

Title	Non Uniform Circle Formation Algorithm for Oblivious Mobile Robots with Convergence toward Uniformity
Author(s)	Defago, Xavier; Souissi, Samia
Citation	Theoretical Computer Science, 396(1-3): 97-112
Issue Date	2008-05-10
Type	Journal Article
Text version	author
URL	<a href="http://hdl.handle.net/10119/4905">http://hdl.handle.net/10119/4905</a>
Rights	<p>NOTICE: This is the author ' s version of a work accepted for publication by Elsevier. Changes resulting from the publishing process, including peer review, editing, corrections, structural formatting and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication.</p> <p>A definitive version was subsequently published in Xavier Defago and Samia Souissi, Theoretical Computer Science, 396(1-3), 2008, 97-112, <a href="http://dx.doi.org/10.1016/j.tcs.2008.01.050">http://dx.doi.org/10.1016/j.tcs.2008.01.050</a></p>
Description	

# Non Uniform Circle Formation Algorithm for Oblivious Mobile Robots with Convergence toward Uniformity <sup>★</sup>

Xavier Défago, Samia Souissi

*School of Information Science,  
Japan Advanced Institute of Science and Technology,  
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan.*

---

## Abstract

This paper presents a distributed algorithm whereby a group of mobile robots self-organize and position themselves into forming a circle in a loosely synchronized environment. In spite of its apparent simplicity, the difficulty of the problem comes from the weak assumptions made on the system. In particular, robots are anonymous, oblivious (i.e., stateless), unable to communicate directly, and disoriented in the sense that they share no knowledge of a common coordinate system. Furthermore, robots' activations are not synchronized. More specifically, the proposed algorithm ensures that robots deterministically form a non uniform circle in a finite number of steps and converges to a situation in which all robots are located evenly on the boundary of the circle.

*Key words:* Mobile Robots, Distributed Algorithms, Circle Formation, Cooperation and Control, Distributed Computing, Theory.

---

---

<sup>★</sup> A preliminary version of this paper, albeit with a less elegant algorithm, was presented at the 2nd ACM Annual Workshop on Principles of Mobile Computing [6]. The newer algorithm presented in this paper was developed in Souissi's master thesis research [13].

*Email addresses:* defago@jaist.ac.jp (Xavier Défago),  
ssouissi@jaist.ac.jp (Samia Souissi).

<sup>1</sup> phone: +81-761-51-1254, fax: +81-761-51-1149.

## 1 Introduction

Mobile computing systems, devices, and applications are gradually becoming more and more pervasive, while the theoretical foundations still remain to be established. Current research on principles of mobile computing mostly aims at systems in which mobility occurs as an external factor, such as in mobile ad hoc networks, mobile information systems, ubiquitous computing, or sensor networks. In contrast, we focus on systems for which the mobility must be controlled, such as groups of mobile robots. In particular, we look at basic algorithms for coordinating the movements of such robots.

This paper presents a distributed algorithm whereby a group of weak mobile robots, sharing no common coordinate system, can self-organize into forming a circle when starting from any arbitrary configuration. Among other things, the ability to form a circle means that the robots are spontaneously able to reach an agreement on an origin and unit distance, albeit not on a complete coordinate system. Besides, the proposed algorithm has the useful property that it allows robots to be added, removed, or relocated during its execution. A circle is guaranteed to be reformed and remain stable after external changes have come to an end.

**Model and problem.** The robots considered in this paper are modelled as points that move on the plane. The robots have no identity, no memory of past actions, no common sense of direction and distance. Besides, robots execute the same deterministic algorithm, are unable to communicate directly, and can only interact by observing each others position.

In this model, we address the problem of forming a circle by a group of mobile robots, for which we give an oblivious algorithm. This problem in particular has interesting applications. For instance, consider the context of space exploration and the initial preparation of a zone. A group of robots could be sent and after landing at random locations, would self-organize to form the initial infrastructure for later expeditions. Besides, pattern formation problems in general provide a first step toward flocking, i.e., allowing a group of robots to move in formation [10]. Also, the formation of geometrical patterns and flocking are both useful in themselves, for instance, for the self-positioning of mobile base stations in a mobile ad hoc network, e.g., as considered by Chatziagiannakis et al. [4], and for the self-deployment of sensors on a network ring [31].

**Contribution.** The paper decomposes the question of circle formation into two parts: (1) forming a circle (possibly an irregular one), and (2) positioning

the robots evenly along the boundary of the circle i.e., the robots form a regular  $n$ -gon, where  $n$  is the number of robots.

Défago and Konagaya [6], in a preliminary version of this paper, proposed in the Suzuki and Yamashita model [15] an algorithm, which is a composition of two independent algorithms whereby oblivious robots deterministically form a circle, and converge to a situation in which all robots are arranged uniformly on its boundary. Unfortunately, that algorithm was unnecessarily complex and the proofs were only sketched in the original paper. In the meantime, we have developed a simpler and more elegant algorithm that should supersede the earlier one, together with complete and rigorous proofs of correctness, rather than the proof sketches of the previous version. Consequently, the main contribution of this paper is to propose an oblivious algorithm by which robots deterministically form a circle (part 1) in a finite number of steps, and asymptotically converge toward a situation in which they are positioned at regular intervals on the boundary of this circle (part 2). Our algorithm, elegantly solves both problems (part 1 and part 2) using a single algorithm. In addition, considering oblivious robots makes the proposed algorithm very robust in that it can tolerate additions, removals and relocations of any of the robots.

**Structure of the paper.** The rest of the paper is structured as follows. In Section 2, we survey the relevant literature on circle formation. In Section 3, we introduce the system model and the terminology used in the paper. In Section 4, we describe our algorithm, and in Section 5, we prove its correctness. In Section 6, we give a discussion on an earlier instance of the algorithm. Finally, in Section 7, we conclude the paper.

## 2 Related work

A vast amount of researches exists in the context of cooperative mobile robotics (see [23] for a slightly outdated survey but nevertheless insightful), but much research focuses on the study of diverse heuristics, for instance by studying how a complex global behavior can emerge from the interactions of many robots exhibiting a simple local behavior, such as free market optimization (e.g., [7]) or swarm intelligence (e.g., [2,12,25,32]). However, only few studies take the problem from a computational standpoint [15,22,24]. This can be partly explained by the difficulty of the task, and the fact that heuristics are perceived as a way to circumvent that difficulty.

Suzuki and Yamashita [15] first proposed a computational model in which mobile robots algorithms could be expressed and studied rigorously. This model is called *semi-synchronous*, and this is the model assumed in this paper.

Prencipe [28] proposed a fully asynchronous (hence weaker) variant of the model, called the *asynchronous* or CORDA. Prencipe [27] compared the two models with respect to the possibility of solving certain basic tasks for robots. Typical problems that have been studied in this perspective are the arbitrary pattern formation problem, where robots are asked to form a pattern given in input in finite time [15,22]; the gathering problem, where robots are asked to gather at some point, not determined in advance [26,29,30,33]; the flocking problem, where robots are required to keep a formation while moving [10]; and the circle formation problem which has also attracted considerable attention.

Flocchini et al. [22] studied the problem of arbitrary pattern formation under several assumptions in the asynchronous model in an oblivious setting. In particular, they showed that when robots share the knowledge of the direction and orientation of both  $x$  and  $y$  axes, the pattern formation problem can be solved. In contrast, if no axis direction is known by the robots, the general problem cannot be solved in that model. They also studied the case when robots know the direction and orientation of one single axis, and they showed that the pattern formation can be formed whenever the number of robots is *odd*. In later work [24], they have considered the case of *even* number of robots, and have shown that there exists no pattern formation algorithm that lets the robots form a symmetric pattern that has all its axes of symmetry passing through a vertex. Among other things, this means that the uniform circle formation is achievable, provided that robots share the direction and orientation of one axis.

Debest [5] briefly discussed the formation of a circle by a group of mobile robots as an illustration of self-stabilizing distributed algorithms. He discussed the problem, but did not provide an algorithm.

Sugihara and Suzuki [14] proposed several algorithms for the formation of geometrical patterns. In particular, they proposed a simple heuristic algorithm for the formation of an approximation of a circle in the limited visibility setting (i.e., a robot can see only part of the robots on the system). Their solution does not always reach a desirable configuration, and sometimes may bring the robots to form a Reuleaux triangle<sup>2</sup> instead of a circle. Later, Suzuki and Yamashita [15] proposed a *non-oblivious* algorithm for the formation of a regular polygon. To achieve this, robots must be able to remember all past actions and observations. The existence of an *oblivious* solution was however left as an open question.

Défago and Konagaya [6], in an earlier instance of this paper, proposed an algorithm by which a group of oblivious robots eventually form a circle in the Suzuki and Yamashita model [15]. The algorithm is a composition of two

---

<sup>2</sup> A Reuleaux triangle is a curve of constant width constructed by drawing arcs from each polygon vertex of an equilateral triangle between the other two vertices [16].

algorithms. The first one allows the robots to form a circle in finite time, and the second algorithm converges the robots toward a situation wherein all of them are arranged evenly on its boundary.

Later on, Chatzigiannakis et al. [3] proposed a partial solution to the circle formation problem in the semi-synchronous model that tried to simplify the algorithm of Défago and Konagaya [6]. Unfortunately, their solution relies on a simplifying assumption that completely removes the difficulty of the problem (in particular robots must not be located on the same radius).

Katreniak [11] proposed in the asynchronous model CORDA [28] a deterministic algorithm that solves a slightly different problem, called biangular circle.<sup>3</sup> In other words, their algorithm allows the robots to rearrange the circle to a symmetric configuration; biangular circle in finite time when the number of robots is *even*. Also, when the number of robots is *odd*, the robots achieve the uniform circle.

Dieudonné et al. [18] build upon the work of Katreniak [11], and extend it for the case with an *even* number of robots. In particular, they proposed an oblivious algorithm that solves the uniform circle formation problem in the semi-synchronous model combined with the solution of Katreniak [11]. More specifically, their algorithm solves the problem in finite time for any number  $n$  of robots, except when  $n = 4, 6$  and  $8$ . Besides, robots are assumed to teleport precisely to their computed destination without stopping on the way. This assumption was lifted in later work [20] by proposing a deterministic algorithm that works for any number of robots, except if  $n = 4$ , and no robot is required to reach its computed destination in one cycle.

The combined results of Katreniak [11] and Dieudonné and Petit [20] are very significant in that it provides an almost complete deterministic solution to the uniform circle formation, where ours only provides an asymptotic solution. However, our solution does not exclude any case (such as  $n \neq 4$ ) and as discussed later, has a lower complexity.

In a different study, Dieudonné and Petit [19] proposed an oblivious algorithm to solve the uniform circle formation problem for a *prime* number of robots in the semi-synchronous model.

---

<sup>3</sup> In a biangular circle, there is a center and two nonzero angles  $\alpha$  and  $\beta$  such that the center between each two adjacent points is either  $\alpha$  or  $\beta$ , and these angles alternate.

### 3 System Model and Definitions

#### 3.1 System Model

In this paper, we consider the system model of Suzuki and Yamashita [15], which is defined as follows. The system consists of a set of autonomous mobile robots  $\mathcal{R} = \{r_1, \dots, r_n\}$  roaming on the two-dimensional plane devoid of any landmark. Each robot is modelled and viewed as a point in the plane and equipped with sensors to observe the positions of the other robots. In particular, each robot proceeds by repeatedly observing the environment, performing computations based on the observed positions of robots, and moving toward the computed destination.

Each robot uses its own local  $x$ - $y$  coordinate system which includes an origin, a unit distance, and the directions of the two  $x$  and  $y$  axes, together with their orientations. The robots share neither knowledge of the coordinate systems of the other robots nor of a global one. However, robots agree on the chirality of the system (i.e., clockwise/counterclockwise orientation).

During its observation, a robot obtains the position of all robots according to its own local coordinate system. We assume that the robots have full visibility of each other and also do not obstruct the view from each other.

In the model, it is assumed that two robots can possibly occupy the same location. This assumption is undesirable for the formation of a circle because the robots may become impossible to separate later.<sup>4</sup> Thus, we assume that all robots occupy distinct locations initially, and let the algorithm ensure that it remains so.

The time is represented in the model as an infinite sequence of discrete time instants  $t_0, t_1, \dots, t_n$ , during which each robot can be either *active* or *inactive*. When a robot becomes active, it observes the environment, computes a new location, and moves toward it. This behavior constitutes its cycle of observing, computing, moving and being inactive. The sequence *look-compute-move* is called the *cycle* of a robot. The model assumes that activations (look, compute, move) occur instantaneously, resulting in a form of implicit synchronization. The model is called *semi-synchronous* model for this reason.

The activation of robots is determined by an activation schedule, unpredictable

---

<sup>4</sup> Consider two robots that happen to have the same coordinate system and that are always activated together. It is impossible to separate them deterministically. In contrast, it would be trivial to scatter them at distinct positions using randomization (e.g., [21]), but this is ruled out in our model.

and unknown to the robots. At each time instant a subset of the robots become active, with the guarantees that: (1) every robot becomes active at infinitely many time instants, (2) at least one robot is active during each time instant,<sup>5</sup> and (3) the time between two consecutive activations is not infinite.

In every single activation, a robot  $r_i$  can travel at most by a distance  $\delta_{r_i} > 0$ . This distance may be different between two robots. We sometimes say that  $r_i$  *moves toward* a point  $p$ . This means that  $r_i$  moves to location  $p$  if  $p$  is within  $\delta_{r_i}$  from  $r_i$ , or as close as possible to  $p$  otherwise.

Robots are anonymous in the sense that they are unable to uniquely identify themselves, neither with a unique identification number nor with some external distinctive mark (e.g., color, flag). Besides, all robots execute the same deterministic algorithm,<sup>6</sup> and thus have no way to generate a unique identity for themselves. Moreover, there is no explicit direct means of communication between robots. The communication occurs in a totally implicit manner; the only way for robots to acquire information is by observing each other's positions.

In this model, the algorithm consists of a deterministic function  $\varphi$  that is executed by every robot  $r_i$  each time it becomes active. The arguments of  $\varphi$  consist of the current position of the robot, and a set of points containing the observed position of all robots at the corresponding time instant. All positions are expressed in terms of the local coordinate system of  $r_i$ . The value returned by  $\varphi$  is the new destination for  $r_i$ .

### 3.2 Problem Definition

The problem addressed in this paper is the formation of a circle by a set of autonomous mobile robots. More rigorously, the problem is defined as follows.

**Problem 3.1 (UNIFORM CIRCLE FORMATION)** *Given a group of  $n$  robots  $r_1, r_2, \dots, r_n$  with distinct positions and located arbitrarily on the plane, eventually arrange them at regular intervals on the boundary of some non-degenerate circle (i.e., with finite radius greater than zero).*

We also consider a weaker problem that requires the robots to form a circle, but not necessarily be at regular intervals. This weaker problem is expressed

---

<sup>5</sup> As the duration of the interval between two time instants is by no means fixed, the second condition incurs no loss of generality. It is in fact only required for convenience.

<sup>6</sup> By deterministic, we mean that any two independent executions of the algorithm with identical input values always yield the same output.



more rigorously as follows.

**Problem 3.2 (CIRCLE FORMATION)** *Given a group of  $n$  robots  $r_1, r_2, \dots, r_n$  with distinct positions and located arbitrarily on the plane, arrange them to eventually form a non-degenerate circle.*

In terms of reaching agreement, it must be obvious that the weaker problem also provides an origin and a unit distance. At the same time, while it is conjectured that Problem 3.1 cannot be solved deterministically with oblivious robots, we show that Problem 3.2 can. In fact, we show that our algorithm solves Problem 3.2 within a finite number of steps, and converges toward a uniform solution (Prob. 3.1).

### 3.3 Notations

**Smallest enclosing circle** The *smallest enclosing circle* of a set of points  $P$  is denoted by  $\mathcal{C}$ , and its center is called  $o$ . It can be defined by either two opposite points, or by at least three points. The smallest enclosing circle is unique, and can be computed in  $O(n)$  time [17]. We shall denote by  $R$ , the radius of  $\mathcal{C}$ .

**Position** Given a robot  $r_i$ ,  $r_i(t)$  denotes its position at time  $t$ , according to some global  $x$ - $y$  coordinate system, and  $r_i(0)$  is its initial position.  $P(t) = \{r_i(t) | 1 \leq i \leq n\}$  denotes the multiset of the positions of all robots at time  $t$ . When no ambiguity arises, we will omit the temporal indication.

We sometimes express positions according to a polar coordinate system, with the center of the smallest enclosing circle as origin. Given a point  $p$ , we denote its polar coordinates by  $\rho_p$  and  $\theta_p$ , where  $\rho_p$  is the length of the segment  $\overline{op}$ , and  $\theta_p$  is the angle that the segment  $\overline{op}$  makes with the  $x$  positive axis (in trigonometric orientation).

**Alignment with the origin** Two robots are said to be *aligned with the origin* if they both have the same angular position (according to the polar coordinates). In other words, two robots are considered to be *aligned with the origin* only if they are located on the same radius (i.e, between the center and the boundary of the circle). In particular, two robots that lie on the same diameter, but on opposite sides with respect to the center, are not together aligned with the origin. This is because their respective angular position differ by  $\pi$ .

**Virtual ring** The robots form a virtual ring according to their respective positions. The ring is defined by looking at the angular part of the polar coordinates of the robots. Given a robot  $r_i$ , robot  $prev_{r_i}$  is its direct neighbor clockwise, and robot  $next_{r_i}$  is its direct neighbor anticlockwise. In the case when robots are *aligned with the origin*, the distance from the origin is used to define the sequence. In other words, when the angle of two robots is the same, a shorter distance is regarded as being a null angle clockwise (and anticlockwise for a longer distance).

## 4 Circle Formation for Oblivious Robots

### 4.1 Algorithm Intuition

Given the Suzuki and Yamashita [15] model (see Section 3.1) with oblivious robots, and an initial configuration in which a collection of robots are located arbitrarily on the plane, the algorithm ensures that the system (1) solves the *Circle Formation* problem (Prob. 3.2) deterministically, and (2) converges toward a solution to the *Uniform Circle Formation* problem (Prob. 3.1).

Informally, the algorithm relies on the fact that the smallest circle enclosing all robots is unique and depends only on the relative positions of the robots. So, the algorithm makes sure that the smallest enclosing circle remains invariant and uses it as a common reference. The invariance is ensured by self-imposing some constraints on the movements of the robots (Section 4.2). Then, robots that are in the interior of the circle are made to move toward its boundary, while the robots that are already on the boundary are made to move along the circumference.

In order to prevent the situation of inseparable robots discussed earlier, the algorithm must guarantee that no two robots move to the same location. To do so, the algorithm defines an exclusive zone for each robot and for each activation step, within which the robot must make its movement. Doing so ensures that no two robots can be at the same place at the same time. Our algorithm must rely on the fact that activations are atomic, and thus two robots activated simultaneously observe the exact same configuration (albeit according to their respective coordinate system).<sup>7</sup>

---

<sup>7</sup> It is not difficult to extend the algorithm to work in a more loosely synchronized model in which some "fast" robots may be activated up to  $k$ -times during a single activation of the "slowest" robot, where  $k$  is a known bound.

## 4.2 Restrictions on Movement

We first present two restrictions imposed on the movement of robots that are located on the boundary of the smallest enclosing circle. The aim of these restrictions is to preserve the invariance of the smallest enclosing circle, that is, to prevent the robots from making movements that may lead to breaking this circle. For the sake of clarity, these restrictions do not appear explicitly in the algorithm, but must be enforced nevertheless.

**Restriction 4.1** *Robots located on the circumference of the smallest enclosing circle do not move unless there are at least three such robots with distinct positions.*

If the smallest enclosing circle is defined by only two points, these points define a diameter of the circle. Thus, if one of them moves, the circle is broken.

**Restriction 4.2** *Let  $P_c(t)$  be the set of robots on the boundary of  $\mathcal{C}$  at time  $t$ , and  $r_i$  one such robot. Let  $prev_{r_i}(t)$  (resp.,  $next_{r_i}(t)$ ) denote the direct clockwise (resp., counter-clockwise) neighbor of  $r_i$  on  $P_c(t)$ . Let also  $\alpha_{prev_{r_i}}(t)$  and  $\alpha_{next_{r_i}}(t)$  be the angular distance from  $r_i$  to  $prev_{r_i}(t)$  and  $next_{r_i}(t)$ , respectively. Then, the angular movement of  $r_i$  at time  $t + 1$ , denoted by  $\alpha_m(t + 1)$  is restricted as follows:*

$$\frac{\alpha_{prev_{r_i}}(t) - \pi}{2} \leq \alpha_m(t + 1) \leq \frac{\pi - \alpha_{next_{r_i}}(t)}{2}$$

The above restriction ensures that the movement of robots located on the smallest enclosing circle does not leave an empty angle greater than  $\pi$ , or else  $\mathcal{C}$  would no longer be the smallest circle enclosing all robots.

## 4.3 Algorithm Description

We now describe the algorithm in more details, and give a pseudo-code description (see Algorithm 1).<sup>8</sup> As already mentioned, the robots use the smallest circle enclosing all robots  $\mathcal{C}$  as the target circle for solving the problem. Starting from any configuration in which the robots are located arbitrarily on the plane (but with distinct locations), the algorithm ensures that robots located in the interior of  $\mathcal{C}$  reach its boundary in a finite number of activations (Prob. 3.2), and that the robots located on the boundary converge to a

---

<sup>8</sup> The problem is trivially solved by doing nothing for cases where there are only one or two robots. Therefore, in the rest of the section we consider the cases with three or more robots.

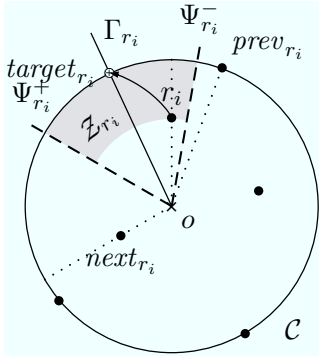


Fig. 1. Principle of the algorithm.

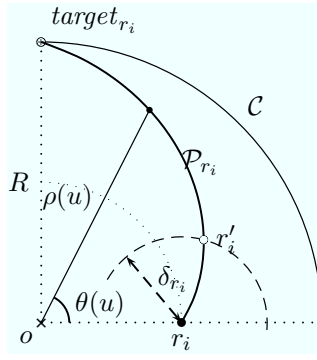


Fig. 2. Parametric path  $\mathcal{P}_{r_i}$  computed by robot  $r_i$ .

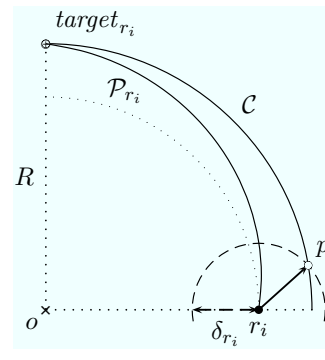


Fig. 3.  $target_{r_i}$  is out of reach, while  $\mathcal{C}$  is not;  $r_i$  joins  $\mathcal{C}$  at point  $p$ .

situation where they are evenly spread on this boundary (Prob. 3.1). In fact, the algorithm can be seen as a combination of two algorithms that solve the two problems simultaneously.

---

**Algorithm 1** Circle Formation for Oblivious Robots (code executed by robot  $r_i$ )

---

**function**  $\varphi_{circle\_uniform}(P, r_i)$

- 1:  $\mathcal{C} :=$  smallest circle enclosing all points in  $P$ ;
  - 2: **if** ( $r_i =$  center of  $\mathcal{C}(P)$ ) **then**
  - 3:  $r_i$  moves to an arbitrary location by some radius  $\rho_{r_i}$  less than the minimum radius of all other robots;
  - 4: **else**
  - 5: Compute  $prev_{r_i}$  and  $next_{r_i}$  (see Sect. 3.3)
  - 6: **if** ( $prev_{r_i}, r_i, next_{r_i}$ ) are aligned with the origin **then**
  - 7: stay still;
  - 8: **else**
  - 9:  $\alpha_{prev_{r_i}} :=$  angular distance between  $r_i$  and  $prev_{r_i}$  in clockwise orientation;
  - 10:  $\alpha_{next_{r_i}} :=$  angular distance between  $r_i$  and  $next_{r_i}$  in anticlockwise orientation;
  - 11:  $\Psi_{r_i}^- :=$  bisector of the angle  $\alpha_{prev_{r_i}}$ ;
  - 12:  $\Psi_{r_i}^+ :=$  bisector of the angle  $\alpha_{next_{r_i}}$ ;
  - 13:  $\Gamma_{r_i} :=$  bisector of the angle formed by  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$ ;
  - 14:  $target_{r_i} := \Gamma_{r_i} \cap \mathcal{C}$ ;
  - 15: Compute path  $\mathcal{P}_{r_i}$  from  $r_i$  to  $target_{r_i}$  (Eq. (2));
  - 16: **if**  $dist(r_i, \mathcal{C}) \leq \delta_{r_i}$  **then**
  - 17: Move to  $\mathcal{C}$ ;
  - 18: **else**
  - 19: Move along  $\mathcal{P}_{r_i}$  toward  $target_{r_i}$  by  $\delta_{r_i}$ ;
  - 20: **end if**
  - 21: **end if**
  - 22: **end if**
- 

The algorithm works as follows: when a robot  $r_i$  becomes active, it executes

the following steps.

- (1)  $r_i$  computes the smallest enclosing circle  $\mathcal{C}$ , based on the observed position of the robots (Alg. 1, line 1), and changes its coordinate system to a polar one, with the origin located at point  $o$ ; the center of  $\mathcal{C}$ .
- (2) If  $r_i$  happens to be located at  $o$ , then  $r_i$  moves out of the center (in any arbitrary direction) by a distance smaller than the minimal radial position of all other robots (Alg. 1, line 3). END.
- (3) Otherwise,  $r_i$  locates two robots  $prev_{r_i}$  and  $next_{r_i}$ , according to the description of the virtual ring in Sect. 3.3 (Alg. 1, line 5).
- (4) If  $prev_{r_i}$ ,  $r_i$ , and  $next_{r_i}$  are together *aligned with the origin*, then  $r_i$  does nothing (Alg. 1, line 7). END.
- (5) If not, then  $r_i$  computes three rays starting from  $o$ , called  $\Psi_{r_i}^-$ ,  $\Psi_{r_i}^+$ , and  $\Gamma_{r_i}$  (see Fig. 1).  $\Psi_{r_i}^-$  is defined as the bisector of the angle  $\alpha_{prev_{r_i}} = \angle r_i o prev_{r_i}$ , and  $\Psi_{r_i}^+$  is defined similarly with  $next_{r_i}$ .  $\Gamma_{r_i}$  is the bisector of the angle formed by  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$  (Alg. 1, line 13).

The algorithm must prevent two robots activated simultaneously from moving to the same location because, otherwise, it may become impossible to separate them (i.e., there exist some activation schedule whereby the robots always move together). To prevent this situation from occurring, we define a zone in which  $r_i$  alone is allowed to move during that activation. We call such a zone the *exclusive zone* of robot  $r_i$  for activation time  $t$ , denoted  $\mathcal{Z}_{r_i}(t)$ , and defined as follows:

$$\mathcal{Z}_{r_i}(t) = \{r_i(t)\} \cup \left\{ p \in \mathbb{R}^2 \mid (\rho_{r_i(t)} \leq \rho_p \leq R) \wedge (\alpha_{\Psi_{r_i}^-(t)} < \alpha_p < \alpha_{\Psi_{r_i}^+(t)}) \right\} \quad (1)$$

The zone is depicted as a gray area on Figure 1. It is important to stress that the bisectors  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$  do not belong to the exclusive zone of  $r_i$ . In fact, when the three robots  $prev_{r_i}$ ,  $r_i$ , and  $next_{r_i}$  are aligned with the origin,  $\Psi_{r_i}^+$  and  $\Psi_{r_i}^-$  are coincident, and thus  $\mathcal{Z}_{r_i}$  includes only the current position of  $r_i$ . We now resume the description of the algorithm.

- (6) Based on  $\Gamma_{r_i}$ ,  $r_i$  computes a target location  $target_{r_i}$ , as the intersection of  $\Gamma_{r_i}$  with  $\mathcal{C}$ . Notice that, by definition,  $target_{r_i}$  is always located in  $\mathcal{Z}_{r_i}$  (Alg. 1, line 14).
- (7) If  $r_i$  can reach  $target_{r_i}$  directly, then it moves there. END.
- (8) If  $r_i$  cannot reach  $target_{r_i}$  directly, but can reach  $\mathcal{C}$ , then it moves<sup>9</sup> to the reachable point on  $\mathcal{C}$  that is nearest to  $target_{r_i}$  (see Fig. 3). Note that this point must be within  $\mathcal{Z}_{r_i}$  of  $r_i$ . (Alg. 1, line 17) END.

---

<sup>9</sup> The movement of step 8 may seem surprising at first. This movement is used to compute an upper bound on the number of activations necessary for robot  $r_i$  to reach the boundary of  $\mathcal{C}$  (see Lemma 5.13). Without this movement, some situation may occur when,  $target_{r_i}$  remains out of reach at every activation (because it rotates), robot  $r_i$  is unable to reach  $\mathcal{C}$  in finite time due to the Zeno paradox.

- (9) Otherwise,  $r_i$  computes a parametric path  $\mathcal{P}_{r_i}$  from  $r_i$  to  $target_{r_i}$ , as a linear motion in the polar space (see definition of  $\mathcal{P}_{r_i}$  below).  $r_i$  moves as far as possible (i.e., maximum is  $\delta_{r_i}$ ) along this path (see Fig. 2). END.

The parametric path  $\mathcal{P}_{r_i}$  computed by a robot  $r_i$  at time  $t$  is defined by the following equations:

$$\mathcal{P}_{r_i}(t) = \begin{cases} \theta(u) = \theta_{r_i(t)} + u(\theta_{target_{r_i}(t)} - \theta_{r_i(t)}) \\ \rho(u) = \rho_{r_i(t)} + u(R - \rho_{r_i(t)}) \\ 0 \leq u \leq 1 \end{cases} \quad (2)$$

## 5 Correctness

In this section, we prove the correctness of our algorithm by first showing that no two robots ever move to the same location (Theorem 5.5). Second, we prove that the smallest enclosing circle remains invariant (Theorem 5.6). Then, we show that all robots reach the boundary of the circle in finite time (Theorem 5.15). Finally, we prove that the algorithm converges toward a configuration wherein all robots are located at regular intervals on the circle (Theorem 5.22).

We first state two lemmas that derive trivially from Algorithm 1.

**Lemma 5.1** *No robot ever moves beyond the boundary of the smallest circle enclosing all robots.*

**Lemma 5.2** *All robots located on the boundary of the smallest enclosing circle remain on that boundary.*

### 5.1 Non-overlapping Zones

We begin by establishing the common context in which we prove several lemmas.

Let us consider some arbitrary time  $t$ , and an arbitrary pair of robots  $r_a$  and  $r_b$ , such that  $r_b = next_{r_a}$  at time  $t$  (i.e.,  $r_a$  and  $r_b$  are consecutive at  $t$ ) and no two robots are located at the same position. The rest of the argument can be repeated for any time and any pair of consecutive robots.

We consider the four robots  $prev_{r_a}$ ,  $r_a$ ,  $r_b$ , and  $next_{r_b}$  and their relative angles at time  $t$ . We set the reference angle of our polar coordinate system to be the

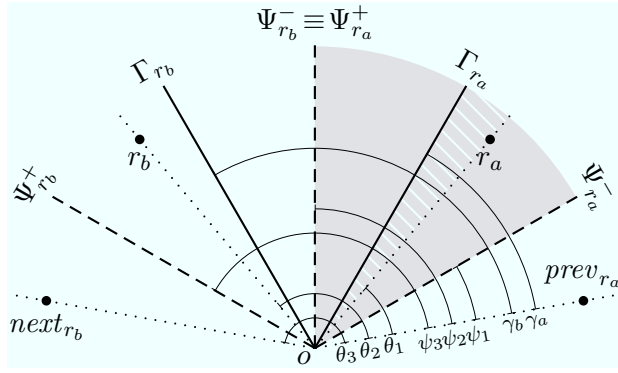


Fig. 4. Invariance of virtual ring: consecutive robots  $r_a$  and  $r_b$ .

angular position of robot  $prev_{r_a}$  (see Fig. 4). Let  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  denote the angles of robots  $r_a$ ,  $r_b$ , and  $next_{r_b}$ , respectively. We also consider the bisectors  $\Psi_{r_a}^-$ ,  $\Psi_{r_a}^+$ ,  $\Psi_{r_b}^+$ , used in the definition of the movement. Notice that  $\Psi_{r_b}^- \equiv \Psi_{r_a}^+$  because  $r_a = prev_{r_b}$ . Let  $\psi_1$ ,  $\psi_2$ , and  $\psi_3$  denote the angles of  $\Psi_{r_a}^-$ ,  $\Psi_{r_a}^+$ , and  $\Psi_{r_b}^+$ , respectively. Finally, we consider the two second-order bisectors  $\Gamma_{r_a}$  and  $\Gamma_{r_b}$ , and let  $\gamma_a$  and  $\gamma_b$  denote their respective angles. Remind that the respective targets of  $r_a$  and  $r_b$  are located on  $\Gamma_{r_a}$  and  $\Gamma_{r_b}$ .

From this, we obtain the following relations between those angles.

$$\begin{aligned}
 0 &\leq \psi_1 \leq \theta_1 \leq \psi_2 \leq \theta_2 \leq \psi_3 \leq \theta_3 \\
 &\parallel \qquad \qquad \parallel \qquad \qquad \parallel \\
 \psi_1 &\leq \gamma_a \leq \psi_2 \leq \gamma_b \leq \psi_3
 \end{aligned} \tag{3}$$

**Lemma 5.3** *There is no overlap between the exclusive zones of any two consecutive robots.*

PROOF. We consider the situation above and reason on the angles. The exclusive zone of robot  $r_a$  consists of the position of  $r_a$  and a zone included in the open angular interval  $(\psi_1; \psi_2)$ . Note that, because it is open, the interval can possibly be empty (when  $\psi_1 = \psi_2$ ). Similarly, the zone of  $r_b$  consists of the position of  $r_b$  and a zone included in the interval  $(\psi_2; \psi_3)$ .

- (1) The locations of  $r_a$  and  $r_b$  are distinct by hypothesis.
- (2) The intervals do not intersect. The intervals are open, which means that the points on the rays do not belong to the zones. We simply need to show that  $\psi_1 < \psi_3$ , but this is already obvious from Relation (3).
- (3) The location of one of the two robots (say  $r_a$ ) does not belong to the interval of the other robot (say  $r_b$ ). Consider the angular position of  $r_a$ ,  $\theta_1$ , and the interval of  $r_b$ ,  $(\psi_2; \psi_3)$ . By Relation (3), we have that  $\theta_1 \leq \psi_2 \leq \psi_3$ . Since the rays do not belong to the interval,  $r_a$  is not in the interval of  $r_b$ , even when  $\theta_1 = \psi_2$ .

**Lemma 5.4** *There is no overlap between the exclusive zones of any two robots.*

PROOF. The proof is a generalization of Lemma 5.3, by a simple induction on a string of consecutive robots.

A special case occurs when a robot is located on the center of the smallest enclosing circle. This is treated separately. Let  $r_o$  be that robot. It must be unique by hypothesis. The zone of  $r_o$  is defined by the circle centered at  $o$  and with radius  $r$ , such that  $r < \min_{r \in \mathcal{R} \setminus \{r_o\}} \rho_r$ . Since the points in the zone of any other robot  $r$  must have a radial position of at least  $\rho_r$ , there can be no intersection with the zone of  $r_o$ . □ Lemma 5.4

**Theorem 5.5** *Under Algorithm 1, no two robots ever move to the same location.*

PROOF. We show that a robot  $r_i$  always move to a location within its own exclusive zone  $\mathcal{Z}_{r_i}$ , and the rest follows from the fact that the zones of two robots do not intersect (Lemma 5.4). Let us consider a robot  $r_i$  and its new location  $r'_i$ . There are two cases.

First,  $prev_{r_i}$ ,  $r_i$ , and  $next_{r_i}$  are aligned together with the origin. The location of  $r_i$  belongs to the zone ( $\mathcal{Z}_{r_i}$  is equal to the location of  $r_i$ ), and  $r_i$  does not move.

Second,  $prev_{r_i}$  and  $next_{r_i}$  are not aligned. Then,  $\Gamma_{r_i}$  is located between  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$ , and all three are distinct. It follows that  $target_{r_i}$  is strictly between  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$  (and thus lies in  $\mathcal{Z}_{r_i}$ ).  $r_i$  is also between  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$ , but not strictly (i.e.,  $r_i$  can be on either one of the two axes). Because  $r_i$  belongs to its zone, and because the angle of points in the path are defined linearly, all points between  $r_i$  and  $target_{r_i}$  must be in  $\mathcal{Z}_{r_i}$ . □ Theorem 5.5

## 5.2 Invariance of the Smallest Enclosing Circle

**Theorem 5.6** *The smallest enclosing circle  $\mathcal{C}$  is invariant.*

PROOF. Let  $\mathcal{C}(t)$  and  $\mathcal{C}(t+1)$  denote the smallest enclosing circle at time instants  $t$  and  $t+1$  respectively. We prove that, regardless of the activation schedule,  $\mathcal{C}(t)$  and  $\mathcal{C}(t+1)$  must be identical, and the rest follows by induction.



Assume, by contradiction, that there is a time instant  $t$  for which  $\mathcal{C}(t)$  and  $\mathcal{C}(t + 1)$  are different. First, we observe that this cannot be caused by the movement of a robot located at the interior of  $\mathcal{C}(t)$ . Indeed, such a robot could change the smallest enclosing circle only by moving outside of it, (a contradiction with Lemma 5.1). Therefore  $\mathcal{C}(t + 1)$  must be defined by the movement of robots located at the boundary of  $\mathcal{C}(t)$ . There are four cases left to consider, depending on the number of robots at the boundary of  $\mathcal{C}(t)$  and their respective position:

- (1) (*2 robots*) The smallest enclosing circle  $\mathcal{C}(t)$  is defined by only two robots. Those robots cannot move by Restriction 4.1 and hence  $\mathcal{C}(t + 1) = \mathcal{C}(t)$ .
- (2) (*3 robots; one quits the circle*) The smallest enclosing circle  $\mathcal{C}(t)$  is defined by three robots, one of which moves outside the boundary of  $\mathcal{C}(t)$ . This is a contradiction with Lemma 5.1.
- (3) (*3 robots; two distinct points*) The smallest enclosing circle  $\mathcal{C}(t)$  is defined by three robots, two of which move to the same location. This is in contradiction with Theorem 5.5.
- (4) (*3 robots; angular distance greater than diameter*) If the angular distance between two of the three robots is larger than the diameter, then the circle defined by the three robots and the smallest enclosing circle for the two robots are different. Since  $\mathcal{C}(t)$  is the smallest enclosing circle at time  $t$ , the angular distance between any two of the three robots must be not greater than the diameter. By Restriction 4.2, the movement of two consecutive robots cannot lead them further away from each other than  $\pi$ , regardless of their activation schedule.

When there are more than three robots on the boundary of  $\mathcal{C}(t)$ , the situation can always be reduced to one of the four cases mentioned above. It follows that  $\mathcal{C}(t)$  and  $\mathcal{C}(t + 1)$  cannot be different; a contradiction.  $\square_{\text{Theorem 5.6}}$

The following lemma is obtained easily from the algorithm.

**Lemma 5.7** *For any robot  $r_i$ , its radial position  $\rho_{r_i}(t)$  is nondecreasing.*

**Lemma 5.8** *There is a time since which no robot is on the center of  $\mathcal{C}$ .*

**PROOF.** Let  $r_o$  be a robot located at the center of  $\mathcal{C}$ . By the fairness of the activation, there is a time  $t$  when it becomes active. From line 3 of Algorithm 1,  $r_o$  is no longer at the center at time  $t + 1$ . From Lemma 5.7, the radial position is nondecreasing, and thus no robot can be located at the center of  $\mathcal{C}$  after time  $t$ .  $\square_{\text{Lemma 5.8}}$

**Theorem 5.9** *From the time when no robot is located at the center of  $\mathcal{C}$ , the virtual ring remains invariant.*

PROOF. We consider again the situation of Section 5.1, and we must show that, at time  $t + 1$ ,  $r_a$  must be before  $r_b$ , and the rest follows by applying the same argument to all pairs of consecutive robots.

The position of  $r_a$  at time  $t + 1$  must be between the axes of  $r_a$  and  $\Gamma_{r_a}$  (i.e., the hatched zone in Fig. 4). This means that the angular position must be in the angular interval  $I_a = [\min(\theta_1, \gamma_a); \max(\theta_1, \gamma_a)]$ . Similarly, the new position of  $r_b$  must be in the interval  $I_b = [\min(\theta_2, \gamma_b); \max(\theta_2, \gamma_b)]$ .

By definition, the position that  $r_a$  will take at time  $t + 1$  must also be located within the zone of  $r_a$  at time  $t$ .

Then, we need to distinguish two cases.

- (1)  $\theta_1 < \theta_2$ . From this and the fact that most angles are defined as bisectors, we can refine Relation (3) as follows.

$$\begin{array}{ccccccc} 0 & \leq & \psi_1 & \leq & \theta_1 & < & \psi_2 < \theta_2 \leq \psi_3 \leq \theta_3 \\ & & \parallel & & & & \parallel & & \parallel \\ & & \psi_1 & < & \gamma_a & < & \psi_2 & < & \gamma_b & < & \psi_3 \end{array}$$

From the above relation, we can directly derive.

$$\max(\theta_1, \gamma_a) < \min(\theta_2, \gamma_b)$$

Thus, the order between  $r_a$  and  $r_b$  is preserved.

- (2)  $\theta_1 = \theta_2$ . The two robots  $r_a$  and  $r_b$  are aligned together with the origin. The only points of that ray that belongs to their zone is their respective location. In this case, the order is defined by the distance from the origin, which cannot change at time  $t + 1$  because of the invariance of the smallest enclosing circle (Theorem 5.6). Since all other points in the zone of  $r_a$ , if they exist, have an angle strictly smaller than  $\theta_1 = \theta_2$ , and strictly greater for  $r_b$ , the order between  $r_a$  and  $r_b$  is preserved.

□ Theorem 5.9

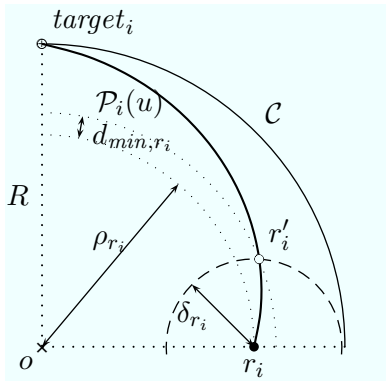


Fig. 5. The minimum distance of progress of  $r_i$  toward the boundary of the circle is  $d_{min,r_i}$ .

#### 5.4 Circle Formation

In the following, we will show that all robots located in the interior of  $\mathcal{C}$  reach its boundary after a finite number of activation steps.

We have observed that, at each time instant a robot  $r_i$  becomes active, it computes a new target (the target is dynamic). Depending on the activation of the neighbors of  $r_i$ , its target at time  $t + 1$  can be closer or farther than at time  $t$ . However, we also observed that the maximum angle that can separate a robot from its target is  $\frac{\pi}{4}$ . Then, before proceeding, we establish the following lemma.

**Lemma 5.10** *The angle that separates a robot  $r_i$  from its target  $target_{r_i}$  is at most  $\frac{\pi}{4}$ .*

PROOF. By Restriction 4.2, the maximum angular distance that can separate any two consecutive robots is  $\pi$ . Consider some robot  $r_i$ , the extreme case occurs where  $r_i$  forms a minimal angle with one of its neighbors, say  $prev_{r_i}$ , and a maximal angle with its other neighbor, say  $next_{r_i}$ . Let us thus consider the situation where  $r_i$  and  $prev_{r_i}$  are aligned with the origin at angle 0, and where the angular distance between  $r_i$  and  $next_{r_i}$  is  $\pi$ .

It follows that  $\Psi_{r_i}^-$  is at a null angle with respect to  $r_i$ , while  $\Psi_{r_i}^+$  is at angle  $\frac{\pi}{2}$ . Being the bisector of  $\Psi_{r_i}^-$  and  $\Psi_{r_i}^+$ ,  $\Gamma_{r_i}$  is at angle  $\frac{\pi}{4}$ . Since  $target_{r_i}$  is located on  $\Gamma_{r_i}$ , this proves the lemma.  $\square_{\text{Lemma 5.10}}$

**Lemma 5.11** *For any robot  $r_i$  that is not aligned with the origin and with its previous and next neighbors, there exists a minimum distance  $d_{min,r_i} > 0$  that  $r_i$  can progress toward the boundary of the circle.*

PROOF. To prove the lemma, we consider the situation where  $r_i$  can progress

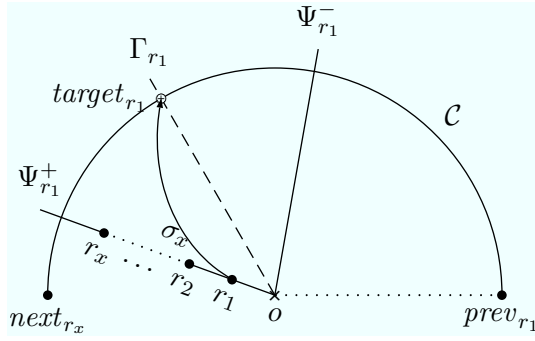


Fig. 6. String of robots *aligned with the origin*.

the least. It is easy to see that this situation occurs when the angular distance with the target is maximal (i.e.,  $\frac{\pi}{4}$  by Lemma 5.10) and  $r_i$  is as close as possible to  $\mathcal{C}$  without being able to reach it (see Figure 5).

Observe that  $r_i$  can progress away from the center of  $\mathcal{C}$  by at least  $d_{min,r_i}$  when moving toward  $target_{r_i}$ . In this situation, the range of  $r_i$  ( $\delta_{r_i}$ ) is just too short for reaching  $\mathcal{C}$ . Thus,  $r_i$  will move to location  $r'_i$ .  $d_{min,r_i}$  is equal to the difference between  $\rho_{r'_i}$  and  $\rho_{r_i}$ , and it is positive. Thus,  $d_{min,r_i} > 0$  represents the minimum distance that  $r_i$  can move away from the center of  $\mathcal{C}$  and the lemma holds. □<sub>Lemma 5.11</sub>

**Lemma 5.12** *Starting from any configuration in which some robots are aligned with the origin, there is a time after which no two robots are aligned together with the origin.*

**PROOF.** We consider an arbitrary string of  $x$  robots  $\sigma_x = r_1, \dots, r_x$  with increasing distance from the origin, and aligned together with the origin (see Fig. 6). First, it is easy to see that no new robot joins  $\sigma_x$  (see proof of Theorem 5.5), and then the rest of the proof is by induction on  $x$ , the number of robots at  $\sigma_x$ .

*Basis:* ( $x = 1$ ). The lemma holds trivially.

*Induction Step:* Assume that the lemma holds for any string  $\sigma_y$  shorter than  $x$  ( $y < x$ ), and let us prove that the lemma holds for a string  $\sigma_x$  of length  $x$ . Let us consider one of the two robots at the extremity of the string, say  $r_1$  (the argument is the same for  $r_x$ ).

By assumption, the scheduler is fair, hence eventually  $r_1$  becomes active. Since  $r_1$  is at the extremity of the string,  $r_1$  and  $prev_{r_1}$  cannot be aligned together with the origin, and thus the test on line 6 in the algorithm evaluates to false. So,  $r_1$  computes a path  $\mathcal{P}_{r_1}$  at line 15.

$r_1$  and  $prev_{r_1}$  not being aligned with the origin, means that  $\Psi_{r_1}^-$  and  $\Psi_{r_1}^+$  are

distinct, and so is  $\Gamma_{r_1}$ . It follows that  $target_{r_1}$  has an angular position different from that of  $r_1$ . Thus, except for the initial location of robot  $r_1$ , no other point on  $\mathcal{P}_{r_1}$  is aligned with  $\Psi_{r_1}^+$  and the other robots of the string. Because  $\delta_{r_1}$  is greater than zero, the destination  $r'_1$  of  $r_1$  cannot be aligned with the robots of  $\sigma$ , regardless of the test in line 16. Thus, after its move,  $r_1$  no longer belongs to the string  $\sigma$ , thus decreasing its length by one. This proves the induction step.  $\square_{\text{Lemma 5.12}}$

**Lemma 5.13** *All robots located in the interior of  $\mathcal{C}$  reach its circumference in finite time.*

PROOF. By Lemma 5.12, if there exists a configuration wherein some robots are *aligned with the origin*, there is a finite number of steps, where this configuration is reduced to the general case. From Lemma 5.11, at each activation step, a robot  $r_i$ , not located on the boundary of  $\mathcal{C}$ , can progress by at least a radial distance  $d_{min,r_i} > 0$  toward the periphery of the circle. It follows that, regardless of the initial position of some robot  $r_i$ , the number of activation steps it takes for  $r_i$  to reach the boundary of  $\mathcal{C}$  is bounded above by  $\frac{R}{d_{min,r_i}}$ . Thus, due to the fairness of the activation schedule, the boundary of  $\mathcal{C}$  is reached in finite time and the lemma holds.  $\square_{\text{Lemma 5.13}}$

**Lemma 5.14** *The global predicate that all robots are located on the boundary of  $\mathcal{C}$  is stable.*

PROOF. Let us denote by  $C_{circle}$ , the set of all configurations in which all robots are located on the boundary of  $\mathcal{C}$ . Then, we show that, for any configuration  $c$  in  $C_{circle}$ , the algorithm always leads to a configuration  $c'$  in  $C_{circle}$ .

Consider some robot  $r_i$  that becomes active. By the algorithm,  $r_i$  computes a new  $target_{r_i}$ , located on  $\mathcal{C}$ . Because  $r_i$  is also on  $\mathcal{C}$ , the entire path  $\mathcal{P}_{r_i}$  is located on  $\mathcal{C}$ . Thus,  $r_i$  can only move to a location on the boundary of  $\mathcal{C}$ . It follows that configuration  $c'$  is in  $C_{circle}$ .  $\square_{\text{Lemma 5.14}}$

**Theorem 5.15** *The algorithm solves the circle formation problem deterministically.*

PROOF. There is a time after which all robots are located on the boundary of a circle (Lemma 5.13), and this situation is stable (Lemma 5.14).  $\square_{\text{Theorem 5.15}}$

## 5.5 Uniform Transformation

We now show that our algorithm *converges* toward a uniform distribution of robots along the boundary. Before we proceed, we give few additional definitions:

**Definition 5.16** For any robot  $r_i$ , let  $\alpha_{r_i}(t)$  denote the angular distance between  $r_i$  and  $next_{r_i}$ . Thus,  $\alpha_{r_i}(t) = \theta_{next_{r_i}}(t) - \theta_{r_i}(t)$ .

**Definition 5.17** Let  $\alpha_{max}(t)$  (resp.,  $\alpha_{min}(t)$ ) be the maximal (resp., minimal) angular distance between any two consecutive robots, at time  $t$ . Thus,  $\alpha_{max}(t) = \max_{r_i} \alpha_{r_i}(t)$  and  $\alpha_{min}(t) = \min_{r_i} \alpha_{r_i}(t)$ .

**Lemma 5.18** The function  $\alpha_{max}(t)$  is nonincreasing, and the function  $\alpha_{min}(t)$  is nondecreasing.

PROOF. We only prove the lemma for  $\alpha_{max}(t)$ , as the proof for  $\alpha_{min}(t)$  is then easily derived by symmetry.

Let  $t$  be some time, and  $r_i$  some robot. Obviously,  $\alpha_{r_i}(t+1)$  is maximized when (1) both robots  $r_i$  and  $next_{r_i}$  are active at time  $t$ , (2) they are moving away from each other, and (3) can reach their respective target point.

Thus, assuming that both robots  $r_i$  and  $next_{r_i}$  are active at time  $t$ , we obtain:

$$\begin{aligned} \alpha_{r_i}(t+1) &= \frac{\alpha_{r_i}(t)/2 + \alpha_{next_{r_i}}(t)/2}{2} + \frac{\alpha_{r_i}(t)/2 + \alpha_{prev_{r_i}}(t)/2}{2} \\ &= \frac{2\alpha_{r_i}(t) + \alpha_{next_{r_i}}(t) + \alpha_{prev_{r_i}}(t)}{4} \\ &\leq \alpha_{max}(t) \end{aligned} \tag{4}$$

The inequality is obtained by replacing  $\alpha_{r_i}(t)$ ,  $\alpha_{prev_{r_i}}(t)$  and  $\alpha_{next_{r_i}}(t)$  by  $\alpha_{max}(t)$ . It follows that, for any time  $t$ ,  $\alpha_{max}(t+1) \leq \alpha_{max}(t)$ .  $\square$  Lemma 5.18

**Corollary 5.19**  $\forall t, \forall r_i : \alpha_{min}(t) \leq \alpha_{r_i}(t+1) \leq \alpha_{max}(t)$

**Lemma 5.20** Every configuration in which all robots are uniformly distributed over the circle is stable.

PROOF. Assume that, at some time  $t$ , the robots are uniformly distributed. In such a configuration, the angular distance between any two consecutive robots must be the same:  $\frac{2\pi}{n}$ . It follows that,  $\alpha_{min}(t) = \alpha_{max}(t) = \frac{2\pi}{n}$ , from

which we derive,

$$\forall t, \forall r_i : \frac{2\pi}{n} = \alpha_{min}(t) \leq \alpha_{r_i}(t+1) \leq \alpha_{max}(t) = \frac{2\pi}{n}$$

and this completes the proof.  $\square$ <sub>Lemma 5.20</sub>

**Lemma 5.21** *The function  $\Delta(t) = \alpha_{max}(t) - \alpha_{min}(t)$  is monotonically decreasing and converges to zero.*

PROOF. First of all, from Lemma 5.18, we can deduce that  $\Delta(t)$  is non-increasing. We must show that, for any time  $t$ , if  $\alpha_{min}(t) < \alpha_{max}(t)$ , then, eventually, either  $\alpha_{min}(t')$  increases or  $\alpha_{max}(t')$  decreases. In other words,

$$\forall t : \alpha_{min}(t) < \alpha_{max}(t) \Rightarrow (\exists t' > t : (\alpha_{max}(t') < \alpha_{max}(t)) \vee (\alpha_{min}(t) < \alpha_{min}(t')))$$

First, let us show that an angle  $\alpha_{r_i}(t)$  strictly smaller than  $\alpha_{max}(t)$  at time  $t$ , must always be smaller than  $\alpha_{max}(t)$  after time  $t$  (although  $\alpha_{r_i}(t)$  can possibly increase). In other words,

$$\forall t \forall r_i : \alpha_{r_i}(t) < \alpha_{max}(t) \Rightarrow (\forall t' > t : \alpha_{r_i}(t') < \alpha_{max}(t))$$

This is done easily by induction. Consider that, at time  $t$ ,  $\alpha_{r_i}(t) < \alpha_{max}(t)$ . From Equation (4) in the proof of Lemma 5.18, we have:

$$\alpha_{r_i}(t+1) = \frac{2\alpha_{r_i}(t) + \alpha_{prev_{r_i}}(t) + \alpha_{next_{r_i}}(t)}{4}$$

From which we deduce that  $\alpha_{r_i}(t+1) < \alpha_{max}(t)$ . Since, by Lemma 5.18,  $\alpha_{max}(t+1) \leq \alpha_{max}(t)$ , we indeed have that, for any time  $t'$  after  $t$ ,  $\alpha_{r_i}(t') < \alpha_{max}(t)$ .

To complete the proof of the lemma, we must now show that, if an angle  $\alpha_{r_i}(t)$  is maximal at time  $t$  ( $\alpha_{r_i}(t) = \alpha_{max}(t)$ ), then there must be a time  $t'$  in the future when it becomes smaller. In other words,

$$\forall t \forall r_i : \alpha_{r_i}(t) = \alpha_{max}(t) \Rightarrow (\exists t' > t : \alpha_{r_i}(t') < \alpha_{max}(t))$$

Observe that if  $\alpha_{r_i}(t)$  is equal to  $\alpha_{max}(t)$ , then  $\alpha_{r_i}(t)$  decreases only when  $\alpha_{prev_{r_i}}(t)$  is less than  $\alpha_{max}(t)$ .

Assume that  $\alpha_{r_i}(t) = \alpha_{prev_{r_i}}(t) = \alpha_{max}(t)$ . Since,  $\alpha_{min}(t) < \alpha_{max}(t)$  by hypothesis, and there is a finite number of robots. Thus, there must be some robot  $r_j$  such that  $\alpha_{r_j}(t) \leq \alpha_{max}(t)$  and  $\alpha_{prev_{r_j}}(t) < \alpha_{max}(t)$ .

By the fairness of the scheduler, there must be a time  $t''$  for  $r_j$  when  $\alpha_{r_j}(t'') < \alpha_{max}(t)$ . By applying induction repeatedly on the robots, we obtain that from some time  $t'''$ , and for all robots  $r_k$ ,  $\alpha_{r_k}(t''') < \alpha_{max}(t)$ .

The same proof can be adapted for the minimum, and we have that, for any time  $t$  when  $\alpha_{min}(t) < \alpha_{max}(t)$ , there will be a time  $t'$  in the future when  $\alpha_{max}(t') < \alpha_{max}(t)$  and  $\alpha_{min}(t') > \alpha_{min}(t)$ . Thus,  $\Delta(t) = \alpha_{max}(t) - \alpha_{min}(t)$  converges toward zero.  $\square_{\text{Lemma 5.21}}$

**Theorem 5.22** *Algorithm 1 converges toward a configuration wherein all robots are arranged at regular intervals on the boundary of the circle.*

The theorem comes as a direct consequence of Lemma 5.20 and Lemma 5.21.

## 6 Discussion on the Earlier Version of the Algorithm

In an earlier instance of this paper, Défago and Konagaya [6] used a different algorithm. In short, the earlier algorithm was a composition of two independent algorithms. The first one, solving the circle formation problem, relied also on the definition of exclusive movement zones. However, the zones were defined using the Voronoi cell<sup>10</sup> of each robot, and was executed by all robots until they all reached the boundary of  $\mathcal{C}$ . The second algorithm, converging toward Problem 3.1, took as input the solution of the first algorithm and simply had each robot move along the boundary, halfway toward the midpoint between each neighbors.

Algorithm 1 has several important advantages over the previous algorithm. Most importantly, it is simpler in many different ways. Firstly, it elegantly combines the solution of the two problems into a single algorithm. Secondly, the only somewhat complex geometric computation on which it relies is the smallest enclosing circle. Finally, the computation complexity is smaller. Indeed, finding the smallest enclosing circle can be achieved in  $O(n)$  [17], whereas computing the Voronoi diagram is normally done in  $O(n \log n)$  [9].

---

<sup>10</sup>The *Voronoi diagram*  $\text{Voronoi}(P)$  of a set of points  $P = \{p_1, p_2, \dots, p_n\}$  is a subdivision of the plane into  $n$  cells, one for each point in  $P$ . The cells have the property that a point  $q$  belongs to the Voronoi cell of point  $p_i$ , denoted  $\text{Vcell}_{p_i}(P)$ , if and only if, for any other point  $p_j \in P$ ,  $\text{dist}(q, p_i) < \text{dist}(q, p_j)$ , where  $\text{dist}(p, q)$  is the Euclidean distance between  $p$  and  $q$ . In particular, the strict inequality means that points located on the boundary of the Voronoi diagram do not belong to any Voronoi cell. Significantly more details about Voronoi diagrams and their principal applications are surveyed by Aurenhammer [1].



One main difference between the algorithm of Défago and Konagaya [6] and Algorithm 1 is that the former does not require that robots agree on the chirality since the sense of direction is not essential to compute Voronoi diagrams and halfway toward mid point. However, in the later algorithm the agreement between robots on the chirality is important when robots are aligned with the origin.

## 7 Conclusion

In this paper, we have presented a distributed algorithm whereby a team of oblivious mobile robots self-organize to form a circle in the semi-synchronous model [15]. Our algorithm allows robots to deterministically form an irregular circle within a finite number of activation steps, and asymptotically converges toward a uniform distribution of the robots along the circumference of the circle.

Our algorithm is self-stabilizing<sup>11</sup> with respect to the weakest problem (non uniform circle formation) provided that no two robots have both the same initial position and the same local coordinate system. Without this restriction, the problem is indeed trivially impossible. In addition, it solves the problem linearly at each activation step since it only relies on the computation of the smallest enclosing circle, which can be computed in  $O(n)$  [17], and finding the pair of *previous* and *next* robots, can be also done linearly; the computation of the whole virtual ring is not necessary by the algorithm.

In the meantime, Dieudonné and Petit [20] provided in the same model assumed in this paper (semi-synchronous model) a deterministic solution to the uniform circle formation problem combined with the work of Katreniak [11] for almost every case, leaving unsolvable only the case of four robots. In particular, their algorithm relies on the computation of the convex hull of robots at each activation step, which takes at least  $O(n \log n)$  [9].

Although the result of Dieudonné and Petit [20] is very significant, the problem is not yet completely solved. Our proposed algorithm solves a weaker problem (namely, convergence rather than formation). However, it does not exclude any special case (e.g.,  $n = 4$ ), and has a smaller complexity (linear instead of lin-log).

The results of this paper leaves open several interesting research questions. For instance, it is interesting to see whether the problem can still be solved

---

<sup>11</sup> Self-stabilization is the property of a system which, started in an arbitrary state, always converges toward a desired behavior. We consider a weaker definition that allows to remove trivially impossible cases [8]

deterministically with limited visibility or inaccurate sensors. Indeed, the proposed algorithm must rely on unlimited (or "sufficiently wide") visibility in order to compute the smallest enclosing circle. With limited visibility, this is no longer possible for the robots to compute this circle. This actually raises the question of the existence of a deterministic solution in that model. The question remains open for future studies.

## Acknowledgement

We are especially grateful to the following persons for their insightful comments regarding this work: Maria Gradinariu, Masafumi Yamashita, and Takuya Katayama. We would like to thank Akihiko Konagaya for his comments on an earlier version of this paper.

This work was supported by MEXT Grant-in-Aid for Young Scientists (A) (Nr. 18680007).

## References

- [1] F. Aurenhammer, Voronoi Diagrams—A survey of a fundamental geometric data structure. *acmcs*. No. 23 3 (1991) 345–405.
- [2] T. Balch and R. C. Arkin, Behavior-based Formation Control for Multi-robot Teams, *IEEE Trans. on Robotics and Automation*, No. 14 6 (1998) 926–939.
- [3] I. Chatzigiannakis and M. Markou and S. Nikolettseas, Distributed Circle Formation for Anonymous Oblivious Robots, *Proc. 3rd Workshop on Efficient and Experimental Algorithms (WEA'04)*, (2004) 159–174.
- [4] I. Chatzigiannakis and S. Nikolettseas and P. Spirakis, On the Average and Worst-Case Efficiency of Some New Distributed Communication and Control Algorithms for Ad-hoc Mobile Networks, *Proc. 1st ACM Int'l Workshop on Principles of Mobile Computing (POMC'01)*, (2001) 1–19.
- [5] X. A. Debest, Remark About Self-Stabilizing Systems, *cacm*, No. 38 2 (1995) 115–117.
- [6] X. Défago and A. Konagaya, Circle Formation for Oblivious Anonymous Mobile Robots with no Common Sense of Orientation, *Proc. 2nd ACM Intl Workshop on Principles of Mobile Computing (POMC'02)*, (2002) 97–104.
- [7] M. B. Dias and A. Stentz, A Free Market Architecture for Distributed Control of a Multirobot System, *Proc. 6th Intl. Conf. on Intelligent Autonomous Systems (IAS'00)*, (2000) 115–122.

- [8] M. Schneider, Self-stabilization, *ACM Computing Surveys*, No. 25 1 (1993) 45–67.
- [9] M. Berg and M. Krefeld and M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer-Verlag (1998).
- [10] V. Gervasi and G. Prencipe, *Flocking by A Set of Autonomous Mobile Robots*, Technical report, Dipartimento di Informatica, Università di Pisa, Italy, (2001), TR-01-24.
- [11] B. Katreniak, *Biangular Circle Formation by Asynchronous Mobile Robots*, Proc. 12th Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO'05), Le Mont St-Michel, France, (2005).
- [12] M. J. B. Krieger and J.-B. Billeter and L. Keller, *Ant-Like Task Allocation and Recruitment in Cooperative Robots*, *Nature*, No. 406 6799 (2000), 992–995.
- [13] S. Souissi, *On Distributed Cooperative Mobile Robotics: Decomposition of Basic Problems and Study of a Self-stabilizing Circle Formation Algorithm*, Master thesis, Japan Advanced Institute of Science and Technology, School of Information Science, (2004).
- [14] K. Sugihara and I. Suzuki, *Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots*, *Journal of Robotics Systems*, No. 3 13 (1996), 127–139.
- [15] I. Suzuki and M. Yamashita, *Distributed Anonymous Mobile Robots: Formation of Geometric Patterns*, *SIAM Journal of Computing*, No. 28 4 (1999), 1347–1363.
- [16] E. W. Weisstein, *Reuleaux Triangle*, From MathWorld—A Wolfram Web Resource, <http://mathworld.wolfram.com/ReuleauxTriangle.html>.
- [17] E. Welzl, *Smallest enclosing disks (balls and ellipsoids)*, *New Results and New Trends in Computer Science*, No. 555 (1991), 359–370.
- [18] Y. Dieudonne and O. Labbani-Igbida and F. Petit, *Circle Formation of Weak Mobile Robots*, Proc. of 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'06), LNCS. No. 4280 (2006), 262–275.
- [19] Y. Dieudonné and F. Petit, *Circle Formation of Weak Robots and Lyndon Words*, *Information Processing Letters*, No. 104 4 (2007), 156–162.
- [20] Y. Dieudonné and F. Petit, *Swing Words to Make Circle Formation Quiescent*, Forteenth International Colloquium on Structural Information and Communication Complexity (SIROCCO '07), No. 4474 (2007) 166–179.
- [21] Y. Dieudonné and F. Petit, *A Scatter of Weak Robots*, Technical Report RR07-10, LARIA, CNRS, Amiens, France, 2007.
- [22] P. Flocchini and G. Prencipe and N. Santoro and P. Widmayer, *Hard Tasks for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots*, Proc. 10th Int'l Symp. on Algorithms and Computation (ISAAC'99), No. 1741 (1999), 93–102.

- [23] Uny Cao, Y. and A. S. Fukunaga and A. B. Kahng, Cooperative mobile Robotics: Antecedents and Directions, *Autonomous Robots*, No. 4 (1997), 1–23.
- [24] P. Flocchini and G. Prencipe and N. Santoro and P. Widmayer, Pattern Formation by Autonomous Robots Without Chirality, *Proc. VIII Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO'01)*, (2001) 147–162.
- [25] E. Bonabeau and M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, (1999).
- [26] P. Flocchini and G. Prencipe and N. Santoro and P. Widmayer, Gathering of asynchronous robots with limited visibility, *Theoretical Computer Science*, No. 337 (1–3) (2005) 147–168.
- [27] G. Prencipe, Instantaneous Actions vs. Full Asynchronicity: Controlling and Coordinating a Set of Autonomous Mobile Robots. In *Proc. 7th Italian Conference on Theoretical Computer Science (ICTCS'01)*, (2001) 154–171.
- [28] G. Prencipe, *CORDA: Distributed coordination of a set of autonomous mobile robots*, In *Proc. 4th European Research Seminar on Advances in Distributed Systems (ERSADS'01)*, (2001) 185–190.
- [29] G. Prencipe, On the feasibility of gathering by autonomous mobile robots, In *Proc. Colloquium on Structural Information and Communication Complexity (SIROCCO'05)*, (2005) 246–261.
- [30] S. Souissi and X. Défago and M. Yamashita, Using Eventually Consistent Compasses to Gather Oblivious Mobile Robots with Limited Visibility, In *Proc. 8th Intl. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS'06)*, No. 4280 (2006) 471–487.
- [31] P. Flocchini and G. Prencipe and N. Santoro, Self-Deployment Algorithms for Mobile Sensors on a Ring, In *Proc. 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors'06)*, No. 4240 (2006) 59–70.
- [32] D. Peleg, Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges, In *Proc. 7th International Workshop on Distributed Computing (IWDC'05)* (2005) 1-12.
- [33] N. Agmon and D. Peleg, Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots, *SIAM J. Comput.* No. 36 1 (2006) 56-82.