Gathering Asynchronous Oblivious Mobile Robots in a Ring

Ralf Klasing[1], Euripides Markou[2,⋆], and Andrzej Pelc[3,⋆⋆]

[1] LaBRI - Université Bordeaux 1 - CNRS, 351 cours de la Libération, 33405 Talence cedex, France. E-mail: klasing@labri.fr
[2] School of Computational Engineering & Science, McMaster University, 1280 Main Street West, Hamilton, Ontario L8S4K1, Canada. E-mail: emarkou@di.uoa.gr
[3] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mail: pelc@uqo.ca

Abstract. We consider the problem of gathering identical, memoryless, mobile robots in one node of an anonymous unoriented ring. Robots start from different nodes of the ring. They operate in Look-Compute-Move cycles and have to end up in the same node. In one cycle, a robot takes a snapshot of the current configuration (Look), makes 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. For an odd number of robots we prove that gathering is feasible if and only if the initial configuration is not periodic, and we provide a gathering algorithm for any such configuration. For an even number of robots we decide feasibility of gathering except for one type of symmetric initial configurations, and provide gathering algorithms for initial configurations proved to be gatherable.

Keywords: asynchronous, mobile robot, gathering, ring.

1 Introduction

Mobile entities (robots), initially situated at different locations, have to gather at the same location (not determined in advance) and remain in it. This problem of distributed self-organization of mobile entities is known in the literature as the gathering problem. The main difficulty of gathering is that robots have to break symmetry by agreeing on a common meeting location. This difficulty is aggravated when (as in our scenario) robots cannot communicate directly but have to make decisions about their moves only by observing the environment.

We study the gathering problem in a scenario which, while very simple to describe, makes the symmetry breaking component particularly hard. Consider

⋆ Research partly supported by the European Research Training Network COMB-STRU HPRN-CT-2002-00278 and MITACS.
⋆⋆ Research partly supported by NSERC discovery grant, by the Research Chair in Distributed Computing at the Université du Québec en Outaouais, and by a visiting fellowship from LaBRI.

an unoriented anonymous ring of stations (nodes). Neither nodes nor links of the ring have any labels. Initially, some nodes of the ring are occupied by robots and there is at most one robot in each node. The goal is to gather all robots in one node of the ring and stop. Robots operate in Look-Compute-Move cycles. In one cycle, a robot takes a snapshot of the current configuration (Look), then, based on the perceived configuration, makes 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, while performing a move. 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 $t'' > t'$ in 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, which adds to the difficulty of achieving the goal of gathering. 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.

This very weak scenario, similar to that considered in [1, 3, 5, 6, 10, 13, 14], is justified by the fact that robots may be very small, cheap and mass-produced devices. Adding distinct labels, memory, or communication capabilities makes production of such devices more difficult, and increases their size and price, which is not desirable. Thus it is interesting to consider such a scenario from the point of view of applications. On the theoretical side, this weak scenario increases the difficulty of gathering by making the problem of symmetry breaking particularly hard, and thus provides an interesting setting to study this latter issue in a distributed environment.

It should be noted that the gathering problem under the scenario described above is related to the well-known leader election problem (cf. e.g. [12]) but is harder than it for the following reason. If robots in the initial configuration cannot elect a leader among nodes (this happens for all periodic configurations and for some symmetric configurations) then gathering is impossible (see Section 3). However, even if leader election is possible in the initial configuration, this does not necessarily guarantee feasibility of gathering. Indeed, while the node elected as a leader is a natural candidate for the place to gather, it is not clear how to preserve the same target node during the gathering process, due to its asynchrony. (Recall that nodes do not have labels, and configurations perceived by robots during their Look operation change during the gathering process, thus robots may not "recognize" the previously elected node later on.)

An important and well studied capability in the literature on robot gathering is the *multiplicity detection* [10, 14]. This is the ability of the robots to perceive, during the Look operation, if there is one or more robots in a given location. In our case, we prove that without this capability, gathering of more than one robot is always impossible. Thus we assume the capability of multiplicity detection in our further considerations. 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 are more than one robots: a robot does not see a difference between a node occupied by a or b robots, for distinct $a, b > 1$.

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. Robots move either in a graph, cf. e.g., [2, 7–9, 11], or in the plane [1, 3–6, 10, 13–15], they are labeled [7, 8, 11], or anonymous [1, 3–6, 10, 13–15], gathering algorithms are probabilistic (cf. [2] and the literature cited there), or deterministic [1, 3–7, 9–11, 13–15]. Deterministic algorithms for gathering robots in a ring (which is a task closest to our current setting) have been studied e.g., in [7–9, 11]. In [7, 8, 11] symmetry was broken by assuming that robots have distinct labels, and in [9] it was broken by using tokens.

To the best of our knowledge, the very weak assumption of anonymous identical robots that cannot send any messages and communicate with the environment only by observing it, was used to study deterministic gathering only in the case of robots moving freely in the plane [1, 3–6, 10, 13–15]. The scenario was further precised in various ways. In [4] it was assumed that robots have memory, while in [1, 3, 5, 6, 10, 13–15] 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, 15] 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 [4–6, 10, 13–15] 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 [10] 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 robots have the capability of multiplicity detection. A complementary negative result concerning the asynchronous model was proved in [14]: without multiplicity detection, gathering robots that do not have orientation is impossible.

Our scenario is the most similar to the asynchronous model used in [10, 14]. The only difference is in the execution of Move operations. This has been adapted to the context of the ring of stations (nodes): moves of the robots are executed instantaneously from a node to its neighbor, and hence robots always see other robots at nodes. All possibilities of the adversary concerning interleaving operations performed by various robots are the same as in the model from [10, 14], and the characteristics of the robots (anonymity, obliviousness, multiplicity detection) are also the same.

Our results. For an odd number of robots we prove that gathering is feasible if and only if the initial configuration is not periodic, and we provide a gathering algorithm for any such configuration. For an even number of robots we decide feasibility of gathering except for one type of symmetric configurations, and provide gathering algorithms for initial configurations proved to be gatherable.

Due to space limitations, most of the proofs have been omitted and will appear in the full version of the paper.

2 Terminology and Preliminaries

We consider an n-node anonymous unoriented ring. Initially, some nodes of the ring are occupied by robots and there is at most one robot in each node. The number of robots is denoted by k. During the gathering process robots move, and at any time they occupy some nodes of the ring, forming a *configuration*. A configuration is denoted by a pair of sequences $((a_1, \ldots, a_r), (b_1, \ldots, b_s))$, where the integers a_i and b_j have the following meaning. Choose an arbitrary node occupied by at least one robot as node u_1 and consider consecutive nodes $u_1, u_2, u_3, \ldots, u_r$, occupied by at least one robot, starting from u_1 in the clockwise direction. (Clockwise direction is introduced only for the purpose of definition, robots do not have this notion, as the ring is not oriented.) Integer a_i, for $i < r$, denotes the distance in the ring between nodes u_i and u_{i+1}, and integer a_r denotes the distance between nodes u_r and u_1 (in the clockwise direction). Next, consider those nodes among $u_1, u_2, u_3, \ldots, u_r$ which are occupied by more than one robot. Such nodes are called *multiplicities*. Suppose that $u_{v_1}, \ldots, u_{v_s}$ are these consecutive nodes (ordered in clockwise direction). Integer b_i is defined as the distance in the clockwise direction between node u_1 and node u_{v_i}. It should be clear that different choices of node 1 give rise to different pairs of sequences. Respective sequences in these pairs are cyclic shifts of each other and correspond to the same positioning of robots. So formally a configuration should be defined as an equivalence class of a pair of sequences with respect to those shifts. To simplify notation we will use pairs of sequences instead of those classes, and for configurations without multiplicities we will drop the second sequence, simply using sequence $(a_1, \ldots, a_r)$.

Consider a configuration $C = (a_1, \ldots, a_r)$ without multiplicities. The *range* of the configuration C is the set $\{a_1, \ldots, a_r\}$. For any integer a_i in the range of C, the *weight* of a_i is the number of times this integer appears in the sequence $(a_1, \ldots, a_r)$. C is called *periodic* if the sequence $(a_1, \ldots, a_r)$ is a concatenation of at least two copies of a subsequence p. The configuration C can be also represented as the set Z of nodes occupied by the robots. C is called *symmetric* if there exists an axis of symmetry of the ring, such that the set Z is symmetric with respect to this axis. If the number of robots is odd and S is an axis of symmetry of the set Z then there is exactly one robot on the axis S. This robot is called *axial* for this axis. Two robots are called *neighboring*, if at least one of the two segments of the ring between them does not contain any robots. A segment of the ring between two neighboring robots is called *free* if there is no robot in this segment.

We now describe formally what a robot perceives during a Look operation. Fix a robot R in a configuration represented by a pair of sequences $((a_1,\ldots,a_r),(b_1,\ldots,b_s))$, where this particular representation is taken with respect to the node occupied by R (i.e., this node is considered as node u_1). The *view* of robot R is the set of two pairs of sequences $\{((a_1,\ldots,a_r),(b_1,\ldots,b_s)), ((a_r,a_{r-1},\ldots,a_1),(n-b_s,\ldots,n-b_1))\}$ (if the node occupied by R is a multiplicity then we define the view of R as $\{((a_1,\ldots,a_r),(0,b_2,\ldots,b_s)), ((a_r,a_{r-1},\ldots,a_1),(0,n-b_s,\ldots,n-b_2))\}$). This formalization captures the fact that the ring is unoriented and hence the robot R cannot distinguish between a configuration and its symmetric image, if R is itself on the axis of symmetry. This is conveyed by defining the view as the *set* of the two couple of sequences because the sets $\{((a_1,\ldots,a_r),(b_1,\ldots,b_s)),((a_r,a_{r-1},\ldots,a_1),(n-b_s,\ldots,n-b_1))\}$ and $\{((a_r,a_{r-1},\ldots,a_1),(n-b_s,\ldots,n-b_1)),((a_1,\ldots,a_r),(b_1,\ldots,b_s))\}$ are equal. As before, if there are no multiplicities, we will drop the second sequence in each case and write the view as the set of two sequences: $\{(a_1,\ldots,a_r),(a_r,a_{r-1},\ldots,a_1)\}$. For example, in a 9-node ring with consecutive nodes $1,\ldots,9$ and three robots occupying nodes 1,2,4, the view of robot R at node 1 is the set $\{(1,2,6),(6,2,1)\}$.

A configuration without multiplicities is called *rigid* if the views of all robots are distinct. We will use the following geometric facts.

Lemma 1. *1. A configuration without multiplicities is non-rigid, if and only if it is either periodic or symmetric.*
2. If a configuration without multiplicities is non-rigid and non-periodic then it has exactly one axis of symmetry.

Consider a configuration without multiplicities that is non-rigid and non-periodic. Then it is symmetric. Let S be its unique axis of symmetry. If the number of robots is odd then exactly one robot is situated on S and S goes through the antipodal node if the size n of the ring is even, and through the (middle of the) antipodal edge if n is odd. If the number of robots is even then two cases are possible:

- *edge-edge symmetry* : S goes through (the middles of) two antipodal edges;
- *node-on-axis symmetry* : at least one node is on the axis of symmetry.

Note that the first case can occur only for an even number of robots in a ring of even size.

We now establish two basic impossibility results. Note that similar results have been proven for gathering robots on the plane. However, these results do not directly imply ours.

Proposition 1. *1. Gathering any 2 robots is impossible on any ring.*
2. If multiplicity detection is not available then gathering any $k > 1$ robots is impossible on any ring.

Proposition 1 justifies the two assumptions made throughout this paper: the number k of robots is at least 3 and robots are capable of multiplicity detection.

All our algorithms describe the Compute part of the cycle of robots' activities. They are written from the point of view of a robot R that got a view in a Look operation and computes its next move on the basis of this view.

The rest of the paper is organized as follows. In Section 3 we first establish two impossibility results: gathering is not feasible for periodic and edge-edge symmetric configurations. We then describe a procedure to gather configurations containing exactly one multiplicity and finally we propose a gathering procedure for rigid configurations. In Section 4 we give the complete solution of the gathering problem for any odd number of robots. Section 5 concludes the paper with a discussion of gathering for an even number of robots and with open problems.

3 Gatherable Configurations

In this section we first show two impossibility results. The first one concerns any number of robots, while the second one concerns only the case of an even number of robots on a ring of even size.

Theorem 1. *Gathering is impossible for any periodic configuration.*

Theorem 2. *Gathering is impossible for any edge-edge symmetric configuration.*

We now show a gathering procedure for any configuration containing exactly one multiplicity, say at node v.

Procedure Single-Multiplicity-Gathering

if R is at the multiplicity **then** do not move
else
    if none of the segments between R and the multiplicity is free
    then do not move
    else move towards the multiplicity along the shortest of the free segments or along any of them in the case of equality.

The idea is to gather all robots at v, avoiding creating another multiplicity (which could potentially create a symmetry, making the gathering process harder or even impossible). Procedure Single-Multiplicity-Gathering achieves this goal by first moving the robots closest to v towards v, then moving there the second closest robots, and so on.

Lemma 2. *Procedure Single-Multiplicity-Gathering performs gathering of robots for any configuration with a single multiplicity.*

Now we describe a gathering procedure for any rigid configuration, regardless of the number of robots.

The main idea of the procedure is to elect a single robot and move it until it hits one of its neighboring robots, thus creating a single multiplicity, and then to apply Procedure Single-Multiplicity-Gathering. The elected robot must be such that during its walk the rigidity property is not lost. In order to achieve this goal, we perform the election as follows. First the robots elect a pair of neighboring robots at maximum distance (there may be several such pairs, whence the need for election). Then they choose among them the robot which has the other neighboring robot closer. Ties can be broken easily.

In order to elect a robot we need to linearly order all possible views. This can be done in many ways. One of them is to order lexicographically all finite sequences of integers and number them by consecutive natural numbers. Then a view becomes a set of two natural numbers. Treat these sets as ordered pairs of natural numbers in increasing order, order these pairs lexicographically, and assign them consecutive natural numbers in increasing order. We fix the resulting linear order of views and this numbering beforehand, adding it to the algorithm for all robots. We call this procedure *Rigid-Gathering*.

Lemma 3. *Procedure Rigid-Gathering performs gathering of robots for any rigid configuration without multiplicities.*

4 Gathering an Odd Number of Robots

In this section we present a gathering algorithm for any non-periodic configuration of an odd number of robots. Together with Theorem 1 this solves the gathering problem for an odd number of robots.

Algorithm Odd-Gathering

if the configuration is periodic **then** output: gathering impossible
else
    if the configuration has a single multiplicity
    then Single-Multiplicity-Gathering
    else
        if the configuration is rigid **then** Rigid-Gathering
        else
            if R is axial **then** move (to any of the adjacent nodes)

The idea of the algorithm is the following. Consider any non-periodic configuration of an odd number of robots (recall that initially there are no multiplicities). If it is rigid then apply Procedure Rigid-Gathering. Otherwise it must be symmetric, by Lemma 1. There is a unique axial robot for its unique axis of symmetry. Move this robot to any adjacent node. We prove that three cases can occur. (1) The resulting situation has a multiplicity (the adjacent node was occupied by a robot): then apply Procedure Single-Multiplicity-Gathering. (2) The resulting configuration is rigid: then apply Procedure Rigid-Gathering. (3) Another axis of symmetry has been created (the previous one has been obviously destroyed by the move). In this case there is a unique axial robot for the unique axis of symmetry. Move this robot to any adjacent node. Again one of the three above cases can occur. We prove that after a finite number of such moves, only cases (1) or (2) can occur, and thus gathering is finally accomplished either by applying Procedure Single-Multiplicity-Gathering or by applying Procedure Rigid-Gathering. In the proof of the correctness of Algorithm Odd-Gathering we will use the following lemmas.

Lemma 4. *Let C be a symmetric configuration of an odd number of robots, without multiplicities. Let C' be the configuration resulting from C by moving the axial node to any of the adjacent nodes. Assume that C' does not have multiplicities. Then C' is not periodic.*

Let C be a symmetric non-periodic configuration of an odd number of robots, without multiplicities. The unique value of odd weight in the configuration C is called the *chief* of C. Let C' be the configuration resulting from C by moving the axial robot to any of the adjacent nodes. If C' does not have multiplicities and is symmetric then we will call it *special*. The subset of the range of a special configuration C' consisting of integers of the same parity as that of the chief is called the *white part* of the range, and its complement is called the *black part* of the range. We denote by $b(C')$ the total number of occurrences in C' of integers from the black part of its range.

Lemma 5. *Consider a sequence $(C_1, C_2, \ldots)$ of special configurations, such that C_{i+1} results from C_i by moving the axial robot to any of the adjacent nodes. Then for some $i \leq k$, we have $b(C_i) = 0$.*

Lemma 6. *Consider a special configuration C, with $b(C) = 0$. Let C' be the configuration resulting from C by moving the axial robot to any of the adjacent nodes. If C' does not have multiplicities then it is not symmetric.*

We are now ready to prove the correctness of Algorithm Odd-Gathering.

Theorem 3. *Algorithm Odd-Gathering performs gathering of any non-periodic configuration of an odd number of robots.*

Proof. Consider an initial non-periodic configuration C of an odd number of robots. By assumption it does not contain multiplicities. If it is rigid then we are done by Lemma 3. Otherwise, it must be symmetric by Lemma 1. Let A be its unique axial robot. Let C_1 be the configuration resulting from C by moving robot A to any of the adjacent nodes. If C_1 contains a multiplicity then we are done by Lemma 2. If C_1 is rigid then we are done by Lemma 3. Otherwise, C_1 is either periodic or symmetric, in view of Lemma 1. By Lemma 4, it cannot be periodic, hence it must be symmetric, and thus special. Consider the configuration C_2 resulting by moving the axial robot of C_1 to any of the adjacent nodes. Again C_2 either contains a multiplicity, or is rigid, or is special. In the first two cases we are done, and in the third case the axial robot is moved again. In this way we create a sequence $C_1, C_2, \ldots$ of special configurations. By Lemma 5, there is a configuration C_i in this sequence, with $b(C_i) = 0$. Let C' be the configuration resulting from C_i by moving the axial robot to any of the adjacent nodes. By Lemma 6, the configuration C' either has a multiplicity, or cannot be symmetric, and thus must be rigid. In the first case we are done by Lemma 2 and in the second case by Lemma 3. □

Theorem 3 and Theorem 1 imply the following corollary.

Corollary 1. *For an odd number of robots, gathering is feasible if and only if the initial configuration is not periodic.*

5 Conclusion

We completely solved the gathering problem for any odd number of robots, by characterizing configurations possible to gather (these are exactly non-periodic configurations) and providing a gathering algorithm for all these configurations. Corollary 1 is equivalent to the following statement: for an odd number of robots, gathering is feasible if and only if in the initial configuration, robots can elect a node occupied by a robot.

For an even number of robots, we proved that gathering is impossible if either the number of robots is 2, or the configuration is periodic, or when it has an edge-edge symmetry. On the other hand, we provided a gathering algorithm for all rigid configurations. This leaves unsettled one type of configurations: symmetric non-periodic configurations of an even number of robots with a node-on-axis type of symmetry. These are symmetric non-periodic configurations in which at least one node is situated on the unique axis of symmetry. This (these) node(s) may or may not be occuppied by robots. In this case, the symmetry can be broken by initially electing one of the axial nodes. This node is a natural candidate for the place to gather. However, it is not clear how to preserve the same target node during the gathering process, due to its asynchrony. Unlike in our gathering algorithm for an odd number of robots, where only one robot moves until a multiplicity is created, in the case of the above symmetric configuration of an even number of robots, some robots would have to move together. This creates many possible outcomes of Look operations for other robots, in view of various possible behaviors of the adversary, which can interleave their actions. We note here that for an even number of robots there are cases where gathering is feasible even when robots cannot initially elect a node occupied by a robot (they will be included in the full version of the paper).

The complete solution of the gathering problem for an even number of robots remains a challenging open question left by our research. We conjecture that in the unique case left open (non-periodic configurations of an even number of robots with a node-on-axis symmetry), gathering is always feasible. In view of our results, this is equivalent to the following statement.

Conjecture: For an even number of more than 2 robots, gathering is feasible if and only if the initial configuration is not periodic and does not have an edge-edge symmetry.

The validity of this conjecture would imply that, for any number of more than 2 robots, gathering is feasible if and only if, in the initial configuration robots can elect a node (not necessarily occupied by a robot).

References

1. N. Agmon, D. Peleg: Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots. SIAM J. Comput. 36(1): 56-82 (2006).
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): 818-828 (1999).
4. M. Cieliebak: Gathering Non-oblivious Mobile Robots. Proc. 6th Latin American Symposium on Theoretical Informatics (LATIN 2004): 577-588.
5. M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro: Solving the Robots Gathering Problem. Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 2003), LNCS 2719: 1181-1196.
6. R. Cohen, D. Peleg: Robot Convergence via Center-of-Gravity Algorithms. Proc. 11th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2004), LNCS 3104: 79-88.
7. G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro: Asynchronous deterministic rendezvous in graphs. Proc. 30th Inter. Symp. on Mathematical Foundations of Computer Science, (MFCS 2005), LNCS 3618: 271-282.
8. A. Dessmark, P. Fraigniaud, D. Kowalski, A. Pelc: Deterministic rendezvous in graphs. Algorithmica, to appear.
9. P. Flocchini, E. Kranakis, D. Krizanc, N. Santoro, C. Sawchuk: Multiple Mobile Agent Rendezvous in a Ring. Proc. 6th Latin American Symposium on Theoretical Informatics (LATIN 2004): 599-608.
10. P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer: Gathering of Asynchronous Robots with Limited Visibility. Theoretical Computer Science 337(1-3): 147-168 (2005).
11. D. Kowalski, A. Pelc: Polynomial deterministic rendezvous in arbitrary graphs. Proc. 15th Annual Symposium on Algorithms and Computation (ISAAC'2004), LNCS 3341: 644-656.
12. N. Lynch: Distributed Algorithms, Morgan Kaufman 1996.
13. G. Prencipe: CORDA: Distributed Coordination of a Set of Autonomous Mobile Robots. Proc. ERSADS 2001: 185-190.
14. G. Prencipe: On the Feasibility of Gathering by Autonomous Mobile Robots. Proc. 12th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2005), LNCS 3499: 246-261.
15. I. Suzuki, M. Yamashita: Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. SIAM J. Comput. 28(4): 1347-1363 (1999).
16. M. Yamashita, T. Kameda: Computing on Anonymous Networks: Parts I and II. IEEE Trans. Parallel Distrib. Syst. 7(1): 69-96 (1996).