Gathering on a Circle with Limited Visibility by Anonymous Oblivious Robots

Giuseppe A. Di Luna

DIAG, Sapienza University of Rome, Italy
diluna@diag.uniroma1.it

Ryuhei Uehara

School of Information Science, JAIST, Ishikawa, Japan
uehara@jaist.ac.jp

Giovanni Viglietta[1]

School of Information Science, JAIST, Ishikawa, Japan
johnny@jaist.ac.jp

Yukiko Yamauchi

Department of Informatics, Graduate School of ISEE, Kyushu University, Japan
yamauchi@inf.kyushu-u.ac.jp

Abstract

A swarm of anonymous oblivious mobile robots, operating in deterministic Look-Compute-Move cycles, is confined within a circular track. All robots agree on the clockwise direction (chirality), they are activated by an adversarial semi-synchronous scheduler (SSYNCH), and an active robot always reaches the destination point it computes (rigidity). Robots have limited visibility: each robot can see only the points on the circle that have an angular distance strictly smaller than a constant ϑ from the robot's current location, where $0 < \vartheta \leq \pi$ (angles are expressed in radians).

We study the Gathering problem for such a swarm of robots: that is, all robots are initially in distinct locations on the circle, and their task is to reach the same point on the circle in a finite number of turns, regardless of the way they are activated by the scheduler. Note that, due to the anonymity of the robots, this task is impossible if the initial configuration is rotationally symmetric; hence, we have to make the assumption that the initial configuration be rotationally asymmetric.

We prove that, if $\vartheta = \pi$ (i.e., each robot can see the entire circle except its antipodal point), there is a distributed algorithm that solves the Gathering problem for swarms of any size. By contrast, we also prove that, if $\vartheta \leq \pi/2$, no distributed algorithm solves the Gathering problem, regardless of the size of the swarm, even under the assumption that the initial configuration is rotationally asymmetric and the visibility graph of the robots is connected.

The latter impossibility result relies on a probabilistic technique based on random perturbations, which is novel in the context of anonymous mobile robots. Such a technique is of independent interest, and immediately applies to other Pattern-Formation problems.

2012 ACM Subject Classification Computing methodologies → Distributed algorithms; Theory of computation → Self-organization

Keywords and phrases Mobile robots, Gathering, limited visibility, circle

Digital Object Identifier 10.4230/LIPIcs.DISC.2020.12

Related Version A full version of the paper is available at `https://arxiv.org/abs/2005.07917`.

Acknowledgements The authors would like to thank the anonymous reviewers for greatly improving the readability of this paper.

[1] Corresponding author

34th International Symposium on Distributed Computing (DISC 2020).
Editor: Hagit Attiya; Article No. 12; pp. 12:1–12:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Background

One of the most popular models for distributed mobile robotics is the *Look-Compute-Move* (LCM) one [17, 18]. In this model, a Euclidean space, usually the real plane $\mathbb{R}^2$, is inhabited by a "swarm" of punctiform and autonomous computational entities, the *robots*. Each robot, upon activation, takes a snapshot of the space (Look), uses this snapshot to compute its destination (Compute), and then reaches its destination point (Move).

The activation pattern of the robots is controlled by an external scheduler. At one end of the synchrony spectrum, there is the *fully-synchronous* scheduler (FSYNCH): in this case, time is divided into discrete units (the *turns*). At each turn, the entire swarm is activated, and all robots synchronously execute one LCM cycle. At the other end, there is the *asynchronous* scheduler (ASYNCH), where robot activations are independent, and LCM cycles are not synchronized. Somewhere halfway, there is the *semi-synchronous* scheduler (SSYNCH), which activates an arbitrary subset of robots at each turn, with the restriction of activating each robot infinitely often.

A common assumption in the context of mobile robots is the lack of persistent memory: a robot does not remember anything about past activations (*obliviousness*). Other assumptions are *anonymity*, where robots do not have visible and distinguishable identifying features, and *silence*, where robots do not have explicit communication primitives (such as lights [14]).

Obliviousness, anonymity, and silence are practical, useful, and desirable properties: an algorithm for oblivious robots is inherently resilient to transient memory failures; one for anonymous robots is ideal in privacy-sensitive contexts; an algorithm for silent robots works even in scenarios where communication is jammed or unfeasible (e.g., hostile environments or underwater deployment).

The purpose of such an ensemble of weak robots is to reach a common goal in a coordinated way. Interestingly, it has been shown that mobile robots can solve an extensive set of problems [18], ranging from forming patterns [8, 19, 21, 28, 30] to simulating a powerful Turing-complete movable entity [12].

Among all tasks, a particularly relevant one is *Gathering* [1, 4, 5, 9, 13, 15, 25, 26]: in finite time, all robots have to reach the same point and stop there. Initial works assumed robots to see the entire space (*full visibility*). However, a more realistic assumption [3, 24] is that a robot be able to see only a portion of the space (*limited visibility*).

In this paper, we study the Gathering problem for a swarm of oblivious robots with limited visibility constrained to move within a circle: each robot can see only the points on the circle that have an angular distance strictly smaller than a certain *visibility range* ϑ. We assume that robots have no agreement on common coordinates apart from sharing the same notion of clockwise direction on the circle.

From a practical perspective, the restriction of moving along a predetermined path arises in wide variety of scenarios: railway lines, roads, tunnels, waterways, etc. We argue that, among all topologically equivalent curves, the circle is the most meaningful to study: a solution for it readily extends to all other closed curves.

From a theoretical perspective, confining the swarm on a circle (hence, a non-simply connected space) rules out all the strategies typically used for robots in the plane, such as moving toward the center of the visible set of robots (an example is in [2]). Moreover, robots cannot use any asymmetries in the environment to identify a gathering point: this makes the circle the most challenging setting for Gathering (and in general, for any problem where symmetry breaking helps).

Apart from [16], which examined the problem of scattering on a circle (reaching a final configuration where the robots are uniformly spaced out), no other works studied the computational power of oblivious robots when confined to curves: this is rather surprising, considering the copious existing literature on oblivious robots [17, 18]. To the best of our knowledge, the present paper is the first to investigate the Gathering problem for oblivious, silent, and anonymous robots on a circle with limited visibility (some works investigated the Gathering problem in a ring graph, which is a discretization of the circle [6, 7, 11, 22, 23]).

Our contributions

We consider a swarm of n oblivious, anonymous, and silent robots that start at distinct locations on a circle. Robots do not agree on a common system of coordinates, but they do share the same *handedness* (i.e., they have a common notion of clockwise direction). When a robot decides to move, it reaches its destination point (robots are *rigid*). Moreover, robots have no information on the swarm's size, n. Each robot can see only the points on the circle that have an angular distance strictly smaller than a certain visibility range ϑ. We must assume that the initial configuration is rotationally asymmetric, otherwise the scheduler may activate robots in a such a way as to preserve the rotational symmetry, and Gathering cannot be achieved.

After giving all the necessary definitions and some preliminary results (Section 2), our first contribution is to show that there is no distributed algorithm that solves the Gathering problem in SSYNCH when $\vartheta \leq \pi/2$, i.e., each robot is only able to see at most half of the circle (Section 3). Surprisingly, this holds even if the initial configuration is rotationally asymmetric, the visibility graph of the swarm (i.e., the graph of intervisibility between robots) is connected, and all robots know n.

Our proof uses a novel technique based on random perturbations, of which we offer an intuitive probabilistic argument, as well as a formal and more elementary proof by derandomization. We show that, for any given distributed algorithm, either there exists an asymmetric configuration of robots that can evolve into a symmetric one within one time unit (in SSYNCH), or there is an asymmetric configuration where no robot can move. In either case, Gathering is impossible.

We stress that our result has a profound meaning, since it shows that, when $\vartheta \leq \pi/2$, any distributed algorithm, including the ones that do not aim to solve Gathering, has an initial asymmetric configuration that either repeats forever or evolves into a symmetric configuration in one step. This implies a novel impossibility result for geometric Pattern Formation on circles: even when robots start from an asymmetric configuration, they cannot form a target asymmetric pattern. This is in striking contrast with the unlimited-visibility setting, where, even under the ASYNCH scheduler, from any asymmetric configuration any pattern is formable [18].

To the best of our knowledge, this the first impossibility proof for oblivious robots that neither relies on invariants induced by symmetries (e.g., [21, 29]) nor on the disconnection of the visibility graph (e.g., [12, 31]). Due to the above, we think that our technique is of independent interest, and its core ideas could be applied to other settings, as well.

On the possibility side, we show that, if $\vartheta = \pi$ (i.e., each robot can see the entire circle except its antipodal point), there is a distributed algorithm that solves the Gathering problem in SSYNCH for swarms of any size (Section 4). The algorithm's strategy is to attempt to elect a unique leader and form a multiplicity point, where all robots will subsequently gather. The main challenge is that, since a robot ignores whether its antipodal point is occupied by another robot or not (robots do not know n), there may be an ambiguity on who is the true leader. Several robots may believe to be the leader, but this also comes with the awareness of the possibility of being wrong: these “undecided” robots will make some adjustment moves, which eventually result in a configuration where one robot is absolutely certain of being the true leader. The leader will then form a multiplicity point by moving to another robot, and finally all other robots will join them.

2 Model definition and preliminaries

Measuring angles

Let $C \subset \mathbb{R}^2$ be a circle, and let a and b be points of C. The *angular distance* between a and b (with respect to C) is the measure of the angle subtended at the center of C by the shorter arc with endpoints a and b. It follows that the angular distance between two points is a real number in the interval $[0, \pi]$, where angles are expressed in radians. Two points of C are *antipodal* of each other if their angular distance is π. The α-*neighborhood* of a point $q \in C$ is the set of points of C whose angular distance from q is strictly smaller than α. The $(\pi/2)$-neighborhood of q is also called the *open semicircle* centered at q.

Furthermore, if a and b are distinct points of C, we define $cw(a, b)$ as the measure of the *clockwise* angle $\angle acb$, where c is the center of C. Note that the order of the two arguments matters, and so for instance $cw(a, b) + cw(b, a) = 2\pi$. We also define $cw(a, a) = 0$ for every a.

Rotational symmetry

Let S be a finite multiset of points on a circle C. We say that S is *rotationally symmetric* if there is a non-identical rotation around the center of C that leaves S unchanged (also preserving multiplicities). If S is not rotationally symmetric, it is said to be *rotationally asymmetric*.

Angle sequences

Let S be a multiset of n points on a circle C, and let $p \in S$. Let $p_1, p_2, \ldots, p_n$ be the points of S taken in clockwise order starting from $p = p_1$ (coincident elements of S are ordered arbitrarily). We define the *angle sequence* of p (with respect to S) as the n-tuple $(cw(p_1, p_2), cw(p_2, p_3), \ldots, cw(p_n, p_1))$. The case where all the elements of S are coincident is an exception, and in this case the angle sequence of the ith point of S, with $1 \le i \le n$, is defined as the n-tuple $(0, 0, \ldots, 0, 0, 2\pi, 0, 0, \ldots, 0, 0)$, where the term 2π appears in the ith position. Note that, with this convention, the sum of the elements of any angle sequence is always 2π.

The following is an easy observation.

▶ **Proposition 2.1.** *A non-empty multiset of points on a circle is rotationally asymmetric if and only if all its points have distinct angle sequences.* ◀

Mobile robots

Our model of mobile robots is among the standard ones defined in [17, 18]. A swarm of $n > 1$ robots is located on a circle $C \subset \mathbb{R}^2$, where each robot is a computational unit that occupies a point of C (which may change over time) and operates in deterministic Look-Compute-Move cycles.

Time is discretized and subdivided into units, and at each time unit an adversarial (*semi-synchronous*) *scheduler* decides which robots are active and which are inactive. An inactive robot remains idle for that time unit, whereas an active robot takes a *snapshot* of its surroundings, consisting of an arc $B \subseteq C$ and a list of points of B that are currently occupied by robots, it computes a destination point in B as a function of the snapshot, and it instantly moves to the destination point. The only restriction to the scheduler is that no robot should remain inactive for infinitely many consecutive time units.

Robots may have *full visibility*, in which case the arc B defining a snapshot coincides with the entire circle C, or they may have *limited visibility*, in which case the arc B consists of the ϑ-neighborhood of the current position of the robot taking the snapshot, where ϑ is a positive constant called the *visibility range* of the robots.

Furthermore, each robot has its own *local coordinate system*, meaning that each snapshot it takes of an arc $B \subseteq C$ is actually a roto-translated copy of B and the positions of the robots within B. Such a copy of B has its midpoint at the origin of the coordinate system (this corresponds to the location of the robot taking the snapshot) and its endpoints have non-negative x coordinate and the same y coordinate.

Robots are also capable of *weak multiplicity detection*, meaning that the snapshots they take contain some information on how many robots occupy each location. Specifically, a robot can tell if a point in a snapshot contains no robots, exactly one robot, or more than one robot: no information on the precise number of robots is given if this number is greater than one. A point occupied by more than one robot is also called a *multiplicity point*.

In order to simplify our notation, when no confusion arises, we will often identify a robot with its position on the circle. So, we may improperly refer to a robot as a point $p \in C$ or to a swarm of robots as a set $S \subset C$.

Gathering

A *distributed algorithm* is a function that maps a snapshot to a point within the snapshot itself. A robot *executes* a distributed algorithm A if, whenever it is activated and takes a snapshot Q, it moves to the destination point corresponding to $A(Q)$. In other words, at each time unit, an active robot chooses its destination point deterministically within its visibility range, based solely on the snapshot it currently has.

We stress that, as a consequence of the previous definitions, the robots in this model are *oblivious* (i.e., they have no memory of past observations), *anonymous* (i.e., a robot only identifies other robots by their positions in its local coordinate system, and not for instance by their IDs), *silent* (i.e., they cannot send messages to one another), *deterministic* (i.e., they cannot flip coins), *rigid* (i.e., they always reach the destination points they compute), they have *chirality* (i.e., they all agree on the clockwise direction on the circle), and they have no knowledge of n (i.e., a robot can only see other robots within its visibility range, and it does not know whether there are further robots outside of it).

We say that a distributed algorithm A solves the *Gathering problem* under condition P if, whenever all the $n > 1$ robots of a swarm located on a circle execute A, they eventually reach a configuration where all robots are in the same point of the circle and no robot ever moves again, provided that their initial configuration satisfies condition P, and regardless of the activation choices of the adversarial scheduler. Equivalently, we say that A is a *Gathering algorithm* under condition P.

We remark that all the robots in the swarm must execute the same algorithm A (i.e., robots are *uniform*), and the algorithm has to work for swarms of any size $n > 1$, where n is *not* a parameter of A. Also note that the robots' positions should not simply converge to the same limit, but they must actually become coincident in a finite number of time units for Gathering to be achieved (albeit there is no bound on the number of time units this process may take). Such a distinction will be important for the design of a correct Gathering algorithm in Section 4, while our impossibility proof of Section 3 shows that robots cannot even converge to a point, and much less gather.

Initial conditions

There are several meaningful options concerning our choice of the initial condition P for the Gathering problem. A typical assumption is that the n robots be initially located in n distinct points of the circle: while not strictly necessary, this is a common requirement for the Gathering problem (e.g., [4, 15, 28]).

Another assumption that we may make is that the *visibility graph* of the robots be initially connected. By "visibility graph" we mean the graph whose nodes are the n robots, where there is an edge between two robots if and only if they are mutually visible, i.e., if their angular distance is less than ϑ. This assumption is another common one (e.g., [2, 3, 10, 20, 27]) and, although not strictly necessary, it is justified by the intuition that different connected components of the visibility graph may never become aware of each other, and therefore may fail to gather. We will make this assumption in Section 3 to strengthen our impossibility result, and we will not need to explicitly make it in Section 4, because it will come as a consequence of other assumptions.

An important mandatory condition is that the multiset of the robots' positions on the circle should be rotationally asymmetric, due to the following.

▶ **Proposition 2.2.** *Let S be any rotationally symmetric multiset of $n > 1$ points on a circle. There is no Gathering algorithm under the initial condition that the multiset of robots' positions is S.* ◀

Since the robots are oblivious, this condition should hold true not only at the beginning, but at all times during the execution of a Gathering algorithm: the robots should never "accidentally" form a rotationally symmetric multiset, or they will be unable to gather.

▶ **Corollary 2.3.** *Throughout the execution of any Gathering algorithm, the robots' positions must always form a rotationally asymmetric multiset.* ◀

3 Impossibility of Gathering for $\vartheta \leq \pi/2$

Outline

In this section we prove that, if each robot can see at most an open semicircle (i.e., $\vartheta \leq \pi/2$), then no distributed algorithm solves the Gathering problem, even under some strong assumptions on the initial configuration, and even if the robots know the size of the swarm.

Our technique is essentially probabilistic, and it starts by defining a set of perturbations of a regular configuration. Then, by analyzing the behavior of a generic distributed algorithm on all perturbations of a swarm that satisfy some initial conditions, we will show that the algorithm either (i) allows the construction of a rotationally asymmetric configuration that can evolve into a rotationally symmetric one (under a semi-synchronous scheduler) or (ii) leaves the configuration unchanged forever. In both cases, the algorithm does not solve the Gathering problem on some configurations.

Perturbations

For the rest of this section, we will denote by C the unit circle centered at the origin. A finite set $S \subset C$ is *regular* if $(1, 0) \in S$ and all points of S have the same angle sequence. Hence, for every positive integer n, there is a unique regular set of size n: for $n \geq 3$, this is the set of vertices of the regular n-gon centered at the origin and having a vertex in $(1, 0)$.

Let S be the regular set of size n, and let p_1, p_2, …, p_n be the points of S taken in clockwise order, starting from $p_1 = (1, 0)$. Let $\varepsilon \in \mathbb{R}$ with $0 < \varepsilon < 2\pi/n$, and let $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n) \in [0, 1]^n \subset \mathbb{R}^n$. The *$\varepsilon$-perturbation* of S with *coefficients* γ is the set

$S' \subset C$ of size n such that, for all $1 \leq i \leq n$, there is a (unique) point $p'_i \in S'$ with $cw(p_i, p'_i) = \gamma_i \cdot \varepsilon$, called the *perturbed copy* of p_i. So, any ε-perturbation of S is obtained by rotating each point of S clockwise around the origin by an angle in $[0, \varepsilon]$.

Furthermore, for $1 \leq i \leq n$, we say that two coefficient n-tuples $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)$ and $\gamma' = (\gamma'_1, \gamma'_2, \dots, \gamma'_n)$ are *i-related* if and only if they differ at most by their ith terms, i.e., for all $j \neq i$, we have $\gamma_j = \gamma'_j$. Note that the i-relation is an equivalence relation on $[0, 1]^n$. With the previous paragraph's notation, we say that the set of all ε-perturbations of S whose coefficients are in a same equivalence class of the i-relation is a *bundle* of ε-perturbations of p_i. Intuitively, a bundle of ε-perturbations of p_i is obtained by first fixing a perturbation of all points of S except p_i, and then perturbing p_i in all possible ways.

Size of the swarm

We will prove that Gathering is impossible for any given visibility range $\vartheta \leq \pi/2$, provided that the size of the swarm n is appropriate. Specifically, we say that a positive integer n is *compatible* with ϑ if three conditions hold on the regular set S of size n:

1. For every $p \in S$, the open semicircle centered at p contains exactly half of the points of S.
2. No two points of S have an angular distance of exactly ϑ.
3. There are two distinct points of S whose angular distance is smaller than ϑ.

We can show that there are arbitrarily large such integers:

▶ **Proposition 3.1.** *For any $\vartheta \leq \pi/2$, there are arbitrarily large integers compatible with ϑ.* ◀

Choice of ε

For every integer n compatible with ϑ, we define a positive number $\varepsilon_{\vartheta,n}$, which will be used to construct perturbations of the regular set S of size n. We set $\varepsilon_{\vartheta,n} = \delta/2$, where $\delta = \min\{|\vartheta - 2\pi a/n| \mid a \in \mathbb{N},\ 0 \leq a \leq n\}$.

Since n is compatible with ϑ, it easily follows that $\varepsilon_{\vartheta,n} > 0$. Also, δ is at most half the angular distance between two consecutive points of S, and therefore $\varepsilon_{\vartheta,n} \leq \pi/2n$. Moreover, our choice of $\varepsilon_{\vartheta,n}$ has some other desirable properties:

▶ **Proposition 3.2.** *Let n be an integer compatible with $\vartheta \leq \pi/2$, let S be the regular set of size n, and let S' be an $\varepsilon_{\vartheta,n}$-perturbation of S. If $p \in S$, and $p' \in S'$ is the perturbed copy of p, the following hold:*

1. *The ϑ-neighborhood of p contains a point $q \in S$ if and only if the ϑ-neighborhood of p' contains the perturbed copy of q in S'.*
2. *The open semicircle D centered at p contains exactly half of the points of S', which are the perturbed copies of the points of S contained in D.*
3. *If D' is the open semicircle centered at p', then $S' \cap D = S' \cap D'$, and hence D' contains exactly half of the points of S'.* ◀

▶ **Corollary 3.3.** *Let n be an integer compatible with $\vartheta \leq \pi/2$, and let S' be an $\varepsilon_{\vartheta,n}$-perturbation of the regular set S of size n. If two swarms of robots form S and S' respectively, their visibility graphs are isomorphic.* ◀

Combining configurations

Next we will describe a way of combining two configurations of robots into a new one that takes an open semicircle from each. This operation will be used to construct configurations of robots where a given distributed algorithm fails to make the robots gather.

Let S_1 and S_2 be two subsets of C, and let D be an open semicircle. The *D-combination* of S_1 and S_2 is defined as the set $(S_1 \cap D) \cup \rho(S_2 \cap D)$, where ρ is the rotation by π around the origin. In other words, this operation takes S_1, discards the points that do not lie in D, and replaces them with the points of S_2 that lie in D, by mapping them to their antipodal points.

Preliminary lemmas

We are now ready to give our first two lemmas, which deal with swarms forming perturbations of a regular configuration, and analyze the distributed algorithms that make robots move in some ways. The first lemma states that, if an algorithm makes a robot move to a point not currently occupied by another robot, then the algorithm cannot solve the Gathering problem.

▶ **Lemma 3.4.** *Let A be a distributed algorithm, let n be compatible with $\vartheta \leq \pi/2$, and consider a swarm of robots that forms an $\varepsilon_{\vartheta,n}$-perturbation S' of the regular set of size n. If there is a robot that, executing A, moves to a point not in S', then A does not solve the Gathering problem, even under the condition that the swarm initially forms a rotationally asymmetric set of n distinct points with a connected visibility graph.*

Proof (sketch). Assume that, if a robot located in $p' \in S'$ executes A, it moves to a point $q \notin S'$. Let $S'' = (S' \setminus \{p'\}) \cup \{q\}$, and let Q be the D-combination of S' and S'', where D is the open semicircle centered at p'. Consider a swarm initially forming Q, which is a rotationally asymmetric set of n distinct points with a connected visibility graph. If the scheduler activates only the robot in p', the configuration becomes rotationally symmetric. So, by Corollary 2.3, A is not a Gathering algorithm. ◀

The second lemma states that, if a distributed algorithm makes a robot r move on top of another robot r', and there is a perturbation of r such that the same algorithm makes r move on top of the same robot r', then the algorithm does not solve the Gathering problem.

▶ **Lemma 3.5.** *Let A be a distributed algorithm, let n be compatible with $\vartheta \leq \pi/2$, let S be the regular set of size n, and let S' and S'' be two distinct sets in the same bundle of $\varepsilon_{\vartheta,n}$-perturbations of $p \in S$, where $p' \in S'$ and $p'' \in S''$ are the perturbed copies of p. Assume that, if a swarm of robots forms S' and the robot in p' executes A, it moves to another robot, located in $q \in S'$. Also assume that, if a swarm of robots forms S'' and the robot in p'' executes A, it moves to the same point q. Then, A does not solve the Gathering problem, even under the condition that the swarm initially forms a rotationally asymmetric set of n distinct points with a connected visibility graph.*

Proof (sketch). Let D be the open semicircle centered at p, and let Q be the D-combination of S' and S''. Consider a swarm initially forming Q, which is a rotationally asymmetric set of n distinct points with a connected visibility graph. Suppose the scheduler activates only two robots: the one in p' and the one in the point antipodal to p''. After these two robots have executed A, the swarm's configuration becomes rotationally symmetric. ◀

Probabilistic argument

Our concluding argument goes as follows. Suppose for a contradiction that there is a Gathering algorithm A for some $\vartheta \le \pi/2$. Let n be an arbitrarily large integer compatible with ϑ, and let S be the regular set of size n. We will derive a contradiction by studying the behavior of A on the swarms forming the $\varepsilon_{\vartheta,n}$-perturbations of S. Specifically, let p_1, p_2, …, p_n be the points of S taken in clockwise order, starting from $p_1 = (1, 0)$. Suppose that a swarm of n robots forms some $\varepsilon_{\vartheta,n}$-perturbation of S, with robot r_i occupying the perturbed copy of p_i, and let all robots execute algorithm A.

Let us first restrict ourselves to a bundle P of $\varepsilon_{\vartheta,n}$-perturbations of some $p_i \in S$, and let us analyze the possible behaviors of the robot r_i. Recall that, by definition of bundle, the perturbations in P have fixed coefficients for all the points of S except p_i, and perturb p_i in every possible way, by varying the coefficient $\gamma_i \in [0, 1]$. Observe that, by Lemma 3.4, A should never make r_i move to some unoccupied location, or A would not be a Gathering algorithm. Also, if two or more perturbations in the bundle P made r_i move to the same robot, then A would not be a Gathering algorithm, due to Lemma 3.5. However, by the pigeonhole principle, if n perturbations in P made r_i move to some other robot, then at least two of them would make it move to the same robot. It follows that at most $n - 1$ perturbations in P can make r_i move at all. So, all perturbations in P except a finite number of them must make r_i stay still.

Now, let us pick an $\varepsilon_{\vartheta,n}$-perturbation of S by choosing its coefficients $\gamma \in [0, 1]^n$ uniformly at random. Let us also define n random variables $X_i \colon [0, 1]^n \to \{0, 1\}$, with $1 \le i \le n$, such that $X_i(\gamma) = 0$ if and only if algorithm A makes the robot r_i stay still when the swarm's configuration is the $\varepsilon_{\vartheta,n}$-perturbation of S defined by the coefficients γ. By the above argument, for every bundle P of $\varepsilon_{\vartheta,n}$-perturbations of p_i, we have $\Pr[X_i(\gamma) = 1 \mid \gamma \in P] = 0$. Then, integrating $X_i(\gamma)$ over $[0, 1]^n$, we obtain $\Pr[X_i = 1] = 0$.

Hence, the probability that A will make the robot r_i stay still when the swarm's configuration is picked at random among all $\varepsilon_{\vartheta,n}$-perturbations of S is 1. Since this is true of all robots separately, it is also true of all robots collectively, by the inclusion-exclusion principle. In other words, with probability 1, on a random $\varepsilon_{\vartheta,n}$-perturbation of S, no robot will be able to move, and therefore the robots will be unable to gather. Moreover, with probability 1, a random $\varepsilon_{\vartheta,n}$-perturbation of S is rotationally asymmetric. As a consequence, there is at least one initial configuration (actually, a great deal of configurations) where the swarm forms a rotationally asymmetric set of n distinct points with a connected visibility graph, and where no robot is able to move. We conclude that A cannot be a Gathering algorithm, even under such strong conditions.

Technical hindrances

The probabilistic proof we outlined above is sound for the most part, but unfortunately making it rigorous is a delicate matter. The problem is that, in order for X_i to be a random variable, it has to be a measurable function. For this to be true, the set of coefficients corresponding to perturbations where algorithm A makes the robot r_i stay still should be a measurable subset of $[0, 1]^n$. In turn, this requires some assumptions on the nature of A, whereas we only defined A as a generic function mapping a snapshot to a point.

However, since the function A actually implements an algorithm, which typically is a finite sequence of operations that are well-behaved in an analytic sense, most reasonable assumptions on A would rule out the pathological non-measurable cases, and would therefore make X_i a properly defined random variable, allowing the rest of the proof to go through.

Nonetheless, we choose to adopt a different approach, which is both less technical and more general in scope. Indeed, we will give a "derandomized" version of the above proof, which will not deal with probability spaces and random variables, and will not require a more restrictive re-definition of which functions are computable by mobile robots.

Derandomization

Next we will show how to complete the previous argument without the use of probability. Note that we do not need to prove that a random perturbation causes all robots to stay still with probability 1: we merely have to show that there is at least one perturbation with such a property. This is significantly easier, and is achieved by the next lemma, where X_i no longer denotes a random variable but simply a set of coefficients.

▶ **Lemma 3.6.** *Let $m, n \in \mathbb{N}^+$, and let $X_1, X_2, \ldots, X_n$ be subsets of the unit hypercube $[0,1]^n \subset \mathbb{R}^n$ such that every line parallel to the ith coordinate axis intersects X_i in less than m points, for all $1 \le i \le n$. Then, there is a point in $[0,1]^n$ whose n coordinates are all distinct that does not lie in any of the sets $X_1, X_2, \ldots, X_n$.* ◀

We can now prove the main result of this section.

▶ **Theorem 3.7.** *If $\vartheta \le \pi/2$, and for arbitrarily large n, there is no Gathering algorithm under the condition that the swarm initially forms a rotationally asymmetric set of n distinct points with a connected visibility graph.*

Proof. Let n be an arbitrarily large integer compatible with ϑ, which exists due to Proposition 3.1. Note that all $\varepsilon_{\vartheta,n}$-perturbations of the regular set of size S have a connected visibility graph, by Corollary 3.3. As before, we assume for a contradiction that A is a Gathering algorithm, and we consider a swarm of size n where all robots execute A, and each robot r_i is initially located in the perturbed copy of point $p_i \in S$, for some $\varepsilon_{\vartheta,n}$-perturbation of S.

For $1 \le i \le n$, let $X_i \subseteq [0,1]^n$ be the set of coefficients corresponding to perturbations where algorithm A causes r_i to make a non-null movement. As we already proved, due to Lemmas 3.4 and 3.5, in each bundle of $\varepsilon_{\vartheta,n}$-perturbations of p_i, at most $n-1$ perturbations cause r_i to move. Rephrased in geometric terms, every line in $\mathbb{R}^n$ parallel to the ith coordinate axis intersects X_i in less than n points.

So, the sets $X_1, X_2, \ldots, X_n$ satisfy the hypotheses of Lemma 3.6 with $m = n$. As a consequence, there exists $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n) \in [0,1]^n$, where the coefficients γ_i are all distinct, such that, in the perturbation corresponding to γ, algorithm A causes all robots to stay still, and therefore does not allow them to gather.

It remains to check that the perturbation S' corresponding to γ is rotationally asymmetric. Let $p_1, p_2, \ldots, p_n$ be the points of S in clockwise order, and let $p'_i \in S'$ be the perturbed copy of p_i, for $1 \le i \le n$. Suppose for a contradiction that S' has a k-fold rotational symmetry with $k > 1$, implying that the angular distance between p'_1 and $p'_{n/k+1}$ is $\alpha = 2\pi/k$. Note that α is also the angular distance between p_1 and $p_{n/k+1}$. Moreover, by definition of perturbation, $\alpha = cw(p_1, p_{n/k+1}) - cw(p_1, p'_1) + cw(p_{n/k+1}, p'_{n/k+1}) = 2\pi/k - \gamma_1 \cdot \varepsilon_{\vartheta,n} + \gamma_{n/k+1} \cdot \varepsilon_{\vartheta,n}$. It follows that $\gamma_1 = \gamma_{n/k+1}$, which contradicts the fact that the coefficients γ_i are all distinct (indeed, $k \ge 2$ implies that $1 < n/k + 1 \le n$). ◀

We remark that, throughout the proofs of Lemmas 3.4 and 3.5 and Theorem 3.7, only swarms of the same size n appear, and so our impossibility result holds even when n is fixed. It follows that Gathering is impossible even if all robots know the size of the swarm.

4 Gathering algorithm for $\vartheta = \pi$

Overview

In this section we give a Gathering algorithm for robots that can see the entire circle except their antipodal point (i.e., $\vartheta = \pi$), under the condition that the initial configuration is a rotationally asymmetric set with no multiplicity points.

First we will describe a simple Gathering algorithm for robots with full visibility, which already provides some useful ideas: elect a leader, form a unique multiplicity point, and gather there. We will then extend the same ideas to the limited-visibility case with $\vartheta = \pi$. There are some difficulties arising from the fact that not all robots will necessarily agree on the same leader, because they may have different views of the rest of the swarm. For instance, two antipodal robots will not see each other, and, if the configuration is rotationally asymmetric, they will get two non-isometric snapshots, which perhaps will cause them to elect two different leaders.

We will show how to cope with these difficulties. Essentially, based on what a robot r knows, there are only two possibilities on who the "true" leader may be, depending on whether there is a robot antipodal to r or not. If r happens to be elected leader in both scenarios, then r has no doubt of being the leader, and therefore behaves as in the full-visibility algorithm, creating a multiplicity point. In most cases, however, no robot will be so fortunate, but the swarm will still have to make some sort of progress toward gathering. So, the robots that see themselves as possible leaders (but could be wrong) make some preparatory moves that will ideally "strengthen their leadership" in the next turns. We will argue that, after a finite number of turns, one robot will become aware of being the leader and will create a multiplicity point, even under a semi-synchronous scheduler.

The design and analysis of our Gathering algorithm are further complicated by some undesirable special cases, where two distinct multiplicity points end up being created, or the multiplicity point is antipodal to some robot, and therefore invisible to it.

Full visibility and leader election

We will describe a simple Gathering algorithm for the scenario where robots have full visibility.

Let S be a rotationally asymmetric finite set of points on a circle. Recall from Proposition 2.1 that all points of S have distinct angle sequences, and therefore there is a unique point $p \in S$ with the lexicographically smallest angle sequence: p is called the *head* of S.

The Gathering algorithm uses the fact that all robots agree on where the head of the swarm is, and the robot located at the head is elected the leader. The algorithm makes the leader move clockwise to the next robot, while all other robots wait. As soon as there is a multiplicity point, the closest robot in the clockwise direction moves counterclockwise to the multiplicity point. The process continues until all robots have gathered.

Note that this algorithm also works in ASYNCH and with non-rigid robots (i.e., robots that can be stopped by an adversary before reaching their destination). Indeed, as the leader moves toward the next robot, its angle sequence remains the lexicographically smallest, and so it remains the leader. After a multiplicity point has been created, only one robot is allowed to move at any time, and therefore no other multiplicity points are accidentally formed.

Undecided leaders and cognizant leader

Let us now consider a swarm of robots with visibility range $\vartheta = \pi$ forming a rotationally asymmetric set S of n distinct points. We say that the *true leader* of the swarm is the robot located at the head of S.

For each robot r, we define the *visible configuration* $V(r)$ as the set of robots that are visible to r, and the *ghost configuration* as $G(r) = V(r) \cup \{r'\}$, with r' antipodal to r. Note that exactly one between $V(r)$ and $G(r)$ is isometric to the "real" configuration S, and therefore at least one between $V(r)$ and $G(r)$ is rotationally asymmetric.

The *visible head* $v(r)$ is defined as follows: if $V(r)$ is rotationally asymmetric, $v(r)$ is the head of $V(r)$; otherwise, $v(r)$ is the head of $G(r)$. The *ghost head* $g(r)$ is defined similarly: if $G(r)$ is rotationally asymmetric, $g(r)$ is the head of $G(r)$; otherwise, $g(r)$ is the head of $V(r)$. Note that the true leader of the swarm must be $v(r)$ or $g(r)$.

If $v(r) \neq g(r)$, and either $r = v(r)$ or $r = g(r)$, then r is said to be an *undecided leader*: a robot that is possibly the true leader, but does not know for sure. If $r = v(r) = g(r)$, then r is a *cognizant leader*: a robot that is aware of being the true leader of the swarm.

► **Proposition 4.1.** *In a rotationally asymmetric swarm with no multiplicity points, the true leader is either an undecided or a cognizant leader, and no robot other than the true leader can be a cognizant leader.* ◄

Point-addition lemma

The following lemma has important implications for the design of our algorithm.

► **Lemma 4.2.** *Let S be a finite non-empty set of points on a circle C, and let $S' = S \cup \{p\}$, where $p \in C \setminus S$. Assume that S and S' are rotationally asymmetric, and let $h \in S$ be the head of S and $h' \in S'$ be the head of S'. Then, either $h = h'$ or $cw(h, p) > 2 \cdot cw(h', p)$.* ◄

► **Corollary 4.3.** *In a rotationally asymmetric swarm with no multiplicity points, a robot r is a cognizant leader if and only if $r = g(r)$.* ◄

Gathering algorithm

Our Gathering algorithm for $\vartheta = \pi$ is illustrated in Listing 1. A robot r executing the algorithm first checks if the current configuration falls under some special cases (which will be discussed later), and then it attempts to determine the true leader of the swarm. By Corollary 4.3, checking if $r = g(r)$ is equivalent to checking if r is a cognizant leader. In this case, by Proposition 4.1, r is the true leader, and hence it behaves like in the full-visibility algorithm: it moves clockwise to the next robot, s (rule 3: see Figure 1a).

If r is not a cognizant leader, it checks if it is at least an undecided leader: $r = v(r)$. In this case, r cannot commit itself to moving to s, because several robots may be undecided leaders, and this would create more than one multiplicity point. Instead, r attempts to "strengthen its leadership" by moving halfway toward s (rule 4.c: see Figure 1d): this ensures that, in the next turn, r will have a lexicographically smaller angle sequence than it currently has (unless, of course, s moves as well).

Another goal of r is to be able to see the entire swarm in the next turns. Therefore, if the midpoint of r and s happens to be antipodal to some robot q, then r moves a bit further past the midpoint (rule 4.b: see Figure 1c). This way, r will be sure to see q in the next turn (unless, of course, q moves as well).

An exception to the above is when $g(r)$ is antipodal to r, and s has an antipodal robot s'. In this situation, if r had an antipodal robot r', then r' would be the true leader, which would then either form a multiplicity point with s' (if r' is activated) or would become visible to r (if r is activated but not r'). However, if r' does not exist, then r is the true leader, but r may never find out: it may keep approaching s without ever reaching it, and there

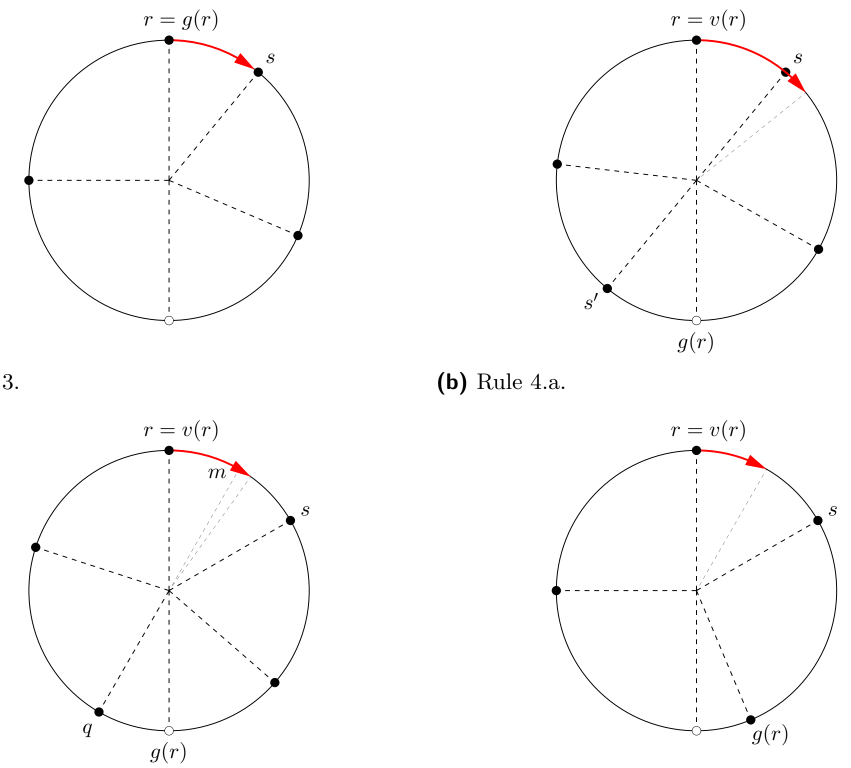
Listing 1 Gathering algorithm for $\vartheta = \pi$.

```
The algorithm is executed by a generic robot r.
Input: V(r), the set of points occupied by robots visible to r
       (expressed in r's coordinate system), with weak multiplicity.
Output: a destination point for r.

Let s ∈ V(r) be such that cw(r,s) > 0 is minimum, if it exists
(s is the visible robot closest to r in the clockwise direction).

Let V'(r) be the set of all the points in V(r) (without multiplicity)
plus their antipodal points.

Let δ be the smallest cw(a,b) > 0 with a,b ∈ V'(r).

 1. If r sees some multiplicity points, then:
   1.a. If r sees a unique multiplicity point, then r moves to it.
   1.b. If r sees two multiplicity points a and b with cw(a,b) > cw(b,a),
        then r moves to a.
 2. Else, if r sees no other robots, then r moves clockwise by π/2.
 3. Else, if r = g(r), then r moves to s.
 4. Else, if r = v(r), then:
   4.a. If g(r) is antipodal to r, and s has an antipodal robot,
        then r moves clockwise by cw(r,s) + δ/3.
   4.b. Else, if there is a point m ∈ V'(r) such that cw(r,m) = cw(r,s)/2,
        then r moves clockwise by cw(r,s)/2 + δ/7.
   4.c. Else, r moves clockwise by cw(r,s)/2.
 5. Else, r does not move.
```

may always be a ghost head antipodal to r. For this reason, if r detects this configuration, it moves slightly past s (rule 4.a: see Figure 1b): this way, s will be the new leader, and it will not be in the same undesirable configuration, because r will not have an antipodal robot.

Note that, when describing rule 4.a and rule 4.b, we mentioned some undefined "small" distances. According to Listing 1, these are respectively $\delta/3$ and $\delta/7$. In turn, δ is defined as the smallest angular distance between two points in $V'(r)$, where $V'(r)$ is the set of all the points in $V(r)$ plus their antipodal points. It is easy to see that all robots that are activated at the same time compute isometric sets V', and therefore they implicitly agree on the value of δ. The reason why the specific values $\delta/3$ and $\delta/7$ have been chosen will become apparent in the proof of correctness of the algorithm *(refer to the full version for details)*.

Finally, let us discuss the special cases, all of which arise when some multiplicity points have been created (due to a cognizant leader moving to some other robot). If only one multiplicity point is visible to r, then r simply moves to it, as in the full-visibility algorithm (rule 1.a). In some exceptional circumstances, two multiplicity points a and b may be created, but we will prove that a and b will not be antipodal to each other, and there will never be a third multiplicity point. In this case, there is an implicit order between a and b on which all robots agree, and so they will all move to the same multiplicity point, say a (rule 1.b). The last special case is when all robots have gathered in a point, except a single robot r located in the antipodal point. r detects this situation because it sees no robots other than itself (and its current location is not a multiplicity point). So, r just moves to another visible point, say, the one forming a clockwise angle of $\pi/2$ with r (rule 2). This ensures that r will see the multiplicity point on its next turn.



(a) Rule 3. **(b)** Rule 4.a. **(c)** Rule 4.b. **(d)** Rule 4.c.

Figure 1 Examples of some of the rules of the Gathering algorithm in Listing 1. Black dots indicate robots that are visible to r. A white dot indicates the point antipodal to r, which may or may not be occupied by a robot.

Correctness

In the following, we will assume that all robots in a swarm of size $n > 1$ execute the algorithm in Listing 1 starting from a rotationally asymmetric initial configuration with no multiplicity points. We will prove that, no matter how the adversarial semi-synchronous scheduler activates them, all robots will eventually gather in a point and no longer move.

We say that, in a given configuration S, a robot r *is able to apply rule* j if, assuming that r is activated when the swarm forms S, r executes rule j (and no other rule).

▶ **Lemma 4.4.** *Assume that the swarm forms a rotationally asymmetric configuration with no multiplicity points, and let ℓ be the true leader. Then:*

1. *No robot is able to apply rule 1 or rule 2.*
2. *At most one robot is able to apply rule 3: the true leader ℓ.*
3. *At most one robot is able to apply rule 4.a: the true leader ℓ (if there is no robot antipodal to ℓ) or the robot antipodal to ℓ.*
4. *A robot $r \neq \ell$ is able to apply rule 4 only if $\pi/2 < cw(r, \ell) \leq \pi$, and only if there is a robot antipodal to r.* ◀

Due to Lemma 4.4, if the swarm forms a rotationally asymmetric configuration with no multiplicity points, we say that a robot *is able to move* if it is able to apply rule 3 or rule 4: indeed, these are the only rules that result in a non-null movement.

▶ **Lemma 4.5.** *In any rotationally asymmetric configuration with no multiplicity points, at most two robots are able to move, and the true leader is always able to move.* ◀

▶ **Lemma 4.6.** *If the swarm has a unique multiplicity point, then all robots eventually gather and no longer move.* ◀

▶ **Lemma 4.7.** *Assume that the swarm forms a rotationally asymmetric configuration with no multiplicity points, and let r and r' be two robots that are able to move. Then:*

1. *If r is activated and executes rule 3, and r' is not activated, then r does not move to r'.*
2. *If r is activated and executes rule 4, then it moves by an angular distance strictly smaller than π, and its destination point is not in $V'(r)$, as defined in Listing 1.*
3. *If both r and r' are activated and execute rule 4.b or rule 4.c, then r and r' are not antipodal of each other, and their destination points are not antipodal of each other.* ◀

▶ **Lemma 4.8.** *If the swarm forms a rotationally asymmetric configuration with no multiplicity points and an active robot executes rule 3 or rule 4.a, then all robots eventually gather in a point and no longer move.* ◀

▶ **Lemma 4.9.** *Assume that, at time t, the swarm forms a rotationally asymmetric configuration with no multiplicity points, and all the robots that move at time t execute rule 4.b or rule 4.c. Then, at time $t+1$, the swarm still forms a rotationally asymmetric configuration with no multiplicity points.* ◀

▶ **Lemma 4.10.** *Assume that, at time t, the swarm forms a rotationally asymmetric configuration with no multiplicity points. If, at all times $t' \geq t$, no robot other than the true leader moves, then all robots eventually gather in a point and no longer move.* ◀

We are now ready to prove the main result of this section.

▶ **Theorem 4.11.** *If $\vartheta = \pi$, there is a Gathering algorithm under the condition that the swarm initially forms a rotationally asymmetric configuration with no multiplicity points.*

Proof. We will prove that the distributed algorithm in Listing 1 solves the Gathering problem under the condition that the swarm initially forms a rotationally asymmetric configuration with no multiplicity points. By the first statement of Lemma 4.4, whenever the swarm forms a rotationally asymmetric configuration with no multiplicity points, any robot that is activated and moves executes either rule 3, or rule 4.a, or rule 4.b, or rule 4.c. In the first two cases, we conclude by Lemma 4.8. In the latter two cases, by Lemma 4.9, the resulting configuration is still rotationally asymmetric and with no multiplicity points. By inductively repeating this argument, we may assume, without loss of generality, that the swarm forms a rotationally asymmetric configuration with no multiplicity points at all times, and all robots that are activated and move execute rule 4.b or rule 4.c.

By Lemma 4.5, at a generic time t, at least one robot r and at most one robot $r' \neq r$ are allowed to move. The semi-synchronous scheduler will activate each of them infinitely often, so let $t' \geq t$ be the first time this happens. Assume that one robot, say r', is not activated at time t', and therefore r is. Then, r does not have an antipodal robot at time $t'+1$, due to the second statement of Lemma 4.7. Similarly, if both r and r' are activated at time t', none of them has an antipodal robot at time $t'+1$, by the second and third statements of Lemma 4.7.

In summary, if a robot is activated and moves at a generic time t, it no longer has antipodal robots at any time after t. Since the robots are finitely many, eventually, say after time t'', only robots without an antipodal robot will move. However, by the fourth statement of Lemma 4.4, a robot that moves must have an antipodal robot, unless it is the current true leader. So, at all times after t'', no robot other than the current true leader will move. Therefore, Lemma 4.10 allows us to conclude. ◀

References

1 N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006.

2 H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–838, 1999.

3 S. Bhagat, K. Mukhopadhyaya, and S. Mukhopadhyaya. Computation under restricted visibility. In *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 134–183. Springer, 2019.

4 M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: gathering. *SIAM Journal on Computing*, 41(2):829–879, 2012.

5 P. Courtieu, L. Rieg, S. Tixeuil, and X. Urbain. Impossibility of gathering, a certification. *Information Processing Letters*, 115(3):447–452, 2015.

6 G. D'Angelo, G. Di Stefano, and A. Navarra. Gathering on rings under the Look–Compute–Move model. *Distributed Computing*, 27(4):255–285, 2014.

7 G. D'Angelo, A. Navarra, and N. Nisse. A unified approach for gathering and exclusive searching on rings under weak assumptions. *Distributed Computing*, 30(1):17–48, 2017.

8 S. Das, G.A. Di Luna, P. Flocchini, N. Santoro, G. Viglietta, and M. Yamashita. Oblivious permutations on the plane. In *23rd International Conference on Principles of Distributed Systems*, pages 24:1–24:16, 2019.

9 X. Défago, M. Gradinariu, S. Messika, P. Raipin-Parvédy, and S. Dolev. Fault-tolerant and self-stabilizing mobile robots gathering. In *20th International Symposium on Distributed Computing*, pages 46–60, 2006.

10 B. Degener, B. Kempkes, P. Kling, F. Meyer auf der Heide, P. Pietrzyk, and R. Wattenhofer. A tight runtime bound for synchronous gathering of autonomous robots with limited visibility. In *23rd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 139–148, 2011.

11 G.A. Di Luna, P. Flocchini, L. Pagli, G. Prencipe, N. Santoro, and G. Viglietta. Gathering in dynamic rings. *Theoretical Computer Science*, 811:79–98, 2020.

12 G.A. Di Luna, P. Flocchini, N. Santoro, and G. Viglietta. TuringMobile: a turing machine of oblivious mobile robots with limited visibility and its applications. In *32nd International Symposium on Distributed Computing*, pages 19:1–19:18, 2018.

13 G.A. Di Luna, P. Flocchini, N. Santoro, G. Viglietta, and M. Yamashita. Meeting in a polygon by anonymous oblivious robots. *Distributed Computing (to appear)*, 2019.

14 G.A. Di Luna and G. Viglietta. Robots with lights. In *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, pages 252–277. Springer, 2019.

15 P. Flocchini. Gathering. In *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 63–82. Springer, 2019.

16 P. Flocchini, G. Prencipe, and N. Santoro. Self-deployment of mobile sensors on a ring. *Theoretical Computer Science*, 402(1):67–80, 2008.

17 P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots*. Morgan & Claypool, 2012.

18 P. Flocchini, G. Prencipe, and N. Santoro, editors. *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*. Springer, 2019.

19 P. Flocchini, G. Prencipe, N. Santoro, and G. Viglietta. Distributed computing by mobile robots: uniform circle formation. *Distributed Computing*, 30(6):413–457, 2017.

20 P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1–3):147–168, 2005.

21 N. Fujinaga, Y. Yamauchi, S. Kijima, and M. Yamahista. Pattern formation by oblivious asynchronous mobile robots. *SIAM Journal on Computing*, 44(3):740–785, 2015.

22 S. Kamei, A. Lamani, F. Ooshita, S. Tixeuil, and K. Wada. Gathering on rings for myopic asynchronous robots with lights. In *23rd International Conference on Principles of Distributed Systems*, pages 27:1–27:17, 2019.

23 E. Kranakis, D. Krizanc, and E. Markou. *The Mobile Agent Rendezvous Problem in the Ring*. Morgan and Claypool, 2010.

24 A. Monde, Y. Yamauchi, S. Kijima, and M. Yamashita. Self-stabilizing localization of the middle point of a line segment by an oblivious robot with limited visibility. In *19th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 172–186, 2017.

25 L. Pagli, G. Prencipe, and G. Viglietta. Getting close without touching: near-gathering for autonomous mobile robots. *Distributed Computing*, 28(5):333–349, 2015.

26 P. Poudel and G. Sharma. Universally optimal gathering under limited visibility. In *19th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 323–340, 2017.

27 S. Souissi, X. Défago, and M. Yamashita. Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. *ACM Transactions on Autonomous and Adaptive Systems*, 4(1):9:1–9:27, 2009.

28 I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.

29 Y. Yamauchi. Symmetry of anonymous robots. In *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, pages 109–133. Springer, 2019.

30 Y. Yamauchi, T. Uehara, S. Kijima, and M. Yamashita. Plane formation by synchronous mobile robots in the three-dimensional euclidean space. *Journal of the ACM*, 64(3):16:1–16:43, 2017.

31 Y. Yamauchi and M. Yamashita. Pattern formation by mobile robots with limited visibility. In *20th International Colloquium on Structural Information and Communication Complexity*, pages 201–212, 2013.