, C' is symmetric if and only if $R = ((0^{i-2}, 1, 0)^\ell, 0^{i-3})$, with $i \geq 3$, that is $\mathcal{C}^{\min} = (0^i, 1, 0^{i-1}, 1, 0, (0^{i-2}, 1, 0)^\ell, 0^{i-3}, 1)$ and then $\mathcal{C}$ is asymmetric.

To show the second part of the proof, let us assume that $\mathcal{C}^{\min} = (0^i, 1^{j'}, 0^x, 1^j, 0^x, 1^{j'}, 0^i, Z)$ with $j > 0$ and $Z = \overline{Z}$. We can assume that $j' = 1$ as otherwise $\mathcal{C}'$ is asymmetric. After applying REDUCE$_1$, we have $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^{x-1}, 1^j, 0^x, 1, 0^i, Z)$ with $(1, 0^{x-1}, 1^j, 0^x, 1, 0^i, Z)$ palindrome, that is $(1, 0^{x-1}, 1^j, 0^x, 1, 0^i, Z) = (Z, 0^i, 1, 0^x, 1^j, 0^{x-1}, 1)$. By Lemma 4 it follows that $Z = ((1^{j+1}, 0, 1, 0^i)^\ell, 1^{j+1})$ for some $\ell \geq 0$, $x = 1$, and $i = 1$; or $Z = ((1, 0, 1, 0^i)^\ell, 1, 0, 1)$ for some $\ell \geq 0$, $x = 1$, and $j = 0$; or $Z = ((1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$ for some $\ell \geq 0$, $x = i$, and $j = 1$. In the first case we obtain a contradiction with the hypothesis that $i > 1$; in the second case, $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$; in the third case $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$.

■ ■

Lemma 7 *Let $\mathcal{C}$ be a symmetric and allowed configuration and let $\mathcal{C}'$ be the configuration obtained by applying* REDUCE$_1$ *on only one robot on $\mathcal{C}$. If $\mathcal{C}'$ is adjacent with respect to* REDUCE$_0$ *to a symmetric configuration $\mathcal{C}''$ different from $\mathcal{C}'$, then $\mathcal{C} \in S_1 \setminus S_3$, $\mathcal{C}'' \in S_1 \setminus S_3$ or $\mathcal{C}'^{\min} = (0^i, 1, 0, 1, 0^x, 0, 1)$.*

Proof. Let $\mathcal{C}'^{\min} = (0^i, 1, R, 1)$ with $i > 1$. Then, $\mathcal{C}$ and $\mathcal{C}''$ are one of the following configurations:

- If R does not start by 0^{i-2} or $i = 2$, $\mathbf{C1}^{\min} = (0^{i-1}, 1, 0, R, 1)$;
- If $\overline{R}$ does not start by 0^{i-2} or $i = 2$, $\mathbf{C2}^{\min} = (0^{i-1}, 1, 0, \overline{R}, 1)$;
- If $R = (1^{j-1}, 0, 1, R')$ and R' does not start by 0^{i-1}, $\mathbf{C3}^{\min} = (0^i, 1^{j+1}, 0, R', 1)$.

The proof proceed by showing that if two among **C1**, **C2**, and **C3** are symmetric than one of them belongs to $S_1 \setminus S_3$ or $\mathcal{C}'^{\min} = (0^i, 1, 0, 1, 0^x, 0, 1)$. The following facts analyze any possible pair of symmetries. In what follows, we assume that **C1**, **C2**, and **C3** are distinct.

Fact 1 **C1** *and* **C2** *cannot have both symmetry* 1.

Proof. By symmetry of **C1**, $(0,R) = (\overline{R},0)$ and, by symmetry of **C2**, $(0,\overline{R}) = (R,0)$. By Lemma 3, $R = \{\emptyset; (0^{n-i-2})\}$ that is $k = n-2$, a contradiction. ■ ■

Fact 2 **C1** *and* **C3** *cannot have both symmetry* 1.

Proof. By symmetry of **C1**, $(0,R) = (\overline{R},0)$ so the last element of R (and of R') is 0, so by symmetry of **C3**, $j = 0$, a contradiction because $j > 0$. ■ ■

Fact 3 *If* **C2** *and* **C3** *have symmetry* 1, *then* $j = 1$ *and* $R = (0,1,0^{n-i-5},1)$. *Therefore,* **C2** $= (0^{i-1},1,0,1,0^{n-i-5},1,0,1)$, **C3** $= (0^i,1,1,0^{n-i-4},1,1)$ *and* $\mathcal{C}'^{\min} = (0^i,1,0,1,0^x,0,1)$.
and $\mathcal{C}'^{\min} = (0^i,1,0,1,0^x,0,1)$.

Proof. Recall that $R = (1^{j-1},0,1,R')$.

By symmetry of **C2**, $(0,\overline{R}) = (R,0)$ and the first element of R is 0 and thus, $j = 1$ and $R = (0,1,R')$. By symmetry of **C3**, $(1,0,R') = (\overline{R}',0,1)$. Therefore, $R' = (1)$ or $R' = (0,1)$, or $R' = (R'',0,1)$.

In the latter case, by symmetry of **C2**, $(0,1,0,\overline{R}'',1,0) = (0,1,R'',0,1,0)$. Hence, $(0,\overline{R}'') = (R'',0)$. By symmetry of **C3**, $(1,0,R'',0,1) = (1,0,\overline{R}'',0,1)$, hence $R'' = \overline{R}''$. Hence, R'' is a sequence of 0 by Lemma 4. ■ ■

Fact 4 *If* **C1** *has (only) symmetry* 1 *and* **C2** *has (only) symmetry* 2, *then* **C2** $\in S_2$.

Proof. Assume that $R \neq (0^{i-2})$ as otherwise, **C1** would be periodic. By symmetry of **C1** $(0,R) = (\overline{R},0)$. This implies that $R = (X,0)$ with $\overline{X} = X$. By symmetry of **C2**, $(1,0,0,X,1) = (Y,0^{i-1},Z)$ with $Y = \overline{Y}$ and $Z = \overline{Z}$.

- Case $X = (0^{i-3},V)$, $V \neq \emptyset$ and starts by 1 and $Y = (1)$. By symmetry of X, $(\overline{V},0^{i-3}) = (0^{i-3},V)$ and, by symmetry of $Z = (V,1)$, $(V,1) = (1,\overline{V})$. Hence, $V = (1,V',0^{i-3})$. Again, the symmetry of X gives that $(\overline{V}',1) = (1,V')$ and, by symmetry of Z, $(V',0^{i-3}) = (0^{i-3},\overline{V}')$. Therefore, if $V' \neq \emptyset$, $V' = (0^{i-3},V'',1)$. The symmetry gives that $(\overline{V}'',1) = (1,V'')$ and $(V'',0^{i-3}) = (0^{i-3},\overline{V}'')$. Hence, $V'' = (1,V''',0^{i-3})$. Going on this way, we get that

$$R = (0^{i-3},(1,0^{i-3})^{\lfloor\frac{n-3i+2}{i-2}\rfloor},1,(0^{i-3},1)^{\lfloor\frac{n-3i+2}{i-2}\rfloor},0^{i-2}).$$

  By plugging R into the definition of **C2**, it follows that **C2** $\in S_2$.
- Case $X = (U,0^{i-2})$, and $Z = (1)$, $U \neq \emptyset$ and finishes by 1. This case is excluded. Indeed, by definition of **C2**, $\overline{R}$ cannot start with 0^{i-2} unless $i = 2$. If $i = 2$, by minimality of **C1** and **C2**, R must start and finish with 1, contradicting $(0,R) = (\overline{R},0)$.
- Case $X = (U,0^{i-1},V)$, $U \neq \emptyset$ and finishes by 1, $V \neq \emptyset$ and starts by 1. In that case, $Y = (1,0,0,U) = (\overline{U},0,0,1)$ and $Z = (V,1) = (1,\overline{V})$, that is $V = (1,V')$ and $U \in \{(1),(0,1),(U',0,0,1)\}$, where $U' = \overline{U}'$. Moreover, by symmetry of X, $(\overline{V},0^{i-1},\overline{U}) = (U,0^{i-1},V)$ that is $(V',1,0^{i-1},\overline{U}) = (U,0^{i-1},1,V')$ that implies that V' starts like U or is empty.
  - If $U = (1)$, then **C1** $= (0^{i-1},1,0,1,0^{i-1},1,V',0,1)$. If V' is not empty, then it starts with 1 and $(V',0) > (0,V')$. Therefore, **C1** is not supermin as $\overline{\mathbf{C1}}_{2i+1}$ is strictly smaller. If V' is empty, then **C1** $= (0^{i-1},1,0,1,0^{i-1},1,0,1)$ which is periodic and has more than one symmetry, a contradiction.
  - If $U = (0,1)$, then **C1** $= (0^{i-1},1,0,0,1,0^{i-1},1,V',0,1)$ and V' either starts with 0 or with $(0,1)$. In the first case, **C1** is periodic with more than one axis of symmetry, in the second case, **C1** is not supermin as $\overline{\mathbf{C1}}_{2i+2}$ is strictly smaller since $(V',0) > (0,V')$, a contradiction to the superminimality of **C1**.

– If $U = (U', 0, 0, 1)$, then $\mathbf{C1} = (0^{i-1}, 1, 0, U', 0, 0, 1, 0^{i-1}, 1, V', 0, 1)$. If $U' \neq (0^x)$ for some $x \geq 0$, then $(0^{i-1}, 1, 0, U') > (, 0^{i-1}, 1, 0, 0, U')$ and hence, $\overline{\mathbf{C1}}_{2i+2+|U'|}$ is strictly smaller than **C1**. If $U' = (0^x)$ for some $x \geq 0$, then $\mathbf{C1} = (0^{i-1}, 1, 0^{x+3}, 1, 0^{i-1}, 1, V', 0, 1)$ and therefore, if $V' \neq (0^y)$ for some $0 \leq y \leq x+2$, then $(V', 0) > (0, V')$ and hence $\overline{\mathbf{C1}}_{2i+x+3}$ is strictly smaller than **C1**. Finally, if $U' = (0^x)$ for some $x \geq 0$ and $V' = (0^x)$ for some $0 \leq y \leq x+2$, then $\mathbf{C1} = (0^{i-1}, 1, 0^{x+3}, 1, 0^{i-1}, 1, 0^{y+1}, 1)$. Moreover, by the symmetry of X, $\mathbf{C1} = (0^{i-1}, 1, 0^{y+1}, 1, 0^{i-1}, 1, 0^{x+3}, 1)$. It follows that $y = x+2$ and **C1** is periodic with more than one axis of symmetry, a contradiction.

■

Fact 5 *If* **C1** *has symmetry* 2 *and* **C2** *has symmetry* 1*, then* $\mathbf{C1} \in S_2$.

Proof. By symmetry with previous fact. ■

Fact 6 *It is not possible that* **C1** *has symmetry* 1 *and* **C3** *has symmetry* 2.

Proof. $\mathbf{C1} = (0^{i-1}, 1, 0, R, 1)$ and $\mathbf{C3} = (0^i, 1^{j+1}, 0, R', 1)$ and $R = (1^{j-1}, 0, 1, R')$, $i > 1$ and $j > 0$.

By symmetry of **C1**, $(0, R) = (\overline{R}, 0)$ so $(0, 1^{j-1}, 0, 1, R') = (\overline{R}', 1, 0, 1^{j-1}, 0)$ and the last element of R' is 0.

By symmetry of **C3**, $(1, 1^j, 0, R', 1) = (Y, 0^i, Z)$ with $Y = \overline{Y}$ and $Z = \overline{Z}$. Note that R' cannot contain a sequence of i consecutive 0 by superminimality of **C1**. Therefore, the only possibility is $R' = (0^{i-1}, R'')$ with $Z = (R'', 1)$ and $Y = 1^{j+1}$.

Then, $(R'', 1) = (1, \overline{R}'')$ and thus, $R'' = (1, R''')$ with R''' palindrome. Moreover, because $(0, R) = (\overline{R}, 0)$, we get $(0, 1^{j-1}, 0, 1, 0^{i-1}, 1, R''') = (R''', 1, 0^{i-1}, 1, 0, 1^{j-1}, 0)$. We now use Lemma 4 by setting $X = R'''$ and $Y = (0, 1^{j-1}, 0, 1, 0^{i-1}, 1)$. The possible cases are:

- $U = (0)$ and $V = (1, 0, 1, 0, 1)$, then $R''' = ((0, 1, 0, 1, 0, 1)^\ell, 0)$ for $\ell \geq 0$, only if $i = j = 2$. In which case $R' = (0, 1, (0, 1, 0, 1, 0, 1, 0, 1)^{\frac{n-10}{6}}, 0)$ and $\mathbf{C1} = (0, 1, 0, 1, 0, 1, 0, 1, (0, 1, 0, 1, 0, 1, 0, 1)^{\frac{n-10}{6}}, 0, 1)$ which is periodic, a contradiction.
- $U = (0), 1^{j-1}, 0$ and $V = (1, 0^{i-1}, 1)$, then $R''' = ((0, 1^{j-1}, 0, 1, 0^{i-1}, 1)^\ell, 0, 1^{j-1}, 0)$ for $\ell \geq 0$. In which case $R' = (0^{i-1}, 1, (0, 1^{j-1}, 0, 1, 0^{i-1}, 1)^{\frac{n-2(i+j+2)}{i+j+2}}, 0, 1^{j-1}, 0)$ and

$$\mathbf{C1} = (0^{i-1}, 1, 0, 1^{j-1}, 0, 1, 0^{i-1}, 1, (0, 1^{j-1}, 0, 1, 0^{i-1}, 1)^{\frac{n-2(i+j+2)}{i+j+2}}, 0, 1^{j-1}, 0, 1)$$

  which is periodic, a contradiction.
- $U = (0, 1, 0, 1, 0)$ and $V = (1)$, then $R''' = ((0, 1, 0, 1, 0, 1)^\ell, 0, 1, 0, 1, 0)$ for $\ell \geq 0$, only if $i = j = 2$. In which case $R' = (0, 1, (0, 1, 0, 1, 0, 1)^{\frac{n-14}{6}}, 0, 1, 0, 1, 0)$ and $\mathbf{C1} = (0, 1, 0, 1, 0, 1, 0, 1, (0, 1, 0, 1, 0, 1)^{\frac{n-14}{6}}, 0, 1, 0, 1, 0, 1)$ which is periodic, a contradiction.

■

Fact 7 *If* **C1** *has symmetry* 2 *and* **C3** *has symmetry* 1*, then* $\mathbf{C1} \in S_1 \setminus S_3$.

Proof. $\mathbf{C1} = (0^{i-1}, 1, 0, R, 1)$ and $\mathbf{C3} = (0^i, 1^{j+1}, 0, R', 1)$ and $R = (1^{j-1}, 0, 1, R')$, $i > 1$ and $j > 0$. Moreover, by superminimality of **C1**, R' cannot contain a sequence of i consecutive 0.

By Symmetry of **C3**, $R' = (R'', 1^j)$ and $(0, R'') = (\overline{R}'', 0)$.

By Symmetry of **C1**, $(1, 0, R, 1) = (Y, 0^i, Z)$ with $Y = \overline{Y}$ and $Z = \overline{Z}$. That is, $(1, 0, 1^{j-1}, 0, 1, R'', 1^{j+1}) = (Y, 0^{i-1}, Z)$.

- Case $R'' = (0^{i-1}, U)$ and $Y = (1, 0, 1^{j-1}, 0, 1)$ and $Z = (U, 1^{j+1})$. Because $(0, R'') = (\overline{R}'', 0)$, U finishes with 0 and, thus, because Z palindrome, $U = (1^{j+1}, U')$ with U' palindrome. Hence, because $(0, R'') = (\overline{R}'', 0)$, we have $(0^i, 1^{j+1}, U') = (U', 1^{j+1}, 0^i)$. By Lemma 4, there is $\ell \geq 0$ such that $U' = ((0^i, 1^{j+1})^\ell, 0^i)$. However, it means that R' contains i consecutive 0, a contradiction.
- Case $R'' = (U, 0^{i-1})$ and $Z = 1^{j+1}$ and $Y = (1, 0, 1^{j-1}, 0, 1, U)$.
  - Case $U = \emptyset$. Then, $R' = (0^{i-1}, 1^j)$ and $\mathbf{C3} = (0^i, 1^{j+1}, 0^i, 1^{j+1})$ which is periodic with more than one axis of symmetry.

- Then, assume $U \neq \emptyset$. Therefore, U must contain at least one 1 since otherwise, $R' = (U, 0^{i-1}, 1^j)$ would have at least i consecutive 0. But, if U contains at least one 1, because $(0, R'') = (\overline{R}'', 0)$, we get $(0, U, 0^{i-1}) = (0^{i-1}, \overline{U}, 0)$ and $U = (0^{i-2}, U')$ with U' palindrome. Because Y is a palindrome, $(1, 0, 1^{j-1}, 0, 1, 0^{i-2}, U') = (U', 0^{i-2}, 1, 0, 1^{j-1}, 0, 1)$. By Lemma 4, there is $\ell \geq 0$ such that
  - $U' = ((1, 0, 1, 0, 1, 0)^\ell, 1)$ only if $i = 3$ and $j = 2$, or
  - $U' = ((1, 0, 1, 0, 1, 0)^\ell, 1, 0, 1)$ only if $i = 3$ and $j = 2$, or
  - $U' = ((1, 0, 1^{j-1}, 0, 1, 0^{i-2})^\ell, 1, 0, 1^{j-1}, 0, 1)$.

  Therefore, since $R' = (0^{i-2}, U', 0^{i-1}, 1^j)$,
  - $R' = (0, (1, 0, 1, 0, 1, 0)^{\frac{n-14}{6}}, 1, 0^2, 1^2)$ only if $i = 3$ and $j = 2$, or
  - $R' = (0, (1, 0, 1, 0, 1, 0)^{\frac{n-16}{6}}, 1, 0, 1, 0^2, 1^2)$ only if $i = 3$ and $j = 2$, or
  - $R' = (0^{i-2}, (1, 0, 1^{j-1}, 0, 1, 0^{i-2})^{\frac{n-3(i+j+1)}{i+j+1}}, 1, 0, 1^{j-1}, 0, 1, 0^{i-1}, 1^j)$.

  In any case, by plugging R' into **C1**, we have that $\mathbf{C1} \in S_1 \setminus S_3$.

- Case $R'' = (V, 0^{i-1}, U)$ with $U \neq \emptyset$ starts with 1, $V \neq \emptyset$ finishes with 1, $Y = (1, 0, 1^{j-1}, 0, 1, V)$ and $Z = (U, 1^{j+1})$.
  Because $(0, R'') = (\overline{R}'', 0)$, we get $(0, V, 0^{i-1}, U) = (\overline{U}, 0^{i-1}, \overline{V}, 0)$. Hence, U finishes by 0. Moreover, because Z is a palindrome, $U = (1^{j+1}, U')$ with $U' = \overline{U}'$.
  - Case $V = (1^{j-2}, 0, 1)$. Plugging V and U in $(0, R'') = (\overline{R}'', 0)$, we obtain $(0, 1^{j-2}, 0, 1, 0^{i-1}, 1^{j+1}, U') = (U', 1^{j+1}, 0^{i-1}, 1, 0, 1^{j-2}, 0)$ and U' palindrome. Because, $i > 1$ and $j > 0$, by Lemma 4, there is a solution for U' only for $i = 2$ and $j = 3$. In this case, there is $\ell \geq 0$ with $U' = ((0, 1, 0, 1, 0, 1^4)^\ell, 0, 1, 0, 1, 0)$. Hence, $R' = (1, 0, 1, 0, 1^4, (0, 1, 0, 1, 0, 1^4)^{\frac{n-24}{9}}, 0, 1, 0, 1, 0, 1^3)$ and

    $$\mathbf{C1} = (0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1^4, (0, 1, 0, 1, 0, 1^4)^{\frac{n-24}{9}}, 0, 1, 0, 1, 0, 1^4).$$

    In this case, $\mathbf{C1}_{15}$ is strictly smaller than **C1**, a contradiction to the superminimality of **C1**.
  - Case $V = (V', 1, 0, 1^{j-1}, 0, 1)$ and $V' = \overline{V}'$. Because $(0, R'') = (\overline{R}'', 0)$, we have $(0, V', 1, 0, 1^{j-1}, 0, 1, 0^{i-1}, 1^{j+1}, U$ $(U', 1^{j+1}, 0^{i-1}, V', 1, 0, 1^{j-1}, 0, 1, 0)$.
    - Case $U' = (0)$. Then, $(V', 1, 0, 1^{j-1}, 0, 1, 0^{i-1}, 1^{j+1}) = (1^{j+1}, 0^{i-1}, V', 1, 0, 1^{j-1}, 0, 1)$ and $j = 0$, a contradiction.
    - Case $U' = (0, U'', 0)$ and U'' palindrome. Then

      $$(V', 1, 0, 1^{j-1}, 0, 1, 0^{i-1}, 1^{j+1}, 0, U'') = (U'', 0, 1^{j+1}, 0^{i-1}, V', 1, 0, 1^{j-1}, 0, 1).$$

      It follows that U'' starts and finishes with 1 and then also V' starts and finishes with 1, while V and R'' start with 1.
      Then, we define R''' such that $R'' = (1, R''')$. As $(0, R'') = (\overline{R}'', 0)$, then $(0, 1, R''') = (\overline{R}''', 1, 0)$. The following two cases may arise.
      - $R''' = (0)$. In this case $R'' = (1, 0)$, $R' = (1, 0, 1^j)$, and $\mathbf{C1} = (0^{i-1}, 1, 0, 1^{j-1}, 0, 1, 1, 0, 1^{j+1})$ which is not symmetric, a contradiction.
      - $R''' = (R'''', 1, 0)$ with R'''' possibly empty. In this case, as $(0, 1, R''') = (\overline{R}''', 1, 0)$, we have that $(0, 1, R'''', 1, 0) = (0, 1, \overline{R}'''', 1, 0)$ and then R'''' is palindrome. Plugging R into **C1** we obtain $\mathbf{C1} = (0^{i-1}, 1, 0, 1^{j-1}, 0, 1, 1, 0, 1^{j+1})$. As **C1** has symmetry 2, then there exist an index ℓ such that $\overline{\mathbf{C1}}_\ell = \mathbf{C1}$. Moreover $\ell > i + j + 3$, in other words, the sequence $\overline{\mathbf{C1}}_\ell$ starts in one of the elements of R''''. As R'''' is palindrome, then there exists an $\ell' \neq \ell$ such that $\mathbf{C1}_{\ell'} = \mathbf{C1}$ and $\mathbf{C1}_{\ell'}$ starts in one of the elements of R''''. In detail, if $\overline{\mathbf{C1}}_\ell$ starts at the k-th element of R'''', then $\mathbf{C1}_{\ell'}$ starts at the $(|R''''| - k)$-th element of R''''. However, the elements at the two sides of R'''' in **C1** are different, in particular at one side of R'''' there is $(1, 1)$ while at the other side it is $(1, 0)$. This is a contradiction as it implies that the element a position $k + 2$ of $\overline{\mathbf{C1}}_\ell$ is 1, while the element at position $k + 2$ of $\mathbf{C1}_{\ell'}$ is 0 and $\overline{\mathbf{C1}}_\ell$ and $\mathbf{C1}_{\ell'}$ are equal until element $k + 1$ and hence $\mathbf{C1}_{\ell'}$ is strictly smaller than $\overline{\mathbf{C1}}_\ell$, a contradiction to the superminimality of **C1**.. ■

■

Fact 8 *If* **C1** *has symmetry* 2 *and* **C2** *has symmetry* 2*, then either* $\mathbf{C1} \in S_1 \setminus S_3$ *or* $\mathbf{C2} \in S_1 \setminus S_3$.

Proof. $\mathbf{C1} = (0^{i-1}, 1, 0, R, 1)$ and $\mathbf{C2} = (0^{i-1}, 1, 0, \overline{R}, 1)$. By symmetry, $(1, 0, R, 1) = (Y, 0^{i-1}, Z)$ and $(1, 0, \overline{R}, 1) = (Y', 0^{i-1}, Z')$ with Y, Z, Y' and Z' palindromes. In any case both **C1** and **C2** are in S_1, we then need to show that they are not in S_3.

- Case $Y = (1)$. In this case $\mathbf{C1} \in S_2$ and, since S_2 is contained in S_1 and disjoint from S_3, then $\mathbf{C1} \in S_1 \setminus S_3$.
- Case $Z = (1)$. Not possible since otherwise, **C2** would have i consecutive 0.
- Case $R = (U, 1, 0^{i-1}, 1, V)$, $Y = (1, 0, U, 1)$ and $Z = (1, V, 1)$. In this case, as $Y = \overline{Y}$ and $Z = \overline{Z}$, then $U = (U', 0)$ with $U' = \overline{U}$ and $V = \overline{V}$. Therefore $\mathbf{C2} = (0^{i-1}, 1, 0, V, 1, 0^{i-1}, 1, 0, U', 1) = (0^{i-1}, Y', 0^{i-1}, Z')$. The following cases may arise.
  - Case $Y' = (1, 0, \overline{V}, 1)$ and $Z' = (1, \overline{U}, 1)$. In this case, V is a palindrome because Z is a palindrome, and U is a palindrome because of Z'. Moreover, by symmetry of Y and Y', $(0, U) = (U, 0)$ and $(0, V) = (V, 0)$. By Lemma 4, there are $\ell, \ell' \geq 0$ such that $U = (0)^\ell$ and $V = (0)^{\ell'}$.
    Hence, $R = (0^\ell, 1, 0^{i-1}, 1, 0^{\ell'})$ with $\ell + \ell' = n - 2i - 3$ and $\ell, \ell' < i - 2$. Therefore, $\mathbf{C1} = (0^{i-1}, 1, 0^{\ell+1}, 1, 0^{i-1}, 1, 0^{\ell'}, 1)$ with $\ell + 1 > \ell'$ and $\mathbf{C1} = (0^{i-1}, 1, 0^{\ell'+1}, 1, 0^{i-1}, 1, 0^{\ell}, 1)$ with $\ell' + 1 > \ell$. Combining $\ell + 1 > \ell'$ and $\ell' + 1 > \ell$, we obtain that $\ell = \ell'$ and then $\mathbf{C1} = \mathbf{C2}$, a contradiction.
  - Case $Y' = (1)$. In this case $\mathbf{C2} \in S_2$ and, since S_2 is contained in S_1 and disjoint from S_3, then $\mathbf{C2} \in S_1 \setminus S_3$.
  - Case $Y' = (1, 0, A)$ and $Z' = (B, 1, 0^{i-1}, 1, 0, U', 1)$ with $V = (A, 0^{i-1}, B)$. By contradiction, we assume that both **C1** and **C2** belong to S_3. Therefore, $U' = (1^j)$ and $A = (1^{j'}, 0, 1)$ for some $j, j' > 0$. As V is palindrome, then $(1^{j'}, 0, 1, 0^{i-1}, B) = (\overline{B}, 0^{i-1}, 1, 0, 1^{j'})$. As Z' is palindrome, then $(B, 1, 0^{i-1}, 1, 0, 1^{j+1}) = (1^{j+1}, 0, 1, 0^{i-1}, 1, B)$. From the first equality, we obtain that
    - $B = (1^{j'})$ and $i = 2$ or
    - $B = (1, 0, 1^{j'})$ or
    - $B = (B', 0^{i-1}, 1, 0, 1^{j'})$ where $B' = \overline{B'}$.

    From the second equality, we obtain:
    - $B = (1^j)$ and $i = 2$ or
    - $B = (1^{j+1}, 0)$ or
    - $B = (1^{j+1}, 0, 1, 0^{i-1}, 1, B'')$ where $B'' = \overline{B''}$.

    Combining the two sets of equalities, we can have only the following cases.
    - $B = (1^{j'})$, $j = j'$, and $i = 2$. In this case, $Z = Z' = (1^{j+1}, 0, 1, 0, 1^{j+1})$ and therefore, $\mathbf{C1} = \mathbf{C2}$, a contradiction.
    - $B = (B', 0^{i-1}, 1, 0, 1^{j'}) = (1^{j+1}, 0, 1, 0^{i-1}, 1, B'')$, $B' = (1^{j+1})$, $B'' = (1^{j-1})$ and $i = 2$. In this case, $Z = (1^{j'+1}, 0, 1, 0, 1^{j+1}, 0, 1, 0, 1^{j'+1})$ and $Z' = (1^{j+1}, 0, 1, 0, 1^{j'+1}, 0, 1, 0, 1^{j.+1})$, then, $\mathbf{C1} = (0, 1, 0, 1^j, 0, 1, 0, 1^{j'+1}, 0, 1, 0, 1^{j+1}, 0, 1, 0, 1^{j'+1})$ and $\mathbf{C2} = (0, 1, 0, 1^{j'}, 0, 1, 0, 1^{j+1}, 0, 1, 0, 1^{j'+1}$, **C1** and **C2** are supermin if and only if $j' + 1 > j$ and $j + 1 > j'$, respectively. It follows that $j = j'$ and then $\mathbf{C1} = \mathbf{C2}$, a contradiction.
    - $B = (B', 0^{i-1}, 1, 0, 1^{j'}) = (1^{j+1}, 0, 1, 0^{i-1}, 1, B'')$, $B' = (1, 0, 1)$, $B'' = (0)$ and $j = j' = 0$. In this case $\mathbf{C1} = (0^{i-1}, 1, 0, 1, 0^{i-1}, 1, 0, 10^{i-1}, 1, 0, 10^{i-1}, 1, 0, 1)$ is periodic, a contradiction.
    - $B = (B', 0^{i-1}, 1, 0, 1^{j'}) = (1^{j+1}, 0, 1, 0^{i-1}, 1, B'')$, $B' = (1^{j+1}, 0, 1, 0^{i-1}, 1, B''')$, and $B'' = (B''', 0^{i-1}, 1, 0, 1^{j'})$ with B' and B'' palindrome. In this case, $B = (1^{j+1}, 0, 1, 0^{i-1}, 1, B''', 0^{i-1}, 1, 0, 1^{j'})$ and

      $$\mathbf{C1} = (0^{i-1}, 1, 0, 1^j, 0, 1, 0^{i-1}, 1^{j'+1}, 0, 1, 0^{i-1}, 1^{j+1}, 0, 1, 0^{i-1}, 1, B''', 0^{i-1}, 1, 0, 1^{j'+1})$$

      $$\mathbf{C2} = (0^{i-1}, 1, 0, 1^{j'}, 0, 1, 0^{i-1}, 1^{j+1}, 0, 1, 0^{i-1}, 1, B''', 0^{i-1}, 1, 0, 1^{j'+1}, 0^{i-1}, 1, 0, 1^{j+1})$$

      C1 and **C2** are supermin if and only if $j' + 1 > j$ and $j + 1 > j'$, respectively. It follows that $j = j'$ and hence $B' = (1^{j+1}, 0, 1, 0^{i-1}, 1, B''')$, and $B'' = (B''', 0^{i-1}, 1, 0, 1^j)$.
      We obtain a contradiction by showing that B' and B'' cannot be palindrome at the same time. To this aim we prove the following statement:
      Let X and Y be two sequences such that $(X, 1, Y) = (\overline{Y}, 1, \overline{X})$ and $(Y, \overline{X}) = (X, \overline{Y})$, then either $X = (1^m)$ or $X = (Y, 1^m)$ with Y palindrome and $m > 0$. To obtain a contradiction, we set $X = (1^{j+1}, 0, 1, 0^{i-1})$ and $Y = B'''$.
      We first assume that $|X| < |Y|$. Then, as $(X, 1, Y) = (\overline{Y}, 1, \overline{X})$, we define Y' such that $Y = (Y', \overline{X})$. By plugging Y' into the same equality, we obtain $(X, 1, Y', \overline{X}) =$

$(X, \overline{Y'}, 1, \overline{X})$ which implies that $(1, Y') = (\overline{Y'}, 1)$ and that we can define Y'' such that $Y' = (Y'', 1)$ and $Y'' = \overline{Y''}$. Then $Y = (Y'', 1, \overline{X})$ and plugging it into $(Y, \overline{X}) = (X, \overline{Y})$ we obtain $(Y'', 1, \overline{X}, \overline{X}) = (X, X, 1, Y'')$. By Lemma 4, it follows that $Y'' = ((X, X, 1)^{\ell'}, X')$ where $X = (X', X'')$, $X' = \overline{X'}$, $X'' = \overline{X''}$. If $\ell = 0$, then $Y = (X', 1, \overline{X})$ and from $(Y, \overline{X}) = (X, \overline{Y})$ follows that $(X', 1, \overline{X}, \overline{X}) = (X, X, 1, X')$ and then $(X', 1, X'', X', X'', X') = (X', X'', X', X'', 1, X')$ which implies that $X = (1^m)$ for some $m > 0$. In fact, if we assume that X'' contains a 0, that is $X'' = (1^{m'}, 0, X''', 0, 1^{m'})$ for $m' \geq 0$, then we have $(X', 1, 1^{m'}, 0, X''', 0, 1^{m'}, X', 1^{m'}, 0, X''', 0, 1^{m'}, X') = (X', 1^{m'}, 0, X''', 0, 1^{m'}, X', 1^{m'}$, which is not palindrome. Similar arguments can be used to show that X' does not contains 0. If $\ell > 0$ similar arguments can be used to show that Y'' is palindrome if and only if $X = (1^m)$ for some $m > 0$.

If $|X| > |Y|$, then as $(X, 1, Y) = (\overline{Y}, 1, \overline{X})$, we define X' such that $X = (\overline{Y}, X')$. By plugging Y' into $(Y, \overline{X}) = (X, \overline{Y})$, we obtain $(Y, \overline{X'}, Y) = (\overline{Y}, X', \overline{Y})$ which implies that $Y = \overline{Y}$ and $X = \overline{X}$. From $(X, 1, Y) = (\overline{Y}, 1, \overline{X})$ it follows that $(Y, X', 1, Y) = (Y, 1, X', Y)$ and then $(X', 1) = (1, X')$. Therefore, $X' = (1^m)$ for some > 0 and $X = (Y, 1^m)$.

– Case $Y' = (1, 0, V, 1, 0^{i-1}, 1, 0, A)$ and $Z' = (B, 1)$ with $U' = (A, 0^{i-1}, B)$. In this case **C2** does not belong to S_3. ■

■

Fact 9 **C1** *and* **C3** *cannot have both symmetry* 2.

Proof. **C1** $= (0^{i-1}, 1, 0, R, 1)$ and **C3** $= (0^i, 1^{j+1}, 0, R', 1)$ and $R = (1^{j-1}, 0, 1, R')$, $i > 1$ and $j > 0$.

By Symmetry of **C3**, $(1^{j+1}, 0, R', 1) = (Y, 0^i, Z)$ with Y, Z palindromes. Moreover, by superminimality of **C1**, R' cannot contain a sequence of i consecutive 0.

Hence, $R' = (0^{i-1}, U)$ which contradicts the definition of **C3**. ■ ■

Fact 10 **C2** *cannot have symmetry* 1 *when* **C3** *has symmetry* 2.

Proof. **C2** $= (0^{i-1}, 1, 0, \overline{R}, 1)$ and **C3** $= (0^i, 1^{j+1}, 0, R', 1)$ and $R = (1^{j-1}, 0, 1, R')$, $i > 1$ and $j > 0$.

By Symmetry of **C3**, $(1^{j+1}, 0, R', 1) = (Y, 0^i, Z)$ with Y, Z palindromes. Moreover, by superminimality of **C2**, R' cannot contain a sequence of i consecutive 0 nor finishes by 0^{i-1}.

Hence, $R' = (0^{i-1}, U)$ which contradicts the definition of **C3**. ■ ■

Fact 11 *If* **C2** *has symmetry* 2 *and* **C3** *has symmetry* 1*, then* **C2** $\in S_1 \setminus S_3$.

Proof. By symmetry with Fact 7. ■

Fact 12 **C2** *and* **C3** *cannot have both symmetry* 2.

Proof. **C2** $= (0^{i-1}, 1, 0, \overline{R}, 1)$ and **C3** $= (0^i, 1^{j+1}, 0, R', 1)$ and $R = (1^{j-1}, 0, 1, R')$, $i > 1$ and $j > 0$.

By Symmetry of **C3**, $(1^{j+1}, 0, R', 1) = (Y, 0^i, Z)$ with Y, Z palindromes. Moreover, by superminimality of **C2**, R' cannot contain a sequence of i consecutive 0 nor finishes by 0^{i-1}.

Hence, $R' = (0^{i-1}, U)$ which contradicts the definition of **C3**. ■ ■

■ ■

Lemma 8 *Let* $\mathcal{C}$ *be a configuration in* $S_1 \setminus S_3$ *with supermin* $\mathcal{C}^{\min} = (0^i, 1, R)$*,* $i > 1$*, and let* $\mathcal{C}'$ *the configuration obtained by applying* REDUCE$_2$ *on only one robot on* $\mathcal{C}$*. Then* $\mathcal{C}'$ *is asymmetric and it is not adjacent with respect to* REDUCE$_1$ *and* REDUCE$_2$ *to any symmetric configuration different from* $\mathcal{C}$*.*

Proof. We first show that $\mathcal{C}'$ is asymmetric and that cannot it be obtained by applying REDUCE$_1$ on a configuration different from $\mathcal{C}$

Note that if $\mathcal{C}'^{\min} = (0^i, 1^j, 0, X, 1)$, then it can be obtained by performing REDUCE$_1$ on a configuration C'' such that (i) $C'' = (0^{i-1}, 1, 0, 1^{j-1}, 0, X, 1)$, or (ii) $C'' = (0^{i-1}, 1^j, 0, X, 0, 1)$, or (iii) $C'' = (0^i, 1, 0, 1^{j+1}, 0, X', 1)$, where this last case can occur only if $X = (1, X')$ for some X'.

We obtain the following cases.

- $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0, R')$. In this case, $\mathcal{C}'$ has a representation $\mathcal{C}' = (0^i, 1, 0^{i+1}, 1, R')$ with $\mathcal{C}'^{\min} = (0^{i+1}, 1, R', 0^i, 1)$ or $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^i, \overline{R'}, 1)$ since there is only one sequence of $i+1$. Therefore, $\mathcal{C}'$ can have only symmetry 1. However, this implies that R' starts with 0^i which is a contradiction to the superminimality of $\mathcal{C}^{\min}$ as in this case $\mathcal{C}^{\min}$ contains a sequence of $i+1$ consecutive occupied nodes. This also implies that $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^i, \overline{R'}, 1)$.
  If C' can been obtained by applying REDUCE$_1$ on a configuration C'' different from C, then (i) $C'' = (0^i, 1, 0^{i+1}, \overline{R'}, 1)$, or (ii) $C'' = (0^i, 1, 0^i, \overline{R'}, 0, 1)$. The case (iii) cannot occur as in this case X starts with 0. In the first case the step from C'' to C' does not correspond to REDUCE$_1$, in the second case we obtain $C'' = \overline{C}$ in any case we get a contradiction.
- $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1^j, 0, R')$, $j > 1$. In this case, $\mathcal{C}'^{\min} = (0^i, 1, 0^i, 1^{j-1}, 0, 1, R')$. Moreover, such sequence is the only supermin sequence as otherwise we obtain a contradiction to the superminimality of $\mathcal{C}^{\min}$. Therefore, C' is asymmetric. If C' has been obtained by applying REDUCE$_1$ on a configuration C'' different from C, then (i) $C'' = (0^{i-1}, 1, 0^{i+1}, 1^{j-1}, 0, 1, R')$, or (ii) $C'' = (0^{i-1}, 1, 0^i, 1^{j-1}, 0, 1, R'', 0, 1)$ with $R'' = (R', 1)$. The case (iii) cannot occur as in this case X starts with 0. In both cases (i) and (ii) the step from C'' to C' does not correspond to REDUCE$_1$, a contradiction.
- $\mathcal{C}^{\min} = (0^i, 1^j, 0^x, 1^{j'}, 0, R')$, $x \leq i$, $j > 0$, and $j' > 0$. We exclude the case $x = i$ and $j = 1$ because it has been already analyzed. In this case, $\mathcal{C}' = (0^i, 1^j, 0^x, 1^{j'-1}, 0, 1, R')$.
  If $x < i - 1$ or $j' > 1$, then $\mathcal{C}'^{\min} = (0^i, 1^j, 0^x, 1^{j'-1}, 0, 1, R')$. Moreover, such sequence is the only supermin sequence as otherwise we obtain a contradiction to the superminimality of $\mathcal{C}^{\min}$. Therefore, C' is asymmetric. If C' has been obtained by applying REDUCE$_1$ on a configuration C'' different from C, then (i) $\mathcal{C}'' = (0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'-1}, 0, 1, R')$, (ii) $\mathcal{C}'' = (0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'-1}, 0, 1, R'', 0, 1)$ with $R' = (R'', 1)$, or (iii) $\mathcal{C}'' = (0^i, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R')$ where $x = 1$ and $j' > 1$ as otherwise such case cannot occur. In cases (i) and (ii) R' contains a sequence of i consecutive occupied nodes and hence the step from C'' to C' does not correspond to REDUCE$_1$, a contradiction. In case (iii) $\mathcal{C}''^{\min} \neq (0^i, 1^{j+1}, 0, 1^{j'-1}, 0, 1, R')$ as the superminimality of C implies that either R' contains a sequence $(0, 1^j, 0^i)$ or it finishes by $(0, 1^j)$. Therefore also in this case, the step from C'' to C' does not correspond to REDUCE$_1$.
  If $x = i - 1$ and $j' = 1$, then $\mathcal{C}^{\min} = (0^i, 1^j, 0^{i-1}, 1, 0, R')$ then $\mathcal{C} \in S_3$ since R'm must finish by 1.
  If $x = 1$, $j' = 1$, and $j > 1$, then $\mathcal{C}^{\min} = (0^i, 1^j, 0^i, 1, 0, R')$ is a contradiction as $(\mathcal{C}^{\min})_{i+j} < \mathcal{C}^{\min}$.

We conclude the proof by showing that $\mathcal{C}'$ cannot be obtained by applying REDUCE$_2$ on a configuration different from $\mathcal{C}$.

Let us assume that $\mathcal{C}^{\min} = (0^i, 1^j, 0^x, 1^{j'}, 0^y, 1, X)$. After performing REDUCE$_2$ on $\mathcal{C}$ we have $\mathcal{C}' = (0^i, 1^j, 0^x, 1^{j'-1}, 0, 1, 0^{y-1}, 1, X)$. Let us assume that $\mathcal{C}'$ can be obtained by performing REDUCE$_2$ on symmetric configuration $\mathcal{C}''$ different from $\mathcal{C}$. The following cases may arise.

- $X = (X', 0^i, 1^{j''})$ and the supermin of $\mathcal{C}''$ starts from the sequence of i consecutive occupied nodes in X. In this case we have that either $\mathcal{C}'' = (0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'-1}, 0, 1, 0^{y-1}, 1, X', 0^i, 1^{j''})$ or $x = 1$ and $\mathcal{C}' = (0^i, 1^{j+1}, 0, 1^{j'-2}, 0, 1, 0^{y-1}, 1, X', 0^i, 1^{j''})$. The supermin of $\mathcal{C}''$ is either $\mathcal{C}''^{\min} = (0^i, 1^{j''}, 0^{i-1}, 1, 0, 1^{j-1}, 0^x, 1^{j'-1}, 0, 1, 0^{y-1}, 1, X')$ or $\mathcal{C}''^{\min} = (0^i, 1^{j''}, 0^i, 1^{j+1}, 0, 1^{j'-2}, 0, 1, 0^{y-1}, 1, X')$, respectively. In any case, we must have that $j'' > j$ otherwise we obtain a contradiction to the superminimality of $\mathcal{C}^{\min}$. It follows that $\mathcal{C}$ has symmetry 2 and hence X' contains a sequence $(0^i, 1^j, 0)$ which is a contradiction to the superminimality of $\mathcal{C}''^{\min}$.
- The supermin of $\mathcal{C}''$ starts from the same node as $\mathcal{C}'$. In this case $\mathcal{C}''^{\min} = (0^i, 1^j, 0^{x-1}, 1, 0, 1^{j'-2}, 0, 1, 0^{y-1}, 1, X)$ but in $\mathcal{C}$ must exist another sequence which starts by $(0^i, 1^j, 0^x)$. As $\mathcal{C} \notin \mathcal{S}_\ni$, such a sequence is still in $\mathcal{C}''$ and induces a view which is smaller than $\mathcal{C}''^{\min}$, a contradiction.
- The supermin of $\mathcal{C}''$ starts from the sequence consecutive occupied nodes corresponding to position $i + j$ of $\mathcal{C}$. Three cases may arise.
  - $x = i$, $y = 2$ and $j' - 1 < j$. In this case $\mathcal{C}'' = (0^i, 1^j, 0^i, 1^{j'-1}, 0, 1, 1, 0, X)$ and $\mathcal{C}''^{\min} = (0^i, 1^{j'-1}, 0, 1, 1, 0, X, 0^i, 1^j)$. As $j' \geq j$ by the superminimality of $\mathcal{C}^{\min}$, then $j' = j$ and hence $\mathcal{C}''^{\min} = (0^i, 1^{j-1}, 0, 1, 1, 0, X, 0^i, 1^j)$. It follows that either X contains a sequence $(0^i, 1^{j-1}, 0)$ or that X starts by 0^{i-1} and $j = 3$. In the first case we obtain a contradiction with the

superminimality of $\mathcal{C}^{\min}$, in the second case the step from C'' to C' does not correspond to REDUCE$_2$, a contradiction.

- $x = i-1$, $j' = 1$, and $j > 1$. In this case, $\mathcal{C}' = (0^i, 1^j, 0^i, 1, 0^{y-1}, X)$ with $\mathcal{C}'^{\min} = (0^i, 1, 0^{y-1}, X, 0^i, 1^j)$. We assume that $X = (1^{j''}, 0, X')$ which implies that $\mathcal{C}''^{\min} = (0^i, 1, 0^{y-1}, 1^{j''-1}, 0, 1, X', 0^i$ In order to be symmetric, $\mathcal{C}''^{\min}$ must contain another sequence starting by $(0^i, 1, 0)$ but this implies a contradiction to the superminimality of $\mathcal{C}^{\min}$.
- $x = i-1$, $j' = 1$, and $j > 1$. In this case, $\mathcal{C}' = (0^i, 1^j, 0^{i+1}, 1, 0^{y-1}, X)$ and $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^{y-1}, X, 0^i, 1^j)$. again we assume that $X = (1^{j''}, 0, X')$ and this implies that $\mathcal{C}''^{\min} = (0^{i+1}, 1, 0^{y-1}, 1^{j''-1}, 0, 1, X', 0^i, 1^j)$ which cannot be symmetric as there is only one sequence of $i+1$ consecutive occupied nodes and $j > 1$. ■

■

Lemma 9 *Let $\mathcal{C}$ be a configuration in S_2, or $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, or $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$ and let $\mathcal{C}'$ the configuration obtained by applying* REDUCE$_{-1}$ *on only one robot on $\mathcal{C}$. Then $\mathcal{C}'$ is asymmetric and it is not adjacent with respect to* REDUCE$_1$, REDUCE$_2$, *and* REDUCE$_{-1}$ *to any symmetric configuration different from $\mathcal{C}$.*

Proof. If $\mathcal{C}^{\min} = (0^i, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 0, 1)$, then $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 1)$. It is easy to see that $\mathcal{C}'$ is asymmetric since there is only one sequence of $i+1$ consecutive occupied nodes. If $\mathcal{C}'$ can be obtained by applying REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}$, then $\mathcal{C}''^{\min} = (0^i, 1, 0, 0, 0, 1, 0^i, (1, 0, 1, 0^i)^\ell, 1, 1)$, or $\mathcal{C}''^{\min} = (0^{i+1}, 1, 0, 1, 0^{i+1}, (1, 0, 1, 0^i)^\ell, 1, 1)$. In any case $\mathcal{C}''$ is asymmetric.

If $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-1}, 1)$, then $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^i, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0$ Therefore, $\mathcal{C}'^{\min}$ can be obtained by applying REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}$ only if $\mathcal{C}'' = (0^i, 1, 0^{i+1}, 1, 0^i, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-2}, 1)$ or $\mathcal{C}'' = (0^{i+1}, 1, 0^{i-1}, 1, 0^{i+1}, 1, 0^i, (1, 0^{i-1}, 1, 0^i, 1, 0^i)^\ell, 1, 0^{i-2}, 1)$. In any case $\mathcal{C}''$ is asymmetric.

If $\mathcal{C}^{\min} \in S_2$, then $\mathcal{C}^{\min} = (0^i, 1^j, 0^i, Z)$ with $Z = \overline{Z}$ and $j \geq 1$. Let us assume w.l.o.g. that $Z = (1^{j'}, 0, Z', 0, 1^{j'})$ with $Z' = \overline{Z'}$ and $j' \geq j$. Then, $\mathcal{C}^{\min} = (0^i, 1^j, 0^i, 1^{j'}, 0, Z', 0, 1^{j'})$. We distinguish the following cases

- $j' = 1$. In this case $j = 1$ and hence $\mathcal{C}^{\min} = (0^i, 1, 0^i, 1, 0, Z', 0, 1)$ and $\mathcal{C}'^{\min} = (0^{i+1}, 1, 0^i, 1, 0, Z', 1)$. By Lemma 4, $\mathcal{C}'$ can be symmetric only if $Z' = (0, 1, 0)^\ell$ and $i = 1$, a contradiction.
  If $\mathcal{C}'$ can be obtained by applying REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}$, then $\mathcal{C}'' = (0^i, 1, 0^{i+1}, 1, 0, Z', 1)$, or $\mathcal{C}'' = (0^{i+1}, 1, 0^{i-1}, 1, 0, 0, Z', 1)$, or $\mathcal{C}'' = (0^{i+1}, 1, 0^i, 1, 1, Z'', 1, 1)$ where $Z' = (1, Z'', 1)$. In the first case, the step from $\mathcal{C}''$ to $\mathcal{C}'$ does not correspond to any of REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$. In the second case $\mathcal{C}''$ is symmetric only if $Z' = ((0, 0, 1, 0, 0)^\ell)$ and $i = 3$, in which case $\mathcal{C}^{\min} = (0, 0, 0, 1, 0, 0, 0, 1, 0, (0, 0, 1, 0, 0)^\ell, 0, 1)$ which is periodic. In the third case $\mathcal{C}''$ is asymmetric.
- $j' > 1$. In this case $\mathcal{C}' = (0^i, 1^j, 0^i, 1^{j'}, 0, Z', 1, 0, 1^{j'-1})$.
  If $j' = j$, then $\mathcal{C}'^{\min} = (0^i, 1^{j-1}, 0, 1, Z', 0, 1^j, 0^i, 1^j)$. It follows that $\mathcal{C}'$ is asymmetric and that it can be obtained by applying REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}$ only if: $\mathcal{C}'' = (0^{i-1}, 1, 0, 1^{j-2}, 0, 1, Z', 0, 1^j, 0^i, 1^j)$ or $\mathcal{C}'' = (0^{i-1}, 1^{j-1}, 0, 1, Z', 0, 1^j, 0^i, 1^{j-1}, 0, 1)$. In the first case $\mathcal{C}''$ is asymmetric, in the second case the step from $\mathcal{C}''$ to $\mathcal{C}'$ does not correspond to any of REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$.
  If $j' > j$, then $\mathcal{C}'^{\min} = (0^i, 1^j, 0^i, 1^{j'-1}, 0, 1, Z', 1, 0, 1^{j'})$ and $\mathcal{C}'$ is asymmetric as another supermin would imply a contradiction to the superminimality of $\mathcal{C}$. $\mathcal{C}'$ can be obtained by applying REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$ on a configuration $\mathcal{C}''$ different from $\mathcal{C}$ only if: $\mathcal{C}'' = (0^{i-1}, 1, 0, 1^{j-1}, 0^i, 1^{j'-1}, 0, 1, Z', 1, 0, 1^{j'})$, or $\mathcal{C}'' = (0^{i-1}, 1^j, 0^i, 1^{j'-1}, 0, 1, Z', 1, 0, 1^{j'-1}, 0, 1)$ or $\mathcal{C}'^{\min} = (0^i, 1^j, 0^{i-1}, 1, 0, 1^{j'-2}, 0, 1, Z', 1, 0, 1^{j'})$. In any case the step from $\mathcal{C}''$ to $\mathcal{C}'$ does not correspond to any of REDUCE$_1$, or REDUCE$_2$, or REDUCE$_{-1}$. ■

■

Lemma 10 *Let $\mathcal{C}$ be a configuration with supermin $\mathcal{C}^{\min} = (0^i, 1, 0, 1, 0^x, 1, 0, 1)$, $i > 1$ and $x < i$, and let $\mathcal{C}'$ the configuration obtained by applying* REDUCE$_2$ *on only one robot on $\mathcal{C}$. Then $\mathcal{C}'$ is asymmetric and it is not adjacent with respect to* REDUCE$_1$ *and* REDUCE$_2$ *to any symmetric configuration different from $\mathcal{C}$.*

Proof. The configuration obtained by applying REDUCE$_2$ on only one robot on $\mathcal{C}$ is $\mathcal{C}'^{\min} = (0^i, 1, 0, 0, 1, 0^{x-1}, 1, 0, 1)$ which is asymmetric as there exists only one sequence of i consecutive occupied nodes and the axis of symmetry cannot pass in the middle of it. Let us assume that $\mathcal{C}'$ can be obtained by applying REDUCE$_1$ or REDUCE$_2$ on a configuration different from $\mathcal{C}$. Then two cases may arise.

- $\mathcal{C}'' = (0^{i-1}, 1, 0, 0, 0, 1, 0^{x-1}, 1, 0, 1)$. In this first case either $\mathcal{C}''^{\min} = (0^{i-1}, 1, 0, 0, 0, 1, 0^{x-1}, 1, 0, 1)$ or $\mathcal{C}''^{\min} = (0, 0, 0, 1, 0^{i-1}, 1, 0, 1, 0^{x-1}, 1)$. In the first case $\mathcal{C}''$ is asymmetric and in the second case the step from C'' to C' does not correspond to REDUCE$_1$ or to REDUCE$_2$.
- $\mathcal{C}'' = (0^{i-1}, 1, 0, 0, 1, 0^{x-1}, 1, 0, 0, 1)$. In this case, if $i-1 > 2$, then $\mathcal{C}''^{\min} = (0^{i-1}, 1, 0, 0, 1, 0^{x-1}, 1, 0, 0, 1)$ but the step from C'' to C' does not correspond to REDUCE$_1$ or to REDUCE$_2$. If $i - 1 = 2$ then $\mathcal{C}''^{\min} = (0, 0, 1, 0, 0, 1, 0, 0, 1, 0^{x-1}, 1)$ and step from C'' to C' does not correspond to REDUCE$_1$, moreover REDUCE$_2$ cannot be performed on $\mathcal{C}''$ has it is in S_3. If $i - 1 = 1$, then $x = 1$, $\mathcal{C}'' = (0, 1, 0, 0, 1, 1, 0, 0, 1)$ and $\mathcal{C}''^{\min} = (0, 0, 1, 0, 1, 0, 0, 1, 1)$ and hence the step from C'' to C' does not correspond to REDUCE$_1$ or to REDUCE$_2$. ■

■

A.5 Searching in a Ring

In this section, we give more details on the algorithm to perpetually search a ring. First, we recall some results concerning the perpetual searching of a ring in the Look-Compute-Move model. In [**?**], it is shown that, for any starting configuration, it is not possible to search an n-node ring using k robots if $n \leq 9$, $k \leq 3$, or $k \geq n-2$. On the other hand, there is an algorithm allowing $5 \leq k \leq n-3$ robots to search a ring with $n \geq 10$ nodes (except for $k = 5$ and $n = 10$), if the initial configuration is rigid. Moreover, in [**?**] an algorithm for $k = n-3$ is given.

If k is even and the axis does not pass through an empty node, the searching is clearly unsolvable because a synchronous execution of any algorithm either cause a collision in the node lying on the axis or does not allow to search the edges incident to such node. It follows that in the searching problem, the symmetric allowed configurations are all those with k odd and those with k even where the axis does not pass through an empty node, provided that $3 < k < n-2$ and $n > 9$. Moreover, allowed configurations must be exclusive.

Here, we improve over the algorithm in [**?**] by addressing also aperiodic symmetric configurations. More precisely, our new algorithm works for any aperiodic allowed configuration such that $k \neq 4$, and if $n = 10$, $k \notin \{5, 6\}$. For asymmetric configuration, the case $(n, k) = (10, 6)$ is feasible, therefore the only cases left open are $k = 4$, $(n, k) = (10, 5)$, and $(n, k) = (10, 6)$ symmetric.

The algorithm given in this section exploits algorithm Align to reach one of the configurations $\mathcal{C}^a$, $\mathcal{C}^b$, or $\mathcal{C}^c$. First, we give two procedures that are used after Align to achieve configuration $\mathcal{C}^b$, in the case Align reaches configuration $\mathcal{C}^c$, and to achieve $\mathcal{C}^a$ in the case that k is odd.

Algorithm Compact-Align. We now define an algorithm that complements Algorithm Align given in the previous section. In detail, Algorithm Compact-Align is applied until one of the configurations $\mathcal{C}^a$ (all robots occupy consecutive nodes) or $\mathcal{C}^b$ (all robots but one occupy consecutive nodes) is achieved.

Algorithm Compact-Align first applies Align. Then, either a configuration $\mathcal{C}^a$ or $\mathcal{C}^b$ is achieved, in which case we are done, or a configuration in the set $\mathcal{R}$ defined below is achieved. Since the configuration $\mathcal{C}^c$ (the robots are divided into two segments of consecutive nodes) belongs to $\mathcal{R}$, the specifications of Align (it achieves either $\mathcal{C}^a$ or $\mathcal{C}^b$ or $\mathcal{C}^c$ and it may reach $\mathcal{C}^c$ only if k and n are even) ensure that such a configuration is eventually reached. Finally, from any configuration in $\mathcal{R}$, Algorithm Compact-Align allows the robots to achieve either $\mathcal{C}^a$ or $\mathcal{C}^b$.

If $\mathcal{C} \in \mathcal{R}$, then both k and n are even. Recall that any allowed configuration in $\mathcal{C}^c$ has the following form: $(0^{\frac{k}{2}}, 1^j, 0^{\frac{k}{2}}, 1^{n-k-j})$, with $0 < j < \frac{n-k}{2}$ and j even. For any $0 \leq a < b$ with a and $b = n-k-a-2$ even (in particular $b > 1$), let us define $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ as the set of all configurations $\mathcal{C} = (u_0, \cdots, u_{n-1})$ with the following form:

- $\mathcal{R}_1(a, b, c) = (0^{k/2-c}, 1, 0^c, 1^a, 0^c, 1, 0^{k/2-c}, 1^b)$, where $0 \leq c < k/2$. Note that $\mathcal{R}_1(a, b, 0) \in \mathcal{C}^c$. Moreover, any configuration $\mathcal{R}_1(a, b, c)$ is symmetric with one unique axis (because $a < b$) and this axis does not pass through an empty node (because a and b are even). Moreover, $u_{k/2-c-1}$ and u_{n-1-b} can be identified because $a < b$ and $b > 1$.
- $\mathcal{R}_2(a, b, c) = (0^{k/2-c-1}, 1, 0^{c+1}, 1^a, 0^c, 1, 0^{k/2-c}, 1^b)$, where $0 \leq c < k/2$. Such a configuration is asymmetric and $u_{n-b-(k/2-c)}$ can be identified. Note also that $\mathcal{R}_2(0, b, k/2-1) \in \mathcal{C}^b$.

If $\mathcal{C} \in \mathcal{R}_1(a, b, c)$ for some a, b, and c, then Compact-Align moves the robot at $u_{k/2-c-1}$ to $u_{k/2-c}$. Otherwise, if $\mathcal{C} \in \mathcal{R}_2(a, b, c)$ for some a, b, and c, then it moves the robot at $u_{n-b-(k/2-c)}$ to $u_{n-b-(k/2-c-1)}$, otherwise it applies Align.

Since a and b are even, then $b \geq a + 2$, and therefore we can check that any configuration $\mathcal{C}$ in $\mathcal{R}$ is not adjacent to any other allowed symmetric configuration. Indeed, if $\mathcal{C}$ is adjacent to an allowed and symmetric configuration, then the axis must pass through the unique long segment of at least $b-1$ consecutive 1's, and it is easy to check that such a configuration would not be allowed because b is even. Therefore, there is no conflict between the moves done if $\mathcal{C} \in \mathcal{R}$ and the moves done by Align when

$\mathcal{C} \notin \mathcal{R}$.

Procedure: Compact-Align
Input: Allowed configuration $\mathcal{C} = (u_0, \cdots, u_{n-1})$, with an even number of robots.

if $\mathcal{C} \in \mathcal{R}$ **then**
  if $\mathcal{C} = (0^{k/2-c}, 1, 0^c, 1^a, 0^c, 1, 0^{k/2-c}, 1^b)$; // $\mathcal{C} \in \mathcal{R}_1$
  then
    Robot at $u_{k/2-c-1}$ moves to $u_{k/2-c}$; // Two symmetrical moves
  if $\mathcal{C} = (0^{k/2-c-1}, 1, 0^{c+1}, 1^a, 0^c, 1, 0^{k/2-c}, 1^b)$; // $\mathcal{C} \in \mathcal{R}_2$
  then
    Robot at $u_{n-b-(k/2-c)}$ moves to $u_{n-b-(k/2-c-1)}$; // unique move performed
else
  Apply Align

Figure 7: Algorithm Compact-Align.

Algorithm Break-Symmetry. In the following, we give an algorithm that allows an odd number k of robots to eventually reach an asymmetric configuration. More precisely, the algorithm first applies Algorithm Align. Then, when all k robots are occupying consecutive nodes, they move to reach a symmetric configuration where one robot is on the axis and has its two neighbors that are empty. The robot on the axis moves to one of its neighbors, breaking the symmetry.

Let $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$ be the set of all configurations $\mathcal{C} = (u_0, \cdots, u_{n-1})$ with the following form:

- $\mathcal{B}_1(\ell) = (0^\ell, 1, 0^{k-2\ell}, 1, 0^\ell, 1^{n-k-2})$, where $0 \leq \ell \leq \lfloor k/2 \rfloor$. Moreover, any configuration $\mathcal{R}_1(\ell)$ is symmetric with one unique axis and nodes $u_{\ell+1}$ and $u_{k-\ell+1}$ can be univocally identified.
- $\mathcal{B}_2(\ell) = (0^{\ell-1}, 1, 0^{k-2\ell+1}, 1, 0^\ell, 1^{n-k-2})$, where $0 < \ell \leq \lfloor k/2 \rfloor$. Such a configuration is asymmetric and u_ℓ can be univocally identified.

When a configuration $\mathcal{C}$ is in $\mathcal{B}_1(\ell)$ for some ℓ, then Break-Symmetry moves the robot at $u_{\ell+1}$ to u_ℓ. When $\mathcal{C}$ is in $\mathcal{B}_1(\ell)$ for some ℓ, then Break-Symmetry moves robot at u_ℓ to $u_{\ell-1}$. Eventually, configuration $\mathcal{C} = (0^{\lfloor k/2 \rfloor}, 1, 0, 1, 0^{\lfloor k/2 \rfloor}, 1^{n-k-2})$ is reached. At this point the robot at $u_{\lfloor k/2 \rfloor+1}$ moves to $u_{\lfloor k/2 \rfloor}$ (or arbitrarily to $u_{\lfloor k/2 \rfloor+2}$). At this point the obtained configuration is asymmetric and applying algorithm Align leads to configuration $\mathcal{C}^a = (0^{k-1}, 1, 0, 1^{n-k-1})$. Finally, the robot at k moves to $k+1$. The obtained configuration is suitable to be used in the algorithm of [?] for graph searching. As any asymmetric configuration in $\mathcal{B}_2$ is not adjacent to any symmetric configuration not in $\mathcal{B}_1$, there are no conflicts between the moves of Align, those of the algorithm in [?] and those of Break-Symmetry.

Procedure: Break-Symmetry
Input: Exclusive configuration $\mathcal{C} = (u_0, \cdots, u_{n-1})$, with $\sum_i u_i = k$ robots, such that k is odd and $\mathcal{C}$ has at most one axis of symmetry and, if any, this axis does not pass through an empty node.

if $\mathcal{C} \in \mathcal{B}$ **then**
  if $\mathcal{C} = (0^{\lfloor k/2 \rfloor}, 1, 0, 1, 0^{\lfloor k/2 \rfloor}, 1^{n-k-2})$; // $\mathcal{C} = \mathcal{B}_1(\lfloor k/2 \rfloor)$
  then
    Robot at $u_{\lfloor k/2 \rfloor+1}$ moves to $u_{\lfloor k/2 \rfloor}$ (or symmetrically to $u_{\lfloor k/2 \rfloor+2}$) ; // symmetry is broken
  if $\mathcal{C} = (0^\ell, 1, 0^{k-2\ell}, 1, 0^\ell, 1^{n-k-2})$ *where* $0 \leq \ell < \lfloor k/2 \rfloor$; // $\mathcal{C} \in \mathcal{B}_1$
  then
    Robot at $u_{\ell+1}$ moves to u_ℓ; // Two symmetrical moves
  if $\mathcal{C} = (0^{\ell-1}, 1, 0^{k-2\ell+1}, 1, 0^\ell, 1^{n-k-2})$ *where* $0 < \ell \leq \lfloor k/2 \rfloor$; // $\mathcal{C} \in \mathcal{B}_2$
  then
    Robot at u_ℓ moves to $u_{\ell-1}$; // unique move performed
else
  Apply Align

Figure 8: Algorithm Break-Symmetry.

Algorithm Search-Ring. Algorithm Search-Ring first checks whether $k = n - 3$ or if n is odd and k is even. In the affermative case, any allowed configuration must be asymmetric, and therefore the algorithm of [?] can be applied and the ring is searched.

If k is odd, we first use Algorithm Break-Symmetry to break the potential symmetry and then use the algorithm of [**?**]. Each of these configurations used during the searching phase of the algorithm of [**?**] are asymmetric and are not adjacent to any symmetric configuration reached by Algorithm Break-Symmetry. Therefore, there is no ambiguity (no pending move) when a robot recognizes such a configuration.

If n and k are even, we may be in allowed symmetric configurations and therefore the Algorithm Search-Ring proceeds in two phases. Algorithm Compact-Align is first applied until one of the configurations in $\mathcal{A}$ (described in Appendix) is achieved. This is guaranteed by the fact that both $\mathcal{C}^a$ and $\mathcal{C}^b$ belong to $\mathcal{A}$. Then, the algorithm proceeds to Phase 2 which actually performs the searching.

Set $\mathcal{A}$ of configurations. We now define the set $\mathcal{A}$ of configurations required to define the algorithm for Phase 2. We consider the following hypothesis: $n-k$ is even, $n-k \geq 4$, $k \geq 6$, $n \geq 10$ and, if $k=6$ then $n \geq 11$. The set $\mathcal{A}$ is defined as the set of all configurations $\mathcal{C} = (u_0, \cdots, u_{n-1})$ with the following forms:

$\mathcal{A}$-**a**$(\ell) = (0^{k-2}, 1^{\ell}, 0, 1^{n-2\ell-k}, 0, 1^{\ell})$, $0 \leq \ell \leq (n-k)/2$. Note that $\mathcal{A}$-a$(0) = \mathcal{C}^a$.
In this case, $\mathcal{C}$ is symmetric with a unique axis because $k-2 > 1$. This axis does not pass through an empty node because $n-k$ is even. Clearly, nodes $u_{k-2+\ell}$ and $u_{n-1-\ell}$ can be identified as occupied and adjacent to one (case $\ell = 0$) or two (case $0 < \ell < (n-k)/2$) empty nodes, or (case $\ell = (n-k)/2$) they form the unique (because $k \geq 5$) segment of exactly two consecutive occupied nodes.

$\mathcal{A}$-**b**$(\ell) = (0^{k-2}, 1^{\ell}, 0, 1^{n-2\ell-k-1}, 0, 1^{\ell+1})$, $0 \leq \ell < (n-k)/2$.
In this case, $\mathcal{C}$ is asymmetric for any $0 \leq \ell \leq (n-k)/2$. In particular, if $\ell = 0$, it is asymmetric because $n-k \geq 4$. Then, $u_{k-2+\ell}$ can be identified.

$\mathcal{A}$-**c** $= (0^{k-3}, 1, 0, 1^{(n-k)/2-1}, 0, 0, 1^{(n-k)/2})$.
In this case, $\mathcal{C}$ is asymmetric, because $k \geq 6$ and $n-k \geq 4$. Then, u_0 can be identified.

$\mathcal{A}$-**d**$(\ell) = (0^{k-4}, 1, 0, 1^{(n-k)/2-1-\ell}, 0, 1^{2\ell}, 0, 1^{(n-k)/2-\ell-1}, 0, 1)$, $0 \leq \ell \leq (n-k)/2-1$.
In this case, $\mathcal{C}$ is symmetric with one unique axis not passing through an empty node. Indeed, it is easy to check if $k \neq 6$. If $k=6$ and $\ell = 0$, it is true because $n > 10$.
Then, $u_{(n+k)/2-3}$ and $u_{(n+k)/2-2}$ can be identified as the single segment of two occupied nodes (if $k > 6$) and as the single segment of two occupied nodes adjacent to segments of more than one empty node (if $k=6$ and $n > 10$).

$\mathcal{A}$-**e**$(\ell) = (0^{k-4}, 1, 0, 1^{(n-k)/2-1-\ell}, 0, 1^{2\ell+1}, 0, 1^{(n-k)/2-\ell-2}, 0, 1)$, $0 \leq \ell \leq (n-k)/2-2$.
In this case, $\mathcal{C}$ is asymmetric (this is true in particular, when $\ell = 0$, because if $k=6$ then $n > 10$)
Then, $u_{(n+k)/2-3-\ell}$ can be identified.

$\mathcal{A}$-**f** $= (0^{k-3}, 1, 0, 0, 1^{n-k-2}, 0, 1)$
In this case, $\mathcal{C}$ is asymmetric because $k \geq 5$ and $n-k \geq 4$. Then, u_{k-2} can be identified.

The pseudo-code of Algorithm Search-Ring is given in Figure 9

The intuitive explication of the Searching algorithm (Phase 2) is as follows. All robots are aligned on consecutive nodes (configuration $\mathcal{A}$-a$(0) = \mathcal{C}^a$). Then, each of the two robots X and Y at the ends of this segment move (one clockwise and the other anti-clockwise) to meet on the two adjacent nodes opposite to the occupied segment (passing "alternatively" from configuration $\mathcal{A}$-$a(\ell)$ to configuration $\mathcal{A}$-$b(\ell)$ for $\ell = 0 \cdots (n-k)/2$). Then, the two robots X' and Y' occupying the ends of the "long" occupied segment move to their empty neighbor ($\mathcal{A}$-a$(n-2\ell-k)$ and $\mathcal{A}$-c). These moves are two indicate to X and Y that it is time to go back toward the "long" segment, and that is what happens (passing "alternatively" from configuration $\mathcal{A}$-$d(\ell)$ to configuration $\mathcal{A}$-$e(\ell)$ for $\ell = 0 \cdots (n-k)/2-2$). Finally, when X is adjacent to X' and Y is adjacent to Y', X' and Y' move to their empty neighbor (passing through the configuration $\mathcal{A}$-f) such that they re-integrate the segment. Then, Configuration $\mathcal{A}$-a(1) is achieved and the process is repeated perpetually.

It is easy to check that such a sequence of moves actually search the ring and, by definition of the configurations met during the process (configurations in $\mathcal{A}$), there is no ambiguity. In Algorithm 9, $\mathcal{O}$ denotes the set of configurations used during the searching phase of the Algorithm of [**?**].

The distinct configurations that can be achieved in Phase 2 are the ones in $\mathcal{A}$ and can be characterized succinctly such that they are pairwise distinguishable without ambiguity. Moreover, each of those configurations is either asymmetric and only one (identifiable) robot can move, or it is symmetric with one unique axis of symmetry and two (identifiable) symmetric robots move. In the latter case, when only one of these symmetric robots moves, then we reach an asymmetric configuration where the only robot permitted to move is the other one (i.e., the possible pending move and the permitted move coincide). Therefore there is never ambiguity in the choice of the robot(s) that must move.

The validity of algorithms for Phase 2 and the fact that they actually search the ring are easy to

Procedure: SEARCH-RING
Input: Exclusive configuration $\mathcal{C} = (u_0, \cdots, u_{n-1})$, with $\sum_i u_i = k$ robots, $k \geq 6$ and $n \geq 10$ and $(k, n) \neq (6, 10)$ and $n - k \geq 3$, and such that $\mathcal{C}$ has at most one axis of symmetry and, if any and if k is even, this axis does not pass through an empty node.

if $k = n - 3$ *or* $(n - k$ *is odd and* k *even)* **then**
  Apply Algorithm of [**?**]
else
  if k *is odd* **then**
    if $\mathcal{C} \in \mathcal{O}$*;* // $\mathcal{C}$ is asymmetric
    then
      Apply Algorithm of [**?**]
    else
      Apply BREAK-SYMMETRY($\mathcal{C}$)
  else
    if $\mathcal{C} \in \mathcal{A}$ **then**
      if $\mathcal{C} = (0^{k-2}, 1^{\ell}, 0, 1^{n-2\ell-k}, 0, 1^{\ell})$ *with* $0 \leq \ell \leq (n-k)/2$ *;* // $\mathcal{C} \in \mathcal{A}$-a
      OR $\mathcal{C} = (0^{k-2}, 1^{\ell}, 0, 1^{n-2\ell-k-1}, 0, 1^{\ell})$ *with* $0 \leq \ell < (n-k)/2$*;* // $\mathcal{C} \in \mathcal{A}$-b
      then
        The robot at $u_{k-2+\ell}$ moves to $u_{k-2+\ell+1}$; // two symmetrical moves if $\mathcal{C} \in \mathcal{A}$-a
      if $\mathcal{C} = (0^{k-2}, 1^{(n-k)/2}, 0, 0, 1^{(n-k)/2})$ *;* // $\mathcal{C} = \mathcal{A}$-a$((n-k)/2)$
      OR $\mathcal{C} = (0^{k-3}, 1, 0, 1^{(n-k)/2-1}, 0, 0, 1^{(n-k)/2})$ *;* // $\mathcal{C} = \mathcal{A}$-c
      then
        The robot at u_0 move to u_{n-1} ; // two symmetrical moves if $\mathcal{C} \in \mathcal{A}$-a
      if $\mathcal{C} = (0^{k-4}, 1, 0, 1^{(n-k)/2-1-\ell}, 0, 1^{2\ell}, 0, 1^{(n-k)/2-\ell-1}, 0, 1)$ *with* $0 \leq \ell \leq (n-k)/2 - 1$ *;*
      // $\mathcal{C} \in \mathcal{A}$-d
      OR $\mathcal{C} = (0^{k-4}, 1, 0, 1^{(n-k)/2-1-\ell}, 0, 1^{2\ell+1}, 0, 1^{(n-k)/2-\ell-2}, 0, 1)$ *with* $0 \leq \ell \leq (n-k)/2 - 2$ *;*
      // $\mathcal{C} \in \mathcal{A}$-e
      then
        The robot at $u_{(n+k)/2-3-\ell}$ moves to $u_{(n+k)/2-4-\ell}$; // two symmetrical moves if $\mathcal{C} \in \mathcal{A}$-d
      if $\mathcal{C} = (0^{k-4}, 1, 0, 0, 1^{n-k-2}, 0, 0, 1)$ *;* // $\mathcal{C} = \mathcal{A}$-d$((n-k)/2 - 1)$
      OR $\mathcal{C} = (0^{k-3}, 1, 0, 1^{n-k-3}, 0, 0, 1)$ *;* // $\mathcal{C} = \mathcal{A}$-f
      then
        The robot at u_{n-2} move to u_{n-1} ; // two symmetrical moves if $\mathcal{C} \in \mathcal{A}$-d
    else
      Apply COMPACT-ALIGN

Figure 9: Algorithm SEARCH-RING.

obtain. Therefore, to prove the correctness of Algorithm SEARCH-RING, it will be sufficient to prove that Phase 1 and Phase 2 are not in conflict (i.e., that robots can decide which phase to proceed). It is enough to note that any configuration in $\mathcal{A}$ is not adjacent to any symmetric configuration not in $\mathcal{A}$.

Theorem 4 *Let* $4 < k \leq n - 3$ *robots, standing in an* n*-node ring forming an allowed configuration, Algorithm* SEARCH-RING *perpetually searches the ring, but for* $n = 10$ *and* $k = 5$*, and for* $n = 10$*,* $k = 6$ *when the configuration is symmetric.*

B Gathering in a ring

In this section, we provide the full strategy for achieving the gathering

The allowed symmetric configurations for the gathering are those aperiodic, without an axis of symmetry passing through two edges.

We make use of procedure ALIGN to reach one of the following configuration types: $\mathcal{C}^a = (0^{k-1}, 1, 0, 1^{n-k-1})$, $\mathcal{C}^b = (0^k, 1^{n-k})$, with k or n odd, $\mathcal{C}^c = (0^{\frac{k}{2}}, 1^j, 0^{\frac{k}{2}}, 1^{n-k-j})$, with j or n odd.

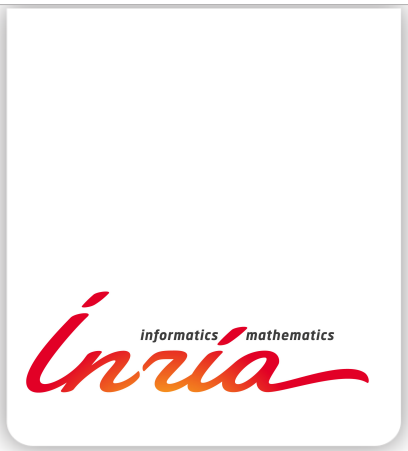
Actually, algorithm ALIGN terminates when either the obtained configuration belongs to one of the three types above, or to the types of configurations generated by Algorithm GATHERING as we are going

to describe.

If the initial configuration has both k and n even, then ALIGN either reaches a configuration of type $\mathcal{C}^a$ or of type $\mathcal{C}^c$ with j odd. In the former case, algorithm GATHERING leads to $\mathcal{C}^d = (0^{k-1}, 1, 1, 0, 1^{n-k-2})$. As $k < n-4$, then $\mathcal{C}^d$ is asymmetric and it is not adjacent to any possible symmetric configuration with respect to any move of one single robot. From $\mathcal{C}^d$, Algorithm GATHERING performs REDUCE$_0$, hence creating a multiplicity, and still obtaining a configuration of type $\mathcal{C}^d$. This process is repeated until only two nodes remain occupied. At this point, only one of the two occupied nodes contains a multiplicity, while the other contains one single robot. The single robot will be the only one permitted to move towards the other occupied node until the gathering is accomplished. In the latter case, that is, from $\mathcal{C}^c$ with j odd, algorithm GATHERING leads to a configuration of type $\mathcal{C}^c$ with $j = 1$. This is achieved by iterating move COMPACT$_0$ as defined below. Let $\mathcal{C} = (v_0, v_1, \ldots, v_n)$ be a configuration of type $(0^{\frac{k}{2}-i}, 1, 0^i, 1^j, 0^i, 1, 0^{\frac{k}{2}-i}, 1^{n-k-j-2})$ where $1 \leq i \leq \frac{k}{2}$ and $j < \frac{n-k-2}{2}$. Note that for $i = \frac{k}{2}$, $\mathcal{C} = \mathcal{C}^c$. Move COMPACT$_0$ consists in moving the robot at $v_{\frac{k}{2}-i-1}$ towards $v_{\frac{k}{2}-i}$. As $\mathcal{C}$ is symmetric, move COMPACT$_0$ permits two robots to move. If both move synchronously, the out-coming configuration $\mathcal{C}'$ is similar to $\mathcal{C}$ but with i increased by one. If only one robot moves, the obtained configuration $(0^{\frac{k}{2}-i-1}, 1, 0^{i+1}, 1^j, 0^i, 1, 0^{\frac{k}{2}-i}, 1^{n-k-j-2})$ is asymmetric and not adjacent to any other symmetric configuration, and hence $\mathcal{C}'$ can be easily obtained with the subsequent move. Once configuration $\mathcal{C}^c$ with $j = 1$ is reached, again COMPACT$_0$ is applied. If both the permitted robots move, a symmetric configuration $\mathcal{C}'' = (0^{\frac{k}{2}-1}, 1, 0, 1, 0^{\frac{k}{2}-1}, 1^{n-k-1})$ with a multiplicity in $v_{\frac{k}{2}}$ is reached. This equals to the case of symmetric configurations with k odd that will be discussed later. If only one robot moves, configuration $(0^{\frac{k}{2}-1}, 1, 0^{\frac{k}{2}+1}, 1^{n-k-1})$ is reached. As k is even, then $4 < k < n-4$ and hence, such a configuration is asymmetric and not adjacent to any other symmetric configuration, and hence $\mathcal{C}''$ can be easily obtained with the subsequent move.

If k is even and n is odd, then ALIGN either reaches a configuration of type $\mathcal{C}^b$ or of type $\mathcal{C}^c$ with either j or $n-k-j$ odd. In this case, Algorithm GATHERING behaves as above but creating the multiplicity at the central node of the only odd sequence of consecutive empty nodes among j and $n-k-j$. Eventually, algorithm GATHERING achieves configuration $\mathcal{C}''$. Again, this equals to the case of symmetric configurations with k odd. Note that, this case is similar to the technique presented in [**?**] where the solved configurations are only those with k even and n odd.

If the initial configuration has k odd, then ALIGN always reaches a configuration of type $\mathcal{C}^b$. In this case, the used technique is similar to that presented in [**?**] where the solved configurations are only those with k odd. From $\mathcal{C}^b$, Algorithm GATHERING permits robots at $v_{\frac{k-1}{2}-1}$ and $v_{\frac{k-1}{2}+1}$ to move towards $v_{\frac{k-1}{2}}$. If only one robot performs the move, configuration $(0^{\frac{k'}{2}-1}, 1, 0^{\frac{k'}{2}+1}, 1^{n-k'-1})$ is achieved with $k' = k-1$. By the parity of k', configuration $\mathcal{C}''$ is achieved by the subsequent move. If both robots perform the move synchronously, again configuration $\mathcal{C}''$ is reached. From here, Algorithm GATHERING performs move COMPACT$_1$ defined as follows. Let $\mathcal{C} = (v_0, v_1, \ldots, v_n)$ be a configuration of type $(0^{\frac{k-i}{2}}, 1, 0^i, 1, 0^{\frac{k-i}{2}}, 1^{n-k-2})$ where $1 \leq i \leq k$, then COMPACT$_1$ consists in moving the robot at $v_{\frac{k-i}{2}-1}$ towards $v_{\frac{k-i}{2}}$. As $\mathcal{C}$ is symmetric, move COMPACT$_1$ permits two robots to move. If both move synchronously, the outcoming configuration $\mathcal{C}'$ is similar to $\mathcal{C}$ but with i increased by two. If only one robot moves, as before, the obtained configuration is asymmetric and not adjacent to any other symmetric configuration, and hence $\mathcal{C}'$ can be easily obtained with the subsequent move. By iterating this process, Algorithm GATHERING achieves a configuration of type $\mathcal{C}^b$ with the number of occupied nodes decreased by two with respect to the original configuration. Eventually, this process terminates with only one occupied node.



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399