Theoretical Computer Science 411 (2010) 3235–3246

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring☆

Ralf Klasing [a], Adrian Kosowski [b,a], Alfredo Navarra [c,*]

[a] LaBRI - Université Bordeaux 1 - CNRS, 351 cours de la Libération, 33405 Talence cedex, France

[b] Department of Algorithms and System Modeling, Gdańsk University of Technology, Narutowicza 11/12, 80952 Gdańsk, Poland

[c] Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Perugia, Italy

ARTICLE INFO

Article history:
Received 29 June 2009
Received in revised form 29 April 2010
Accepted 18 May 2010
Communicated by D. Peleg

Keywords:
Asynchronous system
Mobile robots
Oblivious robots
Gathering problem
Ring

ABSTRACT

One of the recently considered models of robot-based computing makes use of identical, memoryless mobile units placed in nodes of an anonymous graph. The robots operate in Look–Compute–Move cycles; in one cycle, a robot takes a snapshot of the current configuration (Look), takes a decision whether to stay idle or to move to one of the nodes adjacent to its current position (Compute), and in the latter case makes an instantaneous move to this neighbor (Move). Cycles are performed asynchronously for each robot.

In such a restricted scenario, we study the influence of symmetries of the robot configuration on the feasibility of certain computational tasks. More precisely, we deal with the problem of gathering all robots at one node of the graph, and propose a solution based on a symmetry-preserving strategy. When the considered graph is an undirected ring and the number of robots is sufficiently large (more than 18), such an approach is proved to solve the problem for all starting situations, as long as gathering is feasible. In this way we also close the open problem of characterizing symmetric situations on the ring which admit a gathering [R. Klasing, E. Markou, A. Pelc: Gathering asynchronous oblivious mobile robots in a ring, Theoret. Comput. Sci. 390 (1) (2008) 27–39].

The proposed symmetry-preserving approach, which is complementary to symmetry-breaking techniques found in related work, appears to be new and may have further applications in robot-based computing.

1. Introduction

The difficulty of many computational problems involving mobile entities (robots) is aggravated when robots cannot communicate directly, but have to take decisions about their moves only by observing the environment. One of the most restrictive scenarios considered in literature is the asynchronous Look–Compute–Move model for memoryless units which has been studied both for robots on the plane (the *continuous model* [13,20]) and for robots located on the nodes of a graph (the *discrete model* [10,11,16,17]). Herein we focus on computations in the discrete model which is described in more detail below.

☆ The research was partially funded by the European projects COST Action 293, ''Graphs and Algorithms in Communication Networks'' (GRAAL) and, COST Action 295, ''Dynamic Communication Networks'' (DYNAMO). A preliminary version of this paper appeared as [16].

* Corresponding author. Tel.: +39 0755855008.
E-mail addresses: klasing@labri.fr (R. Klasing), adrian@kaims.pl, kosowski@labri.fr (A. Kosowski), navarra@dmi.unipg.it, navarra@dipmat.unipg.it (A. Navarra).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.05.020

1.1. The discrete model

Consider an anonymous graph in which neither nodes nor links have any labels. Initially, some of the nodes of the graph are occupied by robots and there is at most one robot in each node. Robots operate in Look–Compute–Move cycles. 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 makes an instantaneous move to this neighbor (Move). Cycles are performed asynchronously for each robot. This means that the time between Look, Compute, and Move operations is finite but unbounded, and is decided by the adversary for each robot. The only constraint is that moves are instantaneous, and hence any robot performing a Look operation sees all other robots at nodes of the ring and not on edges. However, a robot r may perform a Look operation at some time t, perceiving robots at some nodes, then Compute a target neighbor at some time $t' > t$, and Move to this neighbor at some later time $t'' > t'$, at which some robots are in different nodes from those previously perceived by r because in the meantime they performed their Move operations. Hence, robots may move based on significantly outdated perceptions. It should be stressed that robots are memoryless (oblivious), i.e., they do not have any memory of past observations. Thus, the target node (which is either the current position of the robot or one of its neighbors) is decided by the robot during a Compute operation solely on the basis of the location of other robots perceived in the previous Look operation. Robots are anonymous and execute the same deterministic algorithm. They cannot leave any marks at visited nodes, nor send any messages to other robots.

We remark that the Look operation provides the robots with the entire graph configuration. Moreover, it is assumed that the robots have the ability to perceive, during the Look operation, if there is one or more robots located at the given node of the graph. This capability of robots is important and it has been well studied in the literature on robot gathering under the name of *multiplicity detection* [13,20]. In fact, without this capability, many computational problems (such as the gathering problem considered herein) are impossible to solve for all non-trivial starting configurations. It should be stressed that, during a Look operation, a robot can only tell if at some node there are no robots, there is one robot, or there is more than one robot: a robot does not see the difference between a node occupied by a or b robots, for distinct $a, b > 1$.

Problems studied so far in the discrete model include gathering on the ring [17], exploration of the ring [10], and tree exploration [11].

1.2. Our results

In this paper, we consider one of the most fundamental problems of self-organization of mobile entities, known in the literature as the *gathering* problem. Robots, initially situated at different locations, have to gather at the same location (not determined in advance) and remain in it. Our considerations focus on gathering robots in the discrete model for the undirected ring; such a scenario poses a number of problems due to the high number of potential symmetries of the robot configuration. This problem was initially studied in [17], where certain configurations were shown to be gatherable by means of symmetry-breaking techniques, but the question of the general-case solution was posed as an open problem. In particular, for an odd number of robots it has been proved that gathering is feasible if and only if the initial configuration is not periodic, and a gathering algorithm for any such configuration has been provided. For an even number of robots, the feasibility of gathering has been solved except for one type of symmetric initial configurations.

Herein we provide procedures for gathering all configurations on the ring with more than 18 robots for which gathering is feasible, and give a full characterization of all such configurations (Theorem 5.2). In fact, we provide a new technique for dealing with symmetric configurations: our approach is based on preserving symmetry rather than breaking it.

1.3. Related work

The problem of gathering mobile robots in one location has been extensively studied in the literature. Many variations of this task have been considered in different computational models, empowering the robots with different capabilities, e.g. memory, markers. Robots move either in a graph, see e.g. [2,8,9,12,18], or in the plane [1,3–7,13,19–21]; they are labeled [8,9,18], or anonymous [1,3–7,13,19–21]; gathering algorithms are probabilistic (see [2] and the literature cited there), or deterministic [1,3–8,12,13,18–21]. Deterministic algorithms for gathering robots in a ring (which is a task closest to our current setting) have been studied for example in [8,9,12,14,18]. In [8,9,18], the symmetry was broken by assuming that robots have distinct labels, and in [12] it was broken by using tokens. The very weak assumption of anonymous identical robots was studied in [1,3–7,13,19–21], where robots could move freely in the plane. The scenario was further refined in various ways. In [4,14], it was assumed that robots have memory, while in [1,3,5–7,13,19–21], the robots were oblivious, i.e., it was assumed that they do not have any memory of past observations. Oblivious robots operate in Look–Compute–Move cycles, similar to those described in our scenario. The differences are in the amount of synchrony assumed in the execution of the cycles. In [3,21], cycles were executed synchronously in rounds by all active robots, and the adversary could only decide which robots are active in a given cycle. In [4–7,13,19,20], they were executed asynchronously: the adversary could interleave operations arbitrarily, stop robots during the move, and schedule Look operations of some robots while others were moving. It was proved in [13] that gathering is possible in the asynchronous model if robots have the same orientation of the plane, even with limited visibility. Without orientation, the gathering problem was positively solved in [5], assuming that the robots have the capability of multiplicity detection. A complementary negative result concerning the asynchronous model was proved in [20]: without multiplicity detection, gathering robots that do not have orientation is impossible.

2. Terminology and preliminaries

We consider an n-node anonymous ring without orientation. Initially, some nodes of the ring are occupied by robots and there is at most one robot in each node.

During a Look operation, a robot perceives the relative locations on the ring of multiplicities and single robots. We recall that a multiplicity occurs when more than one robot occupies the same location. For the purpose of the definition only, let us call one of the directions on the ring *clockwise*, and the other *anti-clockwise*. Then, for a fixed robot r, let $S_C(r)$ denote the ordered sequence of distances from r to all single robots when traversing the ring in the clockwise direction, and let $S_A(r)$ be the ordered sequence of such distances when moving anti-clockwise. Sets $M_C(r)$ and $M_A(r)$ are likewise defined for distances from r to all multiplicities. Then the *view* $V(r)$ provided to robot r is defined as the set of ordered pairs $V(r) = \{(S_C(r), M_C(r)), (S_A(r), M_A(r))\}$. If there are no multiplicities, we will drop the second sequence in each case and write the view simply as the set of two sequences $V(r) = \{S_C(r), S_A(r)\}$.

The current configuration C of the system can be described in terms of the view of a robot r which is performing the Look operation at the current moment, but disregarding the location of robot r; formally, $C = \{\{(S_C(r) \oplus i, M_C(r) \oplus i), (S_A(r) \ominus i, M_A(r) \ominus i)\} : i \in [1, n]\}$, where operations $\oplus$ and $\ominus$ denote modulo n addition and subtraction, respectively. Note that the configuration is independent of the choice of robot r and of the choice of the clockwise direction.

A configuration C is called *periodic* if it is invariable under rotation, i.e. $C = C \oplus k$ for some integer $k \in [1, n-1]$. A configuration C is called *symmetric* if the ring has a geometrical *axis of symmetry*, which reflects single robots into single robots, multiplicities into multiplicities, and empty nodes into empty nodes. Note that a symmetric configuration is not periodic if and only if it has exactly one axis of symmetry [17]. A symmetric configuration C with an axis of symmetry s has an *edge–edge symmetry* if s goes through (the middles of) two antipodal edges; it has a *node-on-axis symmetry* if at least one node is on the axis of symmetry.

A *pole* is an intersection point of a line with the ring (this may either be a node or in between two nodes). For configurations with a single axis of symmetry, nodes on the axis of symmetry are natural gathering points. The pole of the axis of symmetry used by the considered algorithm for gathering is known as the *North pole*; the other pole is called the *South pole*.

The set of nodes of the ring forming a path between two robots, excluding end-points, is called an *arc*. Two robots are called *neighbors* if at least one of the two arcs of the ring between them does not contain any robots. When uniquely defined, the arc of the ring between two neighboring robots u, v with no robots on it is called the *gap* $u - v$. The length of gap $u - v$ is denoted as $|u - v|$; obviously $|u - v| = |v - u|$. Two robots forming a multiplicity are assumed to form a gap of length 0. A gap of minimum length in a given configuration is simply called *minimal*.

The notation for gaps is extended to allow for *chains*, $u_1 - u_2 - \cdots - u_k$, i.e. sequences of robots separated by gaps. If some robots $u_i - \cdots - u_j$ form a multiplicity M, then the considered chain may be written compactly as $u_1 - \cdots - u_{i-1} - M - u_{j+1} - \cdots - u_k$.

We now introduce the concept of *extrapolated length* $|u \to v|$ of a gap $u - v$, useful for breaking ties in the gathering process. Let $u - v - v_1 - v_2 - \cdots - v_s$ be the longest possible chain such that, for all i, $v_i \neq u$ and v_i does not belong to a multiplicity. Then $|u \to v| = (|u - v|, |u - v_1|, |u - v_2|, \ldots, |u - v_s|)$. Values of extrapolated gap lengths are compared lexicographically.

A key operation used in the gathering process is known as the *contraction* of a gap. Let $u - v$ be an arbitrary gap belonging to some chain $t - u - v - w$, such that $|u \to t| > |v \to w|$. Then the *contraction of* $u - v$ is the operation of moving robot u a single node towards robot v.

Note that if a configuration C' was formed from a configuration C by contraction of some gap $u - v$ (by moving u) in a chain $t - u - v - w$, then it is clear that in C' we have $|t - u| > |v - w|$. The corresponding *decontraction of* $u - v$ in C' is uniquely defined as the operation of moving robot u a single node away from robot v unless some other symmetry has been determined.

3. Gathering procedure for symmetric configurations

This section contains a description for a gathering procedure, starting from an important class of initially symmetric configurations. This is then combined with approaches for other types of configurations in the next section.

Theorem 3.1. *There exists a procedure for solving the gathering problem starting from all initial configurations of more than* 18 *robots having exactly one axis of symmetry, provided that the axis is not of the edge–edge type and its poles do not contain any robots.*

The proof of the theorem is constructive, and we will confine ourselves to providing an algorithm which describes the Compute part of the cycle of robots' activities. In order to simplify notation, the performed actions will often be expressed using configurations (identical for all robots sharing the same snapshot of the system), and not locally centered views. For example, if we require only robots specifying certain geometrical criteria to move, then each robot will be able to recognize whether to perform the specified action or not.

Since the configuration has exactly one axis of symmetry, it is possible to define a partial order on the robots in which only pairs of robots, symmetric with respect to the axis of symmetry, are incomparable. Our algorithm allows at any given time exactly two robots u and $\bar{u}$, symmetric with respect to the axis of symmetry, to make corresponding moves, hence preserving symmetry. Observe that there are two possible move scenarios leading to different types of new configurations.

- *Both robots, u and $\bar{u}$, make their moves simultaneously.* In this case, in the new configuration the axis of symmetry remains unchanged. For the correctness of the algorithm, it is essential to ensure that no new axes of symmetry are formed (since otherwise the new configuration cannot be gathered).
- *One of the robots, say u, performs its move before the other robot $\bar{u}$.* All other robots must be able to recognize that the current configuration is one move away from a symmetry, and robot $\bar{u}$ is now the only one allowed to perform a move. Observe that it is then irrelevant whether robot $\bar{u}$ performed its Look operation before or after robot u was moved; the outcome of its move is exactly the same.

The algorithm proposed herein detects configurations which have exactly one axis of symmetry of the node-on-axis type with no robots on the poles (which we call *A-type configurations*) and those which are exactly one step away from such a configuration (*B-type configurations*). The remaining types of gatherable configurations are not considered by the algorithm; our approach is extended to include them in the next section.

Our algorithm runs in four main phases; these are informally outlined in the next subsection, and formalized in Section 3.2.

3.1. Illustration of approach

Let us suppose that the system starts in an A-type configuration. (Note that in view of impossibility results from [17] (see Theorem 5.1), symmetric configurations which are not A-type configurations are never gatherable.)

The four phases of our algorithm can be outlined as follows. In the first phase of the algorithm, we lead the system to an A-type configuration with exactly two (symmetrical) multiplicities. In the second phase, all of the other robots (with the exception of two symmetrically located robots called *guards*) are gathered into the multiplicities. In the third phase, the multiplicities are moved to their final gathering point on the axis of symmetry, away from the guards (remember that there is a node-on-axis symmetry in our case). Finally, in the fourth phase, the guards join the single remaining multiplicity in the gathering point.

The current phase of the algorithm can be determined by only looking at the state of the system; this will be discussed in more detail later. The single axis of symmetry is maintained throughout the process, allowing configurations that are at most at one step from a symmetric configuration with respect to the considered axis of symmetry. In the first phase, the locations of all minimal gaps are used for this purpose. In the second phase, the axis is determined by positions of the multiplicities, while in the third phase the axis is determined by the positions of the guards. Finally, in the fourth phase, the gathering point with the only multiplicity is known.

Referring to Figs. 1 and 2, we now describe in more details the basic intuitions of our algorithm. In both figures, configurations a describe two possible initial states of the system (A-type configurations). In the first phase, the objective is to create two symmetric multiplicities such that both arcs of the circle between them contain at least two robots, neither of which is at a distance of one from a multiplicity. The normal move (Fig. 1(a)) consists in the contraction of two symmetrical minimal gaps. The pair of minimal gaps is selected in such a way that the contraction does not create two multiplicities which violate the imposed constraint on robots on the arcs between them; if there exists a minimal gap crossing the axis of symmetry, this is not chosen either. It may happen that no minimal gap appropriate for contraction exists (Fig. 2(a)). In that case, we select for contraction the pair of (not necessarily minimal) gaps which are central in terms of the number of robots separating them from the poles of the axis of symmetry (gaps between robots 5–5 in Fig. 2(a)); if there are two pairs of candidate gaps, a tie-breaking mechanism is applied.

The performed contractions result in a new symmetric configuration (configurations c in both figures), possibly preceded by a state of violated symmetry in a B-type configuration (configurations b). The process of selecting the gap for contraction allows the robots to recreate configuration a knowing configuration b only, and to proceed from there to configuration c. Configuration c is in fact an A-type configuration just as configuration a, and the procedure repeats until the two sought multiplicities are created (configurations d). At this point, the first phase of the algorithm is complete. Note that the contraction rules applied in Fig. 2 require a sufficiently large number of robots (more than 18; see Lemma 3.2) to guarantee correctness.

The next phases of the algorithm are shown in Fig. 1 only. In Phase 2 it is necessary to decide on one of the two poles of the axis of symmetry as the gathering point (the North pole). The poles are chosen so that the northern arc between multiplicities has more robots than the southern arc; in the case of a tie, the side on which the nearest robots are closer to the multiplicities is the northern one. The robots are moved in symmetrical pairs towards their respective multiplicities, starting from the robots on the northern arc (Fig. 1(e), (f)). Note that the definition of the North and South is consistently preserved throughout the process. Phase 2 ends when nearly all the robots have been merged into the multiplicities, and the remaining robots occupy not more than six nodes in an arrangement matching a specific pattern (Fig. 1(f)). Two robots,

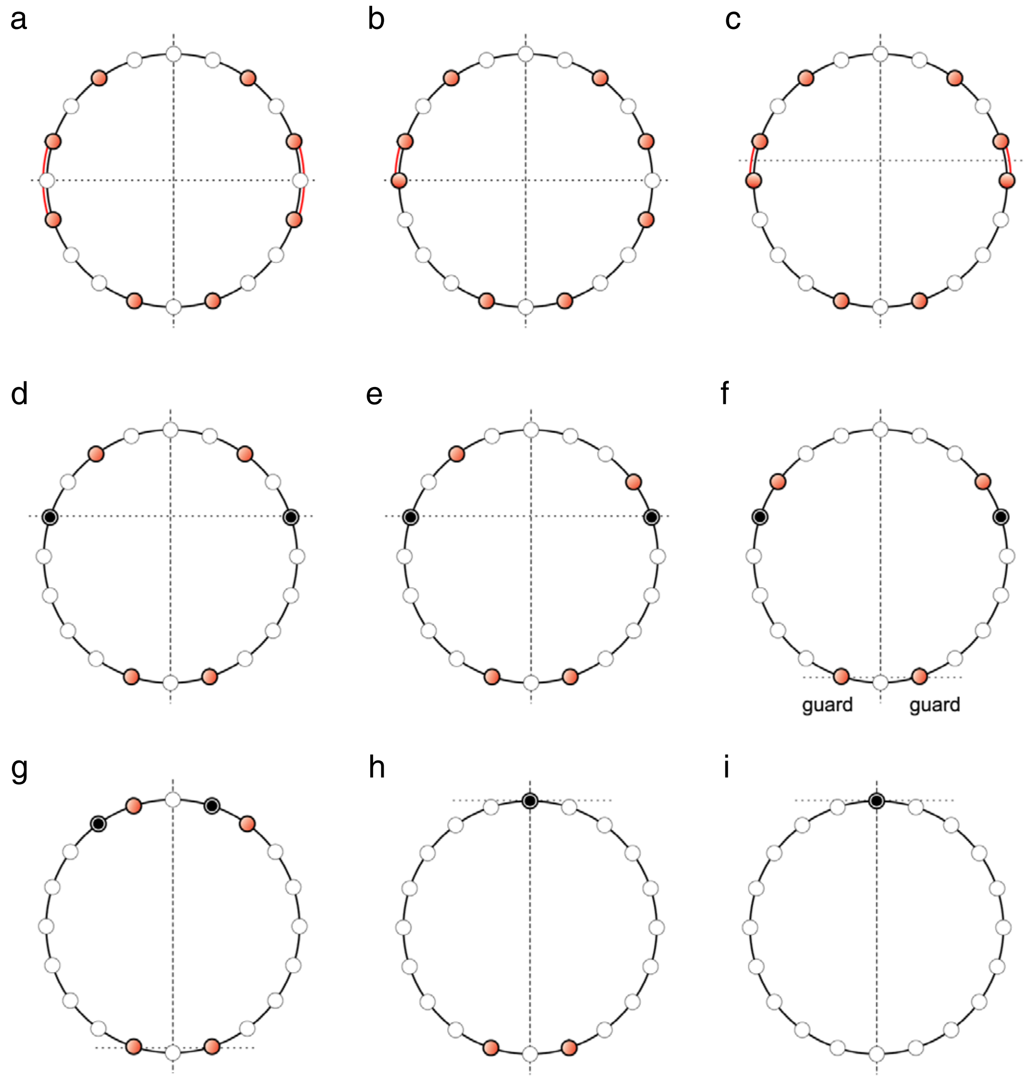

Fig. 1. An example of a scenario for a symmetric configuration. White nodes represent empty nodes, shaded nodes are nodes occupied by a single robot, and black nodes are nodes occupied by at least two robots, i.e., multiplicities. The North pole is at the top of the axis of symmetry. The dashed horizontal line can be understood as a helper line for recognizing the axis of symmetry.

separated by gaps from the multiplicities, always remain on the southern arc and act as the guards of the axis of symmetry throughout Phase 3. The multiplicities are moved step by step towards the North pole; note that not all the robots in a multiplicity have to move simultaneously (Fig. 1(g)). When the pattern shown in Fig. 1(h) is achieved, Phase 4 starts, and the two remaining robots are moved step by step until they reach the North pole (Fig. 1(i)), and the algorithm is complete.

3.2. *Formalization of approach*

The distinction among the four phases of the proposed algorithm is in fact possible knowing only the current configuration C. To do this, we now introduce some further notation.

A configuration can also be represented in the form of a string of characters as follows: starting from an arbitrary node and moving around the ring in a chosen direction, for each node we append a character representing whether the node is empty, contains a single robot, or a multiplicity. We say that configuration C matches a *chain pattern* $[P]$, $C \in [P]$, if there exists a string representation of C belonging to the language described by the tokens in $[P]$. For some integer values a and b, token $\sigma_{a:b}$ is understood as between a and b occurrences of single robots (possibly separated by any number of empty nodes) followed by at least one empty node. Token $\mu_{a:b}$ is understood as between a and b occurrences of consecutive non-empty nodes, at least one of which is a multiplicity, followed by at least one empty node. Ranges of the form $a : a$ are simply written as a. For example, in Fig. 1, the pattern $[\mu_{1:2}, \sigma_{1:2}, \mu_2]$ is matched by configurations f and g.

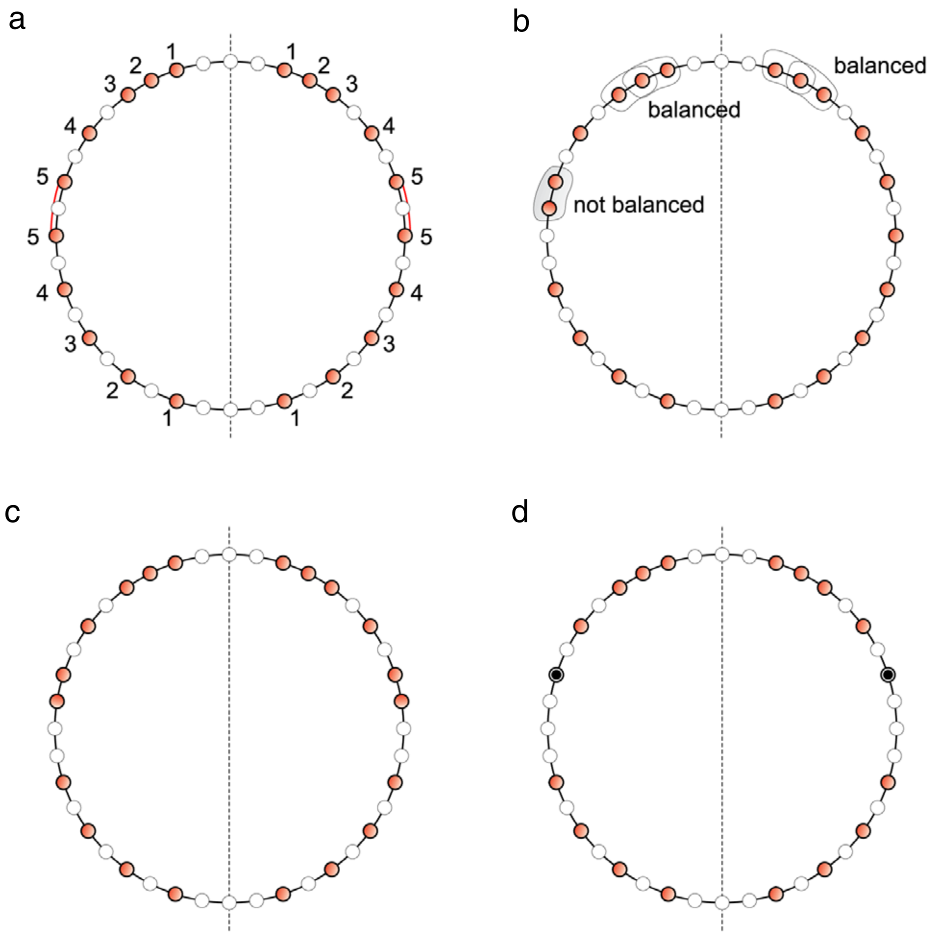

Fig. 2. An example of a scenario for a symmetric configuration (contraction of equatorial gaps). Details of the construction are given in Section 3.3.

Table 1
Division into phases, assuming no robots on the axis of symmetry in the initial state: $m(C)$ — number of multiplicities in configuration C, $p(C)$ — total number of different nodes occupied by robots in C.

Phase	Multiplicities	Occupied nodes	Additional constraints
1	$m(C) < 2$	$p(C) > 6$	None
2	$m(C) = 2$	$p(C) \geq 6$	If $p(C) = 6$, then $C \notin [\mu_{1:2}, \sigma_2, \mu_{1:2}]$
3	$m(C) = 2$	$4 \leq p(C) \leq 6$	If $p(C) = 6$, then $C \in [\mu_{1:2}, \sigma_2, \mu_{1:2}]$
4	$m(C) \geq 1$	$p(C) \leq 3$	None

Herein we restrict ourselves to a presentation of the algorithm for the case of an initial configuration with exactly one axis of symmetry, having a node-on-axis-type symmetry, without any robots on the axis.

The proposed algorithm performs the gathering in four basic phases, as defined in Table 1.

When performing its Compute step, each robot can clearly determine which phase of the algorithm it is currently running (cases not covered in the table cannot appear in the initial state and do not occur later due to the construction of the algorithm). The algorithm is defined so as to guarantee that, when two robots are allowed to move simultaneously, their views correspond to the same phase of the algorithm. Bearing this in mind, we can now consider the four phases separately in the following subsections.

3.3. *Phase 1: obtaining two non-adjacent multiplicities*

The algorithm is defined by the following elements.

- A subroutine defining a move for an A-type configuration which leads to a new A-type configuration, assuming that both the robots which are chosen to move perform their action simultaneously.

- A subroutine for detecting the preceding A-type configuration when the current state of a system is a B-type configuration.

The procedure for A-type configurations is presented as Algorithm 1. A gap $u - v$ is called *equatorial* with respect to a line s if the number of robots on the arc from u to one pole of s and from v to the other pole of s differs by at most 1 (a multiplicity is counted as two robots).

Algorithm 1 Procedure for A-type configurations (Phase 1)

(i) Choose a pair of minimal gaps $u - v$ and $\bar{u} - \bar{v}$ such that the following conditions are fulfilled:
  - $u - v$ does not intersect the axis of symmetry ($u \neq \bar{v}$),
  - the contraction of $u - v$ and $\bar{u} - \bar{v}$ does not create two multiplicities with no other robots in between them,
  - the contraction of $u - v$ and $\bar{u} - \bar{v}$ does not create two multiplicities with exactly two robots in between, adjacent to these multiplicities,

  then perform the contractions of $u - v$ and $\bar{u} - \bar{v}$.

(ii) If no such pair exists (see Fig. 3 for a visualization), perform the contraction of chosen gaps $u - v$ and $\bar{u} - \bar{v}$ which are equatorial with respect to the axis of symmetry. If there are two pairs of equatorial gaps of different lengths, the shorter pair is always chosen for contraction.

For completeness of the procedure, it is necessary to provide some mechanism of choosing one of several possible candidate gaps. Such ties are easily broken, since for a given configuration it is possible to define a partial order on the set of robots in which only symmetrical robots are not comparable [17].

The definition of the procedure always allows a move of exactly two symmetrical robots. We first show that the above set of rules is sufficient to gather an A-type configuration, provided that both symmetrical robots always perform their Look operations as well as Move operations simultaneously. A scheduler that ensures this property is called a *symmetry-preserving scheduler*.

3.3.1. Case of a symmetry-preserving scheduler

Before proceeding with the proofs, we recall the obvious geometrical fact that, if for a configuration on the ring it is in some way possible to distinguish (select) exactly two arcs, then the configuration can only have zero, one, or two perpendicular axes of symmetry.

Lemma 3.1. *Under a symmetry-preserving scheduler, the new configuration after performing rule* (i) *of Algorithm* 1 *is also an A-type configuration.*

Proof. Indeed, consider the contraction of minimal gap $u - v$ in a chain $t - u - v - w$ and its complement $\bar{u} - \bar{v}$ in chain $\bar{t} - \bar{u} - \bar{v} - \bar{w}$. The obtained configuration has exactly two minimal gaps, $u - v$ and $\bar{u} - \bar{v}$. Thus, after the move the axis of symmetry remains unchanged and no new axes are created. In fact, the new configuration is the same as before but with the minimal gaps $u - v$ and its complement $\bar{u} - \bar{v}$ contracted. This move clearly maintains the axis of symmetry of the original configuration. Moreover, $u - v$ and $\bar{u} - \bar{v}$ are the minimal gaps in the new configuration and no further gaps can have their length. Hence, if a new axis of symmetry is created after the move, it must necessarily cut the configuration between $u - v$ and $\bar{u} - \bar{v}$. However, this is not possible, as by the properties of a contraction $|t - u| \neq |v - w|$. □

For a given configuration C, we will call a gap $u - v$ *balanced* if for the chain $s - t - u - v - w - x$ we have $|t - u| = |v - w|$ or $|u - v| \in \{|s - t|, |t - u|, |v - w|, |w - x|\}$.

Lemma 3.2. *If for a given A-type configuration rule* (i) *of Algorithm* 1 *cannot be applied, the following claims hold.*

(a) *All the minimal gaps have their end-points within a set of* 12 *robots — six robots surrounding each pole of s (the three nearest robots on one side and the three on the other side).*
(b) *The set of minimal gaps consists of between* 1 *and* 10 *gaps.*
(c) *All the minimal gaps are balanced.*
(d) *If $u - v$ is an equatorial gap, then none of the gaps in the chain $s - t - u - v - w - x$ is minimal.*
(e) *There exist either one or two symmetrical chains, maximal in terms of inclusion, which do not contain any minimal gaps and which consist of more than four robots each.*

Proof. If some minimal gap $u - v$ cannot be contracted according to rule (i) of the Algorithm, then the location of this gap with respect to some pole of the axis must be one of those shown in Fig. 3(a)–(c). (Note that, if $u - v$ does not intersect the axis of symmetry, then its length is necessarily equal to 1.) Since by assumption rule (i) cannot be applied to any minimal gap, we immediately obtain Claim (a), and also Claim (b) as a direct consequence. To show Claim (c), consider any minimal gap $u - v$ and the surrounding chain $s - t - u - v - w - x$. For the configuration from Fig. 3(a) we have $|t - u| = |v - w|$, for the configuration from Fig. 3(b), $|u - v| \in \{|s - t|, |w - x|\}$, and finally, for the configuration from Fig. 3(c), $|u - v| \in \{|t - u|, |v - w|\}$. In any of these cases, the gap is balanced. Taking into account the assumption that there are more than 18 robots on the ring, Claim (d) results directly from Claim (a), since the equatorial gaps are then sufficiently far from the poles (in terms of the number of separating robots). Likewise, we obtain Claim (e) by considering the maximal chains which contain an equatorial gap and do not contain any minimal gaps. □

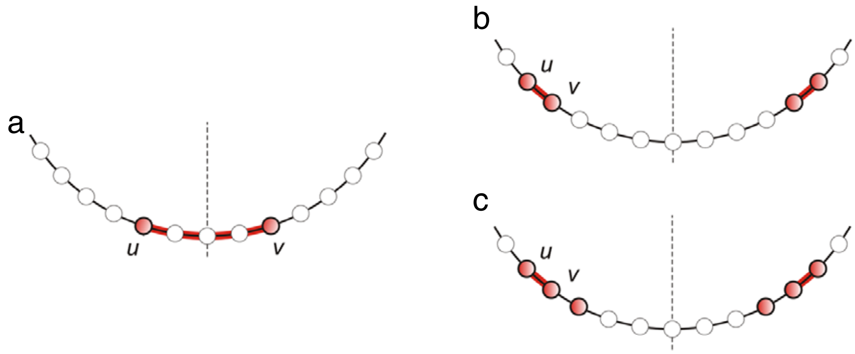

Fig. 3. Configurations of minimal gaps for which rule (i) of Algorithm 1 cannot be applied.

Lemma 3.3. *Under a symmetry-preserving scheduler, the new configuration after performing rule* (ii) *is also an A-type configuration.*

Proof. We need to consider two cases.

(1) If the contraction of $u - v$ and $\bar{u} - \bar{v}$ creates no new minimal gaps, then the set of minimal gaps remains unchanged. The old axis of symmetry is still an axis of symmetry of the new configuration. Now, taking into account Lemma 3.2(e) and the possible symmetries of the maximal chains characterized by it, there could exist at most one more candidate for an axis of symmetry for the new configuration, perpendicular to the original axis. Since the number of robots on both sides of an axis of symmetry is the same, either the new axis crosses the newly contracted gaps $u - v$ and $\bar{u} - \bar{v}$ or it has robots u and $\bar{u}$ on its poles. In the first case, the axis cannot be an axis of symmetry, since for the chain $t - u - v - w$ we have $|t - u| \neq |v - w|$. In the second case, a contradiction arises as well, since the shorter equatorial gaps have been contracted and hence these gaps cannot be reflected by the new axis into the unchanged (longer) equatorial gaps.

(2) If the contraction of $u - v$ and $\bar{u} - \bar{v}$ creates two new minimal gaps, then these gaps are non-balanced by Lemma 3.2(d); moreover, these are now the only two non-balanced minimal gaps on the ring by Lemma 3.2(c). By a similar argument as before, these two non-balanced minimal gaps have to be reflected by any axis of symmetry into each other, so the axis of symmetry is unique. □

Finally, we make a note on the convergence of the performed process.

Lemma 3.4. *Under a symmetry-preserving scheduler, Phase* 1 *is completed after a finite number of steps.*

Proof. If rule (i) is performed then the length of the minimal gap decreases in each step. Otherwise, the length of the shorter equatorial gap decreases, while the length of the minimal gap remains unchanged (since all minimal gaps are then concentrated around the poles). The process obviously converges to a minimal gap length of 0; hence we obtain two multiplicities and, by Table 1, Phase 1 is complete. □

3.3.2. Extension to the general scheduler

Depending on the rule used in the preceding A-type configuration and the outcome of the move, we obtain a B-type configuration which belongs to one of the following subtypes.

B1: The current configuration was obtained by contracting a minimal gap in an A-type configuration using rule (i).
B2: The current configuration was obtained by contracting an equatorial gap in an A-type configuration using rule (ii), but without creating any new minimal gaps in the process.
B3: The current configuration was obtained by contracting an equatorial gap in an A-type configuration using rule (ii), but creating one new minimal gap in the process.

Before proceeding any further, for a configuration we define a *compass axis* as any line *s* fulfilling the following constraints:

- *s* is an axis of symmetry of the set of balanced minimal gaps,
- the number of robots on both sides of *s* is equal,
- all the balanced minimal gaps have their end-points within a set of 12 robots — six robots surrounding each pole of *s* (the three nearest robots on one side and the three on the other side).

We are now ready to prove the following theorem.

Lemma 3.5. *The sets of* A-, B1-, B2-, *and* B3-*type configurations are all pairwise disjoint.*

Proof. A B1-type configuration has exactly one non-balanced minimal gap. In consequence, such a configuration obviously cannot have an axis of symmetry.

Table 2
Telling apart different types of configurations: $q(C)$ — total number of minimal gaps in C, $q_b(C)$ — total number of balanced minimal gaps in C, $s(C)$ — number of axes of symmetry.

Type	Minimal gaps	Balanced minimal gaps	Axes of symmetry
A	Irrelevant	Irrelevant	$s(C) = 1$
B1	$q(C) = 1$	$q_b(C) = 0$	$s(C) = 0$
B2	$1 \leq q(C) \leq 10$	$q_b(C) = q(C)$	$s(C) = 0$
B3	$2 \leq q(C) \leq 11$	$q_b(C) = q(C) - 1$	$s(C) = 0$

A B2-type configuration has the same set of minimal gaps as the original A-type configuration; hence we can make use of Lemma 3.2 also for this configuration. In consequence, a B2-type configuration has between 1 and 10 minimal gaps, all of which are balanced, and has exactly one compass axis identical to the axis of symmetry of the original A-type configuration. Since the compass axis of a configuration is the only possible candidate for its axis of symmetry, and a B2-type configuration is exactly one move apart from an A-type configuration having this axis as an axis of symmetry, a B2-type configuration has no axes of symmetry.

A B3-type configuration has the same set of balanced minimal gaps as the original A-type configuration, and additionally one more minimal gap is obtained as a result of the contraction (thus between 2 and 11 minimal gaps in total); however, this gap is non-balanced, and as such does not affect the existence of the compass axis. Analogously to the previous case, we obtain that a B3-type configuration has exactly one compass axis and no axes of symmetry.

Putting together the above properties (see Table 2), we directly obtain the claim. □

Lemma 3.6. *Under a general scheduler, Phase* 1 *is completed after a finite number of steps.*

Proof. Taking into account Lemmas 3.4 and 3.5, we now only need to define a procedure to determine for a B-type configuration a unique preceding A-type configuration. The next move is then defined by imitating the behavior of a symmetry-preserving scheduler.

Taking into account the above observations (see Table 2), we obtain that for a given configuration C we can determine if it is an A-type configuration, or a candidate for a B1, B2, or B3-type configuration. In the latter cases, there exists exactly one possibility of recreating the potentially preceding A-type configuration. For a B1-type configuration, it is necessary to decontract the unique minimal gap. For a B2- or B3-type configuration, the shortest of the gaps equatorial with respect to the compass axis should be decontracted. □

3.4. Phase 2: partial gathering with two multiplicities

The first phase ends when two symmetrical multiplicities are created. Throughout the second phase of the algorithm, the two existing multiplicities M_1 and M_2 make no moves. Multiplicities M_1 and M_2 divide the ring into two parts, which we will call *northern* (around the North pole) and *southern* (around the South pole). Each of these parts initially contains at least two robots not directly adjacent to a multiplicity. Throughout the process North and South are defined in such a way as to fulfill the following conditions:

- the number of nodes in the northern part is odd,
- if both parts have an odd number of nodes, the southern part always contains not less than one robot, and not fewer robots than the northern part,
- if both parts have an odd number of nodes and contain the same number of robots, consider the chain $r_N - M_1 - r_S$ with robot r_N in the northern part and robot r_S in the southern part; then $|M_1 \to r_S| > |M_1 \to r_N|$.

The gathering procedure, presented as Algorithm 2, is defined so as to move all but at most four of the single robots into the two existing multiplicities (without creating any new multiplicities).

Algorithm 2 Procedure for Phase 2

(i) If the northern part contains at least one robot, move a robot in the northern part, such that there are no robots between itself and one of the multiplicities, towards this multiplicity (in the case of choice of robots, select the one with a longer way left to go; if the distance is the same, both robots are allowed to move).
(ii) Otherwise, perform an analogous operation in the southern part but for the two symmetric nodes closest to the pole (these nodes will serve as guards in the next phase).

It is important to note that the adopted definition of North and South guarantees that the same labeling of the poles is maintained throughout the process.

In accordance with Table 1, the phase ends when all but at most four single robots have been merged with the multiplicities. The last pair of robots in the southern part have not yet made a move and they are separated by at least one empty field from a multiplicity; these robots will serve as guards in the last phases of the algorithm.

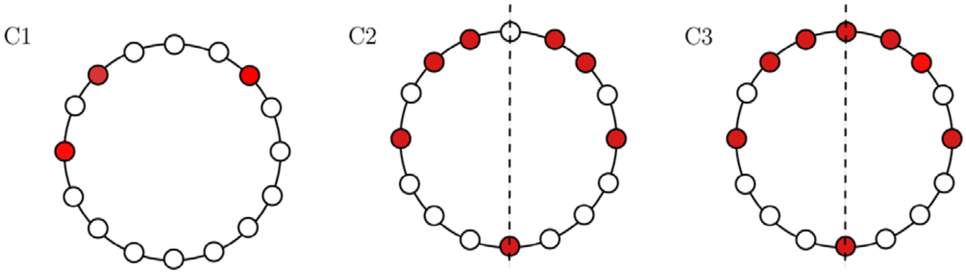

Fig. 4. Examples of possible C-type configurations.

3.5. Phase 3: gathering two multiplicities using guards

The third phase of the algorithm is performed when $C \in [\mu_{1:2}, \sigma_2, \mu_{1:2}]$. The two robots u and v corresponding to the token σ_2 define a unique axis of symmetry, orthogonal to the gap $u - v$. The remaining robots (and multiplicities) can move towards the North pole of this axis; for a given configuration, only those robots which have the longest way to go are allowed to move. In this way the configuration pattern is maintained throughout the process, until the moving robots converge on the pole and (optionally) the two directly adjacent nodes. The configuration pattern then changes to $C \in [\mu_{1:3}, \sigma_2]$, and can be likewise maintained until all robots except for the guards gather at the required pole in a single multiplicity ($C \in [\mu_1, \sigma_2]$).

3.6. Phase 4: withdrawing guards to the gathering point

In this phase, the unique multiplicity on the North pole determines the gathering point for the remaining guards. The guards can be moved towards the multiplicity following the rule that, if the guards are at a different distance from the multiplicity, the guard further away should move (in the case of a tie, both guards are allowed to move). The configuration is maintained in the pattern $C \in [\mu_1, \sigma_2]$. Only in at most two final moves do we have $C \in [\mu_{1:3}]$ or $C \in [\mu_{1:2}]$ (still with exactly one multiplicity). Eventually, $C \in [\mu_1]$ and the gathering is achieved.

This completes the proof of Theorem 3.1.

4. Extension to general configurations

In this section, we extend our gathering algorithm to work also when the initial configuration is neither an A-type nor a B-type configuration. For the ease of readability, in order to not distinguish among subcases with a small number of robots, we consider only gatherable configurations with more than 18 robots.

Let a *C-type configuration* be any configuration with more than 18 robots which is not periodic, does not have an edge–edge symmetry, and is not an A-type configuration or a B-type configuration in the sense used in the proof of Theorem 3.1. Taking into account the procedure for gathering A-type and B-type configurations defined in the previous section, it now suffices to show that there exists a sequential approach for gathering C-type configurations, i.e., an approach such that, in every C-type configuration, exactly one specific robot is selected by the algorithm (regardless of the robot running the algorithm), and only this robot is then allowed to perform a move. Indeed, if the system starts in an A-type or B-type configuration, then we apply the approach from the previous section, and it remains in A-type configurations possibly alternating with B-type configurations. If the system starts in a C-type configuration, it may either perform a gathering passing through C-type configurations only using the sequential approach, or may at some point switch from a C-type configuration to a B-type configuration, and then remain confined to A-type and B-type configurations. As shown in Fig. 4, observe that C-type configurations necessarily belong to one of the following distinct subtypes.

C1: The configuration is *rigid*, i.e., is not periodic and has no axes of symmetry.
C2: The configuration has exactly one axis of symmetry of the node-on-axis type, with a robot on exactly one of its poles; the number of robots is odd.
C3: The configuration has exactly one axis of symmetry of the node-on-axis type, with robots on both of its poles; the number of robots and the length of the ring are even.

A gathering algorithm in [17] called RigidGathering provides a sequential procedure for gathering initial configurations which are rigid (progressing through rigid configurations in intermediate steps, only), as well as a sequential procedure for gathering C2-type initial configurations (by transformation to a rigid configuration). In order to provide a gathering algorithm that works for any initial gatherable configuration with more than 18 robots, it therefore suffices to show a sequential procedure for gathering a C3-type initial configuration. Hence the following theorem can be stated.

Theorem 4.1. *There exists a procedure for gathering any starting configuration of more than* 18 *robots on the ring, which is not periodic and does not have an edge–edge symmetry.*

Table 3
Feasibility of the gathering based on the number of robots.

Configuration	Number of robots	
	Odd	Even
Periodic	Not gatherable	
With only two robots	Not gatherable	
Rigid	Gatherable	
Edge–edge axis	Not possible	Not gatherable
Node-on-axis	Gatherable	Gatherable with more than 18 robots

Proof. As discussed above, it suffices to show a sequential procedure for gathering a C3-type initial configuration. First of all, note that, since the configuration has exactly one axis of symmetry, using a partial ordering of the robots, it is always possible to tell apart the two robots occupying the poles, hence allowing us to choose one of them. To complete the proof, we will show that, after a single movement of a robot from one of the poles (in an arbitrary direction), the resulting configuration is rigid, provided that the robot to be moved is chosen with the constraint that, if the robots occupying poles are adjacent to gaps of lengths differing in parity, then the robot adjacent to gaps of odd length is chosen to make the move. We will show that the new configuration does not have an edge–edge symmetry, a node-on-axis symmetry, and it is not periodic.

First of all, observe that any configuration which has an edge–edge symmetry fulfills the following constraints: (1) if there exist precisely a gaps in the configuration having length l, and a is odd, then l is also odd; (2) the set of robots which are end-points of at least one gap of even length has even cardinality. Now, observe that, if in a node-on-axis type configuration with robots on both poles, a robot adjacent to a gap of some odd length $2s+1$ makes a move, then the new configuration has an odd number of gaps of length $2s$; hence, constraint (1) is violated. Otherwise, suppose that both the robots on the poles are initially adjacent to gaps of even length; then, after one of the robots is moved off a pole, the set of robots which are end-points of at least one gap of even length will have odd cardinality (since it consists of some number of robots reflected into each other w.r.t. the original axis of symmetry and the unmoved robot on the pole); hence, constraint (2) is violated.

To show that the new configuration does not have a node-on-axis symmetry, recall that C3-type configurations consist of an even number of robots on an even ring, and notice that for a black–white 2-coloring of the ring, any configuration of the node-on-axis symmetry must contain an even number of robots occupying white nodes. Since the initial and new configuration differ in parity of the number of robots occupying white nodes, the new configuration cannot have node-on-axis symmetry.

Finally, suppose that the new configuration is periodic, and can be mapped onto itself by rotating the cycle by angle $2\pi/p$, $p \geq 2$. This means that for any robot r the lengths (distances on the ring) of the chains consisting of exactly k/p gaps, counting in the clockwise and anti-clockwise directions starting from r, must be identical. However, this is not the case for the robot which moved off the pole, since these lengths differ by precisely 2 due to the symmetry of the original configuration.

Hence, we obtain a rigid configuration, which completes the proof. □

5. Conclusions

We have studied the gathering problem in the discrete model, solving it on a ring for any number of robots larger than 18. The applied technique relies on preserving symmetries (in fact, our algorithm occasionally creates symmetric configurations from asymmetric initial configurations).

For configurations with more than 18 robots, our algorithm is complementary to the impossibility result shown in [17].

Theorem 5.1 (*[17]*). *Gathering is not feasible for initial configurations which are periodic or have an edge–edge symmetry.*

In this way, we have obtained the sought characterization of initial configurations on the ring.

Theorem 5.2. *For more than* 18 *anonymous and oblivious robots located on different nodes of a ring, gathering is feasible if and only if the initial configuration is not periodic and does not have an edge–edge symmetry.*

Theorem 5.2 implies that, for any number of robots larger than 18, gathering is feasible if and only if, in the initial configuration, the robots can elect a node (not necessarily occupied by a robot). Although it is conjectured in [17] that such a claim should also hold in cases with an even number of robots between 4 and 18 (as two robots are not gatherable), this is not always true. For instance, the only possible configuration of four robots on a five-node ring is not gatherable, although the single empty node can be initially elected as a candidate for the gathering point. Providing an additional characterization for the cases of between 4 and 18 robots is an interesting open problem. Some partial results in this direction have recently been shown in [15]. To summarize, in Table 3 all the gatherable configurations are shown.

A natural next step is to consider the gathering problem for other graph classes with high symmetry (such as tori), and if possible propose an algorithmic approach which solves the problem in the general case. The gathering problem could also be considered for variants of the model, such as robots having limited visibility, although such restrictions often lead to a large number of initial configurations for which gathering is impossible. It is not clear whether allowing robots to have small (constant) memory would help address such problems with achieving a gathering. Finally, it is interesting to ask whether the technique of preserving symmetries proposed herein can also be applied in other contexts.

References

[1] N. Agmon, D. Peleg, Fault-tolerant gathering algorithms for autonomous mobile robots, SIAM J. Comput. 36 (1) (2006) 56–82.
[2] S. Alpern, S. Gal, The Theory of Search Games and Rendezvous, Kluwer Academic Publishers, 2002.
[3] H. Ando, Y. Oasa, I. Suzuki, M. Yamashita, Distributed memoryless point convergence algorithm for mobile robots with limited visibility, IEEE Trans. on Robotics and Automation 15 (5) (1999) 818–828.
[4] M. Cieliebak, Gathering non-oblivious mobile robots, in: Proc. 6th Latin American Theoretical Informatics Symposium, LATIN, in: LNCS, vol. 2976, 2004, pp. 577–588.
[5] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Solving the robots gathering problem, in: Proc. 30th International Colloquium on Automata, Languages and Programming, ICALP, in: LNCS, vol. 2719, 2003, pp. 1181–1196.
[6] R. Cohen, D. Peleg, Robot convergence via center-of-gravity algorithms, in: Proc. 11th International Colloquium on Structural Information and Communication Complexity, SIROCCO, in: LNCS, vol. 3104, 2004, pp. 79–88.
[7] R. Cohen, D. Peleg, Convergence of autonomous mobile robots with inaccurate sensors and movements, SIAM J. Comput. 38 (1) (2008) 276–302.
[8] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoret. Comput. Sci. 355 (2006) 315–326.
[9] A. Dessmark, P. Fraigniaud, D. Kowalski, A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (2006) 69–96.
[10] P. Flocchini, D. Ilcinkas, A. Pelc, N. Santoro, Computing without communicating: ring exploration by asynchronous oblivious robots, in: Proc. 11th International Conference on Principles of Distributed Systems, OPODIS, in: LNCS, vol. 4878, 2007, pp. 105–118.
[11] P. Flocchini, D. Ilcinkas, A. Pelc, N. Santoro, Remembering without memory: tree exploration by asynchronous oblivious robots, in: Proc. 15th International Colloquium on Structural Information and Communication Complexity, SIROCCO, in: LNCS, vol. 5058, 2008, pp. 33–47.
[12] P. Flocchini, E. Kranakis, D. Krizanc, N. Santoro, C. Sawchuk, Multiple mobile agent rendezvous in a ring, in: Proc. 6th Latin American Theoretical Informatics Symposium, LATIN, LNCS, vol. 2976, 2004, pp. 599–608.
[13] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous robots with limited visibility, Theoret. Comput. Sci. 337 (1–3) (2005) 147–168.
[14] L. Gąsieniec, E. Kranakis, D. Krizanc, X. Zhang, Optimal memory rendezvous of anonymous mobile agents in a unidirectional ring, in: Proc. 32nd Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM, in: LNCS, vol. 3831, 2006, pp. 282–292.
[15] K. Haba, T. Izumi, Y. Katayama, N. Inuzuka, K. Wada, On gathering problem in a ring for 2n autonomous mobile robots, in: Proc. 10th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS (poster) 2008.
[16] R. Klasing, A. Kosowski, A. Navarra, Taking advantage of symmetries: gathering of asynchronous oblivious robots on a ring, in: Proc. 13th International Conference On Principle Of DIstributed Systems, OPODIS, in: LNCS, vol. 5401, 2008, pp. 446–462.
[17] R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theoret. Comput. Sci. 390 (1) (2008) 27–39.
[18] D. Kowalski, A. Pelc, Polynomial deterministic rendezvous in arbitrary graphs, in: Proc. 15th International Symposium on Algorithms and Computation, ISAAC, in: LNCS, vol. 3341, 2004, pp. 644–656.
[19] G. Prencipe, CORDA: distributed coordination of a set of autonomous mobile robots, in: Proc. European Research Seminar on Advances in Distributed Systems, ERSADS, 2001, pp. 185–190.
[20] G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoret. Comput. Sci. 384 (2–3) (2007) 222–231.
[21] I. Suzuki, M. Yamashita, Distributed anonymous mobile robots: formation of geometric patterns, SIAM J. Comput. 28 (4) (1999) 1347–1363.