About Ungatherability of Oblivious and Asynchronous Robots on Anonymous Rings

Gabriele Di Stefano[1], Pietro Montanari[2], and Alfredo Navarra[2](✉)

[1] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica,
Università degli Studi dell'Aquila, Via G. Gronchi 18, 67100 L'Aquila, Italy
gabriele.distefano@univaq.it

[2] Dipartimento di Matematica e Informatica, Università degli Studi di Perugia,
Via Vanvitelli, 1, 06123 Perugia, Italy
pietro.montanari@studenti.unipg.it, alfredo.navarra@unipg.it

Abstract. We investigate on gathering of identical, memoryless, and mobile robots placed on the nodes of anonymous graphs. According to the well-known *Look-Compute-Move* model, robots operate in asynchronous cycles. In one cycle, a robot takes a snapshot of the current configuration (Look), decides whether to stay idle or to move to one of its neighbors (Compute), and in the latter case makes the computed move (Move). Cycles are performed asynchronously for each robot. The gathering problem asks for a strategy that brings all robots to a common node.

Several papers have been investigating the problem for various settings on ring graphs due its combinatorial relevance. However, none of the provided solutions can cope with the case of four robots, the only case still open on ring graphs, even though it is conjectured that the gathering is possible. We consider the specific cases of four robots placed on a ring of seven and nine nodes. We present an exhaustive proof about the impossibility of designing a strategy that solves the gathering in the considered setting. The proof makes use of both theoretical and computer-assisted approaches. Despite the specific cases considered, the relevance of the provided proof is twofold. On the one hand, it disproves the conjecture posed by previous works. On the other hand, it provides a new approach and new insights to the gathering problem on rings.

1 Introduction

We study one of the basic problems concerning self-organization of mobile entities, known in the literature as the *gathering* problem. In particular, we consider oblivious (memoryless) robots initially located on different nodes of an anonymous ring that have to gather at a common node (not determined in advance) and there remain. Neither nodes nor edges are labeled. Initially, some of the nodes of the ring are occupied by the robots and there is at most one robot in

Work supported by the Italian Ministry of Education, University, and Research: PRIN 2010N5K7EB "ARS TechnoMedia" and PRIN 2012C4E3KT "AMANDA", and by the National Group for Scientific Computation (GNCS-INdAM).

Z. Lipták and W.F. Smyth (Eds.): IWOCA 2015, LNCS 9538, pp. 136–147, 2016.
DOI: 10.1007/978-3-319-29516-9_12

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 it 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 it is decided by the adversary for each robot. Hence, robots may move based on significantly outdated perceptions. 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. Robots are all identical, anonymous, and execute the same deterministic algorithm. They cannot leave any marks at visited nodes, nor send messages to other robots. This model is referred in the literature also as the *CORDA* model [9,16]. However, it is assumed that a robot has the ability to perceive during the Look operation whether there is one or more robots located at a given node of the ring, but not the exact number. This capability of robots is important and well-studied in the literature under the name of *multiplicity detection* (see e.g. [11] for a discussion), as a node with more than one robot located on it is called *multiplicity*. By definition, *initial* configurations do not contain multiplicities.

1.1 Related Works

The problem of let mobile entities meet on graphs or open spaces has been extensively studied in the last decades. When only two robots are involved, the problem is usually referred to as the *rendezvous* problem [3,4,7,15]. Under the Look-Compute-Move model, the rendezvous problem has been proved to be unsolvable on rings [13], hence instances with more than two robots have been investigated. 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 the literature, different types of configurations have required different approaches. In particular periodicity and symmetry arguments have been investigated for rings. A configuration is called *periodic* if it is invariable under non-trivial (i.e., non-complete) rotation. A configuration is called *symmetric* if the ring admits a geometrical *axis of symmetry* that reflects single robots into single robots, multiplicities into multiplicities, and empty nodes into empty nodes. A symmetric configuration admits a *node-edge symmetry* if the axis passes through one node and one edge (see, e.g. configurations (i)–(iii) in Fig. 1); an *edge-edge symmetry* if the axis passes through (the middles of) two edges; a *node-node symmetry* if the axis passes through two nodes; a *robot-on-axis symmetry* if there is at least one node on the axis occupied by a robot.

In [13], it is proved that the gathering is not feasible if the configuration is periodic, or symmetric of type edge-edge, or contains only two multiplicities, or if the multiplicity detection capability is removed. Then all configurations with an odd number of robots, and all the asymmetric configurations with an even number of robots have been solved by different algorithms. In [12], the problem was solved in the symmetric cases with an even number of robots greater than 18. This left open the cases of symmetric configurations of types node-node or node-edge with even number of robots between 4 and 18. The case of 4 robots has been addressed in [10,14]. In [10], symmetric configurations of type robot-on-axis with $2k$ robots, $k \geq 2$, have been addressed. Moreover, in [12] it has been observed that configurations of 4 robots on a five node ring are ungatherable. The case of 6 robots has been solved in [6]. Finally, in [5], a unified approach dealing with all the gatherable cases has been designed. Besides the new techniques, the algorithm also exploits some of the previous results. In particular, the resolution of symmetric configurations with only 4 and 6 robots is delegated to the previous algorithms from [6,14], respectively. However, as we will show, the algorithm proposed in [14] cannot cope with all the symmetric cases of 4 robots.

1.2 Our Results

Although all the configurations with 4 robots on rings with more than five nodes have been claimed to be gatherable as long as the initial configurations are asymmetric and aperiodic (from [13]), or symmetric of type node-edge, node-node, or robot-on-axis (from [14]), we revise the very special case of 4 robots on a seven and nine nodes ring. We then show there cannot exist any strategy allowing the gathering in such cases. The obtained result points out a twofold aspect. On the one hand, the obtained result disproves some claimed conjectures of previous works concerning gatherable configurations of 4 robots. In particular, the algorithm proposed in [14] cannot cope with all the symmetric cases of 4 robots on odd rings. On the other hand, the new approach exploited to prove the impossibility result provides useful advances in the study of the gathering task. It is worth remarking that configurations with few robots imply more difficulties in designing suitable gathering strategies as the movement of a robot easily incurs in making the current configuration symmetric or even periodic. Our main result is constituted by the following theorem:

Theorem 1. *Four robots on a seven or nine nodes ring are ungatherable.*

Indeed, the above theorem proved by means of theoretical and computer-assisted analysis reveals sufficient hints for a more general claim. Consider the intervals of free nodes between two nodes occupied by robots (an interval could be empty if the robots are adjacent). Let $SP4$ be the set of symmetric initial configurations with four robots on odd rings where the maximal odd interval of free nodes cut by the axis is bigger than the even one (see, e.g., configurations (i) and (ii) of Fig. 1, and configurations (i)–(iv) of Fig. 7).

The initial configuration of four robots on a five nodes ring belongs to $SP4$ and it can be proved to be ungatherable by an exhaustive proof on the possible moves that can be performed. Indeed, specific configurations in $SP4$ could be gatherable but requiring suitable strategies difficult to be generalized. The main difficulty faced when dealing with configurations in $SP4$ comes from the fact that among the two intervals cut by the axis, the odd one is bigger than the even one. In [8], it has been proved that the middle node of the odd interval is the only possible candidate to finalize the gathering. Hence, when robots move towards such a node , it may happen that only one of the two symmetric robots allowed to move makes the movement. The subsequent configuration contains now two intervals of even size corresponding to those intervals originally cut by the axis of symmetry. Possibly, they can be of the same size, hence inducing a different symmetry with respect to the original one.

Proving that initial configurations with four robots on a seven nodes ring are ungatherable is challenging as exploring exhaustively all the possible moves becomes computationally intractable. In fact, we exploit both theoretical and computer-assisted analysis to obtain the proof of Theorem 1.

2 Definitions and Notation

We consider anonymous rings without orientation consisting of either seven or nine nodes. Initially, four nodes of the ring are occupied by single robots. For instance, all possible initial configurations for a ring of seven nodes are shown in Fig. 1. During a Look operation, a robot perceives the relative locations on the ring of multiplicities and single robots. We remind that a multiplicity occurs when more than one robot occupy the same node.

The current configuration of the system can be described in terms of the view of a robot r that is performing the Look operation at the current moment. It is the sequence of robots, multiplicities and empty nodes seen by r starting from its position and proceeding towards an arbitrary direction. It follows that, given a configuration, a robot recognizes its own position in the ring if the configuration is asymmetric. In case of symmetry, the robot has two possible choices for its position. For instance, referring to Fig. 1, robots denoted x and x' are indistinguishable. In initial symmetric configuration we will denote x and x' the two robots closest to the node on the axis, whereas the other two nodes will be denoted y and y'.

In a symmetric configuration, the axis of symmetry passes through one node. A move of a robot r towards such a node is denoted by $r\uparrow$, while $r\downarrow$ denotes the move towards the opposite direction. In the asymmetric configuration (iv) in Fig. 1, the robots are referred to as a, b, c, and d. The move of a robot r in the direction of a robot r' is denoted by $r \rightarrow r'$.

The *strategy* of a gathering algorithm specifies for each configuration which is the robot that has to move and the direction of the movement. Note that in case of symmetric configuration or multiplicities a single robot can not be identified as well as a single direction can not be defined.

3 Four Robots on a Seven Nodes Ring

Consider four robots on a seven nodes ring, the only possible initial configurations are shown in Fig. 1. The first three configurations are symmetric, while the last one is asymmetric.

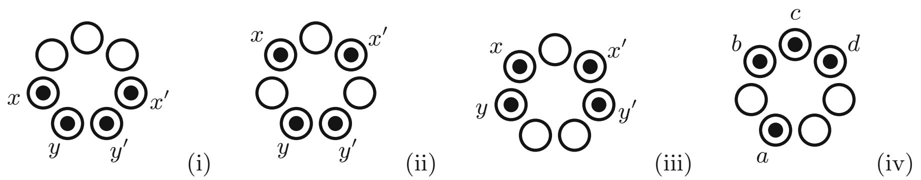


Fig. 1. The four possible initial configurations of four robots over a seven nodes ring. Edges between consecutive nodes are not drawn.

Being in an asynchronous system, it is clearly impossible to design a gathering algorithm that forces more than one robot to move from the same configuration. In fact, this would rely on the assumption that such robots have started their Look-Compute-Move cycles from the same configuration. Moreover, if an algorithm relies on the movement of more than one robot from a same configuration, the adversary can always force a robot to wake up after the movement of another robot, i.e. form different configurations.

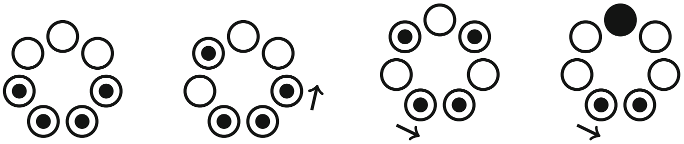
By the above discussion, the next two lemmata show that for each initial configuration there exists only one possible move that any gathering strategy can allow.

Lemma 1. *Let C be a symmetric initial configuration among (i), (ii), and (iii), a strategy to solve the gathering task can allow only the $x\uparrow$ move.*

Proof. Considering Fig. 1, let C coincide with (i). The move $x\downarrow$ may produce two multiplicities as well as $y\uparrow$, and from [13] it follows that such configurations are ungatherable. In fact, in a configuration composed of just two multiplicities, robots might behave exactly like in a configuration where there are only two single robots. By allowing $y\downarrow$, configuration (i) can be obtained infinitely many times as y and y' may move simultaneously and exchange their positions. Hence, only $x\uparrow$ remains. If C coincides with (ii), then $x\downarrow$ would output configuration (i) from which we have shown that only $x\uparrow$ is allowed, hence the configuration would cycle between (i) and (ii), infinitely many times. Move $y\uparrow$ could generate configuration (ii) itself if only one robot moves. Move $y\downarrow$ again generates (ii) infinitely many times. Again, only $x\uparrow$ remains. If C coincides with (iii), then $x\downarrow$ may produce two multiplicities as well as $y\uparrow$. Move $y\downarrow$ would outputs configuration (ii) from which we have shown that only $x\uparrow$ is allowed, hence the configuration would cycle between (ii) and (iii) if a single robot moves, infinitely times. It follows that $x\uparrow$ is the only possible move from each symmetric initial configuration. □

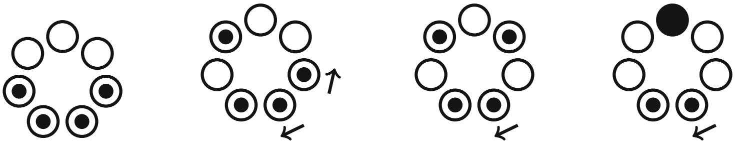
Lemma 2. *Let C be the asymmetric initial configuration (iv), a strategy to solve the gathering task can allow only the move $d \rightarrow c$.*

Proof. Let C be the asymmetric configuration (iv) shown in Fig. 1. If $a \rightarrow b$ is allowed, then configuration (i) is created, from which again configuration C can occur, by Lemma 1. If $a \rightarrow d$ is allowed, then again configuration (iv) is obtained with node a that should move backwards to the original position. If $b \rightarrow a$ is allowed, then configuration (iii) is created, from which again configuration (iv) can occur by applying the move of Lemma 1. If $b \rightarrow c$ is allowed, then the sequence of configurations shown below might occur. By the arrow on top of some robots we denote the decision made by the corresponding robot during its Compute operation to move towards the indicated direction. We refer to such a move as a *pending* move that will be performed during the Move operation, eventually.

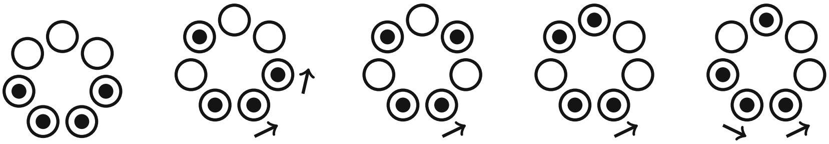


The above sequence of configurations shows that starting from (i), by Lemma 1, both x and x' can start their Look-Compte-Move cycle and move the configuration to (iv) with a pending move. Then, by hypothesis, $b \rightarrow c$ is applied. While $b \rightarrow c$ remains pending, the previous pending move is performed, leading the configuration to (iii) but with a pending move. Finally, again $x\uparrow$ is applied leading the configuration to have one multiplicity (represented by the full-black node) and a pending move. Since the last configuration might lead to two multiplicities, the considered move $b \rightarrow c$ cannot be allowed.

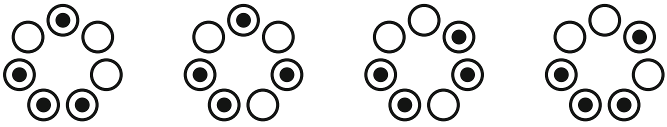
If $c \rightarrow b$ is allowed, then the sequence of configurations shown below might occur, leading to a configuration with two multiplicities.



If $c \rightarrow d$ is allowed, then the cycling sequence of configurations shown below might occur.



If $d \rightarrow a$ is allowed, then the cycling sequence of configurations shown below might occur.



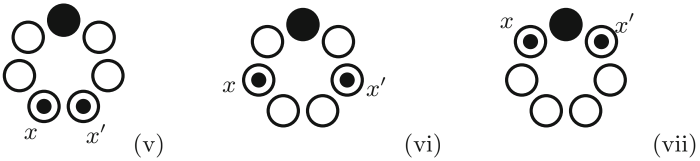


Fig. 2. The three possible symmetric configurations with one multiplicity.

Then the only move left is $d \to c$, and the claim follows. □

Lemma 3. *Let C be a symmetric configuration with one multiplicity, a strategy to solve the gathering task can allow only the $x\uparrow$ move.*

Proof. Let C be configuration (v) shown in Fig. 2. If x and x' move toward each other, the same configuration might be obtained. If robots composing the multiplicity are allowed to move, then configuration (ii) might be obtained it the two robots move simultaneously in opposite directions (note that no algorithm can move the two robots in the same direction due to the symmetry of the configuration). From (ii), by Lemma 2, again configuration (v) can be obtained.

From configuration (vi), if single robots are allowed to move toward each other, then configuration (v) can be obtained. By the above discussion, the two single robots should move back, and configuration (vi) would be again obtained. If robots composing the multiplicity are allowed to move, then configuration (iii) might be obtained. From (iii), by Lemma 2, again configuration (vi) can be obtained.

From configuration (vii), if single robots are allowed to move away from the multiplicity, then configuration (vi) can be obtained. By the above discussion, the two single robots should move back, and configuration (vii) would be again obtained. If robots composing the multiplicity are allowed to move, then a configuration with two multiplicity can be obtained.

In any case, the only move left concerns single robots to move toward the multiplicity. □

3.1 Further Simplifications

All configurations with two multiplicities must not be reached since they are ungatherable, hence any strategy does not need to specify a move for such configurations. Similarly, any configuration with a multiplicity made of four robots is final, hence no moves must be specified. All remaining configurations are reported in Fig. 3, and for each one, a strategy should specify one move. From (viii) to (xiii) there are six possible moves for each configuration to be tested, while the other configurations induce four moves each. Overall, there are still $6^6 * 4^3 = 2985984$ possible strategies to check. Testing all such strategies might be computationally prohibitive, so we need to eliminate some possibilities.

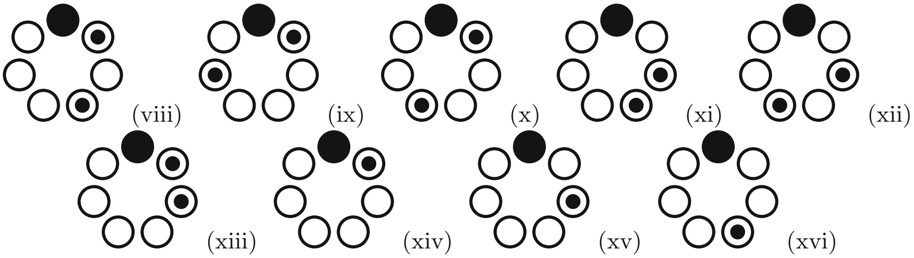


Fig. 3. All the remaining configurations that must be managed by a strategy.

For instance, from (viii), we cannot move the multiplicity toward left since if only one robot composing the multiplicity moves, then configuration (iv) is created and by Lemma 2 again configuration (viii) is obtained. From (ix) (from (xii), resp.), the move of the single robot closest to the multiplicity toward right would generate configuration (vi) (configuration (v), resp.) and by Lemma 3 again configuration (ix) (configuration (xii), resp.) can be obtained. From (x), moving the multiplicity to the right may generate the same configuration if only one robot moves. From (xi) and (xiii), we can avoid moving one single robot toward the other one as it would generate two multiplicities. From (xii), if the multiplicity is moved to the left, the same configuration is obtained. From (xiv), moving the multiplicity toward the single robot may produce two multiplicities, while the opposite move may generate (vii), and by Lemma 3, again configuration (xiv) can be obtained. From (xvi), moving the single robot or the multiplicity toward left may generate the same configuration.

Finally, since configuration (i) can generate any other initial configuration by applying the move of Lemma 1, when testing a strategy it can be discarded if it generates (i) since this implies a cyclic sequence of configurations. Then from (ix) and from (xiii), the move of the multiplicity toward left can be avoided. Similarly, the move of the multiplicity toward right in (xi) can be eliminated.

By removing all such moves, there remain 57600 possible strategies. In the next section, we make use of an automatic generator to check whether there is at least one strategy that allows gathering.

3.2 Computer-Assisted Results

We made use of the functional language OCaml [1]. Each strategy is represented by a string of 9 digits, corresponding to the nine configurations from (viii) to (xvi). The i-th digit j represents the j-th move associated with the i-th configuration. For instance, the moves associated with (xiv) are in order: the single robot moves away from the multiplicity or it moves toward the multiplicity. As shown in the previous section no further moves can be associated with configuration (xiv). It follows that in each string representing a strategy, the 7-th digit (corresponding to configuration (xiv)) ranges from 1 to 2. So the maximum string is 545343242 since there are 5 moves for (viii), 4 for (ix), and so forth.

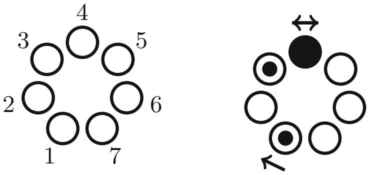


Fig. 4. The order of representation for each configuration and the representation of configuration $[(1,9);(0,0);(1,0);(2,14);(0,0);(0,0);(0,0)]$.

Each configuration is represented by a string of 7 pairs of integers. The first integer of each pair represents the number of robots lying in the corresponding node. The second integer represents the possible pending moves that corresponding robots might implement. As shown in Fig. 4, the correspondence of pairs with nodes is given by the clockwise order, starting from the bottom left node. For implementing pending moves, a robot is associated with 9 if its pending move is clockwise with respect to the current representation, with 5 if the pending move is anti-clockwise. Combinations of such numbers determine different pending moves provided by robots lying at the same node. The configuration shown in the figure admits a multiplicity with two robots that implement two opposite pending moves, and in fact, they are represented by the pair $(2,14)$.

For each strategy we explore the graph of configurations that can be obtained by starting from (i). A strategy is successful if all the branches lead to the final configuration, that is the generated graph is a tree, and each leaf is the final configuration. Contrary, we stop a test if a cycle is found or if a configuration with two multiplicities (and without pending moves) is generated. We then report in an output file the sequence of configurations that makes the tested strategy fail. For instance, strategy 422232221 corresponds to set of moves depicted in Fig. 5. This may induce the sequence of configurations depicted in Fig. 6 that ends up in a cycle given by configurations (7)–(9).

By exhaustively exploring all the 57600 strategies left, our computations show that there exists no successful strategy, that is, the first part of Theorem 1

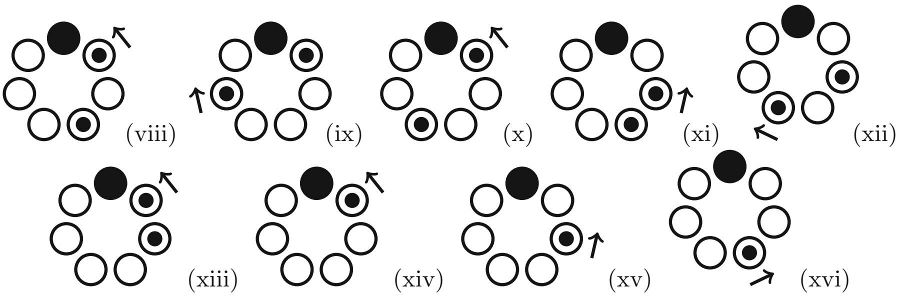


Fig. 5. Strategy 422232221. Arrows represent the defined moves.

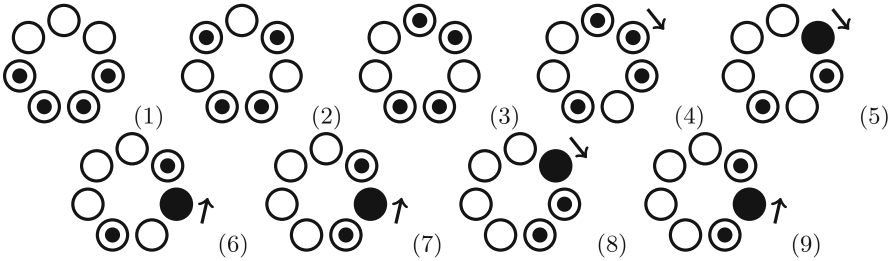


Fig. 6. Cyclic sequence of configurations generated by strategy 422232221.

is proven. Interested readers can found the implementation as well as the output of our computations in [2].

4 Four Robots on a Nine Nodes Ring

When considering four robots on a nine nodes ring, it can be easily verified there are 10 possible initial configurations out of which 6 are symmetric. Among those 6 configurations, 4 belong to $SP4$, see the first four configurations of Fig. 7. Recall from [8] that the only node where gathering can be potentially finalized in configurations belonging to $SP4$ is the middle one of the odd interval of free nodes cut by the axis. Moreover, it is worth noting that in order to gather a configuration belonging to $SP4$ it is necessary to reach a configuration not belonging to $SP4$ (like configuration (v) in Fig. 7). We now prove the second part of Theorem 1 by defining a specific behavior of the adversary. Starting from one configuration in $SP4$, whatever a gathering algorithm specifies to move, the adversary allows synchronous moves as long as the configuration remains in $SP4$, otherwise the adversary allows only one robot to move.

Similarly to the case shown for the seven nodes ring, from (i) only $x\uparrow$ can be allowed, hence reaching (ii). From (ii) only two moves can be allowed by any

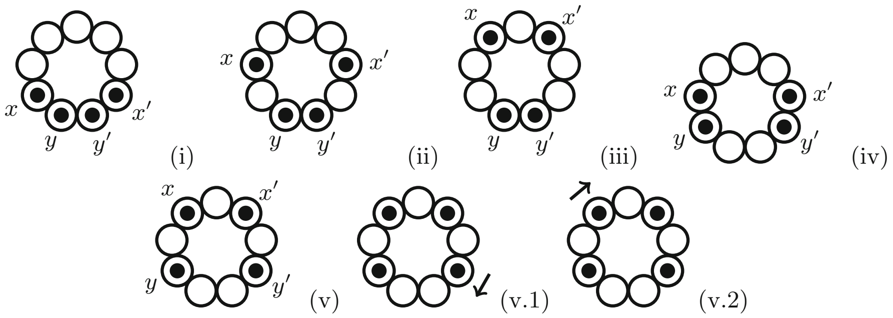


Fig. 7. The four possible initial configurations belonging to $SP4$ of four robots over a nine nodes ring and a configuration not in $SP4$ with two possible pending moves

gathering algorithm, that is $x\uparrow$ and $y\uparrow$ that lead to (iii) and (iv), respectively. From (iv), the only way to exit class $SP4$ is by $x\uparrow$, in which case the adversary makes only one robot move, hence leading to (iii). From (iii), the only ways to exit $SP4$ are by $x\uparrow$ or $y\uparrow$. By making move only one robot, in the former case (iv) is again obtained while in the latter case (v) is obtained with a possible pending move, as shown in (v.1).

From (v), by considering $x\uparrow$, the adversary can bring the configuration to (iii) starting from (v.1) and performing both the move of the pending robot and $x\uparrow$. Moves $x\downarrow$ and $y\downarrow$ bring to (iv) and (iii), respectively. So, it remains only $y\uparrow$. The adversary brings the configuration to (v.2) by starting from (v.1), performing the pending move and leaving pending $y\uparrow$ which now corresponds to $x\uparrow$. From (v.2), by performing both the pending moves $x\uparrow$ and $y\uparrow$, again (ii) is obtained.

We conclude there exists no strategy to exit class $SP4$.

5 Concluding Remarks

We have shown that the gathering of four robots on a seven or nine nodes ring is not feasible. Apart from its own interest as combinatorial problem, the obtained results disprove some previous works. It is worth nothing that some initial configurations are gatherable according to [13] if considered alone within the set of aperiodic and asymmetric configurations. Indeed, our results state the impossibility for gathering in general, i.e., when no assumptions are made on initial configurations. Moreover, configurations belonging to the set $SP4$ do not allow any gathering strategy even when considered the only possible initial configurations. Unfortunately, similar arguments applied for the case of nine nodes ring do not extend for rings of different size (including seven). Gathering configurations in $SP4$ with more than nine nodes still constitute an open problem. Actually, we have tried some strategies by extending the simulator described in Sect. 3.2 to gather configurations with eleven nodes but without succeeding so far. In fact, the conducted computer-assisted approaches seem to reveal the following statement:

Conjecture 1. Configurations in $SP4$ are ungatherable.

The intuition comes by observing that when starting from configurations where the four robots lay one after the other (like in configuration (i) of Fig. 1 and configuration (i) of Fig. 7), we know that by [8] the only node ν where gathering can be potentially finalized is the one opposite to the middle robots. To reach such a node, robots should be moved toward the other pole of the ring where ν lies. If they all move then it is possible to create a configuration similar to the initial one near to ν. If only two symmetric robots move, then orthogonal symmetries can easily occur.

References

1. http://caml.inria.fr/ocaml/
2. https://onedrive.live.com/?cid=7DEC52E233F33396&id=7DEC52E233F333961172
3. Alpern, S.: The rendezvous search problem. SIAM J. Control Optim. **33**, 673–683 (1995)
4. Czyzowicz, J., Labourel, A., Pelc, A.: How to meet asynchronously (almost) everywhere. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 22–30 (2010)
5. D'Angelo, G., Di Stefano, G., Navarra, A.: Gathering on rings under the look-compute-move model. Distrib. Comput. **27**(4), 255–285 (2014)
6. D'Angelo, G., Di Stefano, G., Navarra, A.: Gathering six oblivious robots on anonymous symmetric rings. J. Discrete Algorithms **26**, 16–27 (2014)
7. Dessmark, A., Fraigniaud, P., Kowalski, D., Pelc, A.: Deterministic rendezvous in graphs. Algorithmica **46**, 69–96 (2006)
8. Di Stefano, G., Navarra, A.: Optimal gathering of oblivious robots in anonymous graphs. In: Moscibroda, T., Rescigno, A.A. (eds.) SIROCCO 2013. LNCS, vol. 8179, pp. 213–224. Springer, Heidelberg (2013)
9. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Hard tasks for weak robots: the role of common knowledge in pattern formation by autonomous mobile robots. In: Aggarwal, A.K., Pandu Rangan, C. (eds.) ISAAC 1999. LNCS, vol. 1741, pp. 93–102. Springer, Heidelberg (1999)
10. Haba, K., Izumi, T., Katayama, Y., Inuzuka, N., Wada, K.:On gathering problem in a ring for 2n autonomous mobile robots. Technical report COMP2008-30, IEICE, Japan (2008)
11. Izumi, T., Izumi, T., Kamei, S., Ooshita, F.: Randomized gathering of mobile robots with local-multiplicity detection. In: Guerraoui, R., Petit, F. (eds.) SSS 2009. LNCS, vol. 5873, pp. 384–398. Springer, Heidelberg (2009)
12. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: gathering of many asynchronous oblivious robots on a ring. Theor. Comput. Sci. **411**, 3235–3246 (2010)
13. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theor. Comput. Sci. **390**, 27–39 (2008)
14. Koren, M.: Gathering small number of mobile asynchronous robots on ring. Zesz. Nauk. Wydzialu ETI Politech. Gdanskiej. Technol. Informacyjne **18**, 325–331 (2010)
15. Lim, W., Steve, A.: Minimax rendezvous on the line. SIAM J. Control Optim. **34**, 1650–1665 (1996)
16. Prencipe, G.: Impossibility of gathering by a set of autonomous mobile robots. Theor. Comput. Sci. **384**, 222–231 (2007)