Self-stabilizing Algorithm for Circle Formation by Disoriented Oblivious Mobile Robots

Xavier Défago and Samia Souissi

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST)

February 26, 2007

IS-RR-2007-003

Japan Advanced Institute of Science and Technology (JAIST)

School of Information Science
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

http://www.jaist.ac.jp/

ISSN 0918-7553

Self-stabilizing Algorithm for Circle Formation by Disoriented Oblivious Mobile Robots *

Xavier Défago Samia Souissi

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
Tel.: +81-761-51 1254 — Fax.: +81-761-51 1149

E-mail: {defago,ssouissi}@jaist.ac.jp

Abstract

This paper presents a distributed algorithm whereby a group of mobile robots self-organize and position themselves into forming a circle. The difficulty of the problem results from the fact that robots are anonymous, oblivious, unable to communicate directly, and also disoriented, i.e, share no knowledge of a common coordinate system. More precisely, the proposed algorithm ensures that the robots deterministically form a 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. In addition, thanks to the nature of the assumed model (i.e., oblivious robots), the algorithm is also self-stabilizing.

Keywords: Self-organizing Robots, Cooperative Mobile Robots, Distributed Algorithms, Mobile Computing, Circle Formation, Self-Stabilization.

1 Introduction

Mobile computing systems, devices, and applications are gradually becoming more and more pervasive, while the theoretical foundations are still not yet fully established. Current researches addressing the principles of mobile computing are mostly aimed at systems in which mobility occurs as an external factor, such as 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 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.

**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 [5].*
The newer algorithm presented in this paper was developed in Souissi's master thesis research [15]

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, and execute the same deterministic algorithm. Besides, robots 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 a self-stabilizing distributed 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 to move in formation [10]. The formation of geometrical patterns and flocking are both useful, for instance, for the self-positioning of mobile base stations in a mobile ad hoc network, e.g., as considered by Chatzigiannakis et al. [3]. From a conceptual standpoint, forming a circle provides a way for robots to agree on both a common origin and a common unit distance [17].

Motivation. Our principal motivation for studying the problem of circle formation is however different. We indeed intend to use this problem as a starting point for studying the role and strengths of several different communication models. For instance, the algorithm presented in this paper relies exclusively on the fact that robots can detect each others position, as is the case with vision (with unlimited range). It is now interesting to see whether replacing vision with other communication models (e.g., ad hoc networking with directional antennas) still allows for solving the circle formation problem. This question is however not addressed here and left for later studies.

Contribution. The paper decomposes the question 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. Interestingly, the first part of the problem is already sufficient to agree on an origin and unit distance.

The main contribution of this paper is to propose a distributed algorithm by which oblivious robots deterministically form a (possibly irregular) circle (part 1) in a finite number of steps, and asymptotically converge toward a situation in which they are positioned evenly on the boundary of this circle (part 2).

Although Suzuki and Yamashita [17] had previously given a solution in the case of *non-oblivious* robots, our algorithm assumes that robots are *oblivious*, that is, the algorithm does not require the robots to memorize past actions and observations (in other words, the robots are stateless). This difference is very significant because, as Suzuki and Yamashita [17] have pointed out, algorithms for oblivious robots are intrinsically self-stabilizing.[1] It follows that our algorithm is itself self-stabilizing.[2]

Related work. A vast amount of researches exists in the context of cooperative mobile robotics, but much research focus on the study of diverse heuristics, such as free market optimization (e.g., [6]) or swarm intelligence (e.g., [2, 13]). However, only few studies take the problem from a computational standpoint. 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.

Debest [4] 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 [16] proposed several algorithms for the formation of various geometrical patterns. They proposed a simple heuristic algorithm for the formation of an approximation of a circle in the limited visibility setting. Their solution does not reach always a desirable configuration, and sometimes

[1] Self-stabilization is the property of a system which, started in an arbitrary state, always converges toward a desired behavior [9].

[2] Our algorithm is self-stabilizing provided that no two robots have both the same initial position and the same local coordinate system. This restriction is necessary as two such robots are impossible to separate in a deterministic manner, as they may happen to be always activated simultaneously. However, while the second part of the problem requires the robots to be separated, this issue is irrelevant for solving the first part and thus reaching agreement on the origin.

it brings the robots to form a Reuleaux triangle.[3]

Later on, Suzuki and Yamashita [17] 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* (and thus self-stabilizing) solution was however left as an open question.

Défago and Konagaya [5], in a preliminary version of this paper, proposed an algorithm whereby oblivious robots deterministically form a circle. In the meantime, we have developed a simpler and more elegant algorithm, together with complete and rigorous proofs of correctness, rather than the proof sketches of the previous version.

Chatzigiannakis et al. [11] proposed a partial solution to the circle formation problem that tried to simplify the previous algorithm of Défago and Konagaya [5]. Unfortunately, their solution relies on a simplifying assumption that completely removes the difficulty of the problem (in particular the robots must not be located on the same radius). In contrast, our algorithm can cope with any initial configuration. Recently, Katreniak [12] proposed, in a different model, an algorithm that solves a slightly simpler problem, called biangular circle.[4] An other recent study on the circle formation was by Dieudonné et al. [7], in which they proposed a deterministic solution to the problem combined with the work of Katreniak [12]. Their solution works in the semi-synchronous model [17] for any number of robots except for $n = 4$,6 or 8. Finally, Dieudonné and Petit [8] proposed an algorithm to solve the circle formation problem for a *prime* number of robots, which is based on Lyndon Words. In contrast to previous works mentioned above, we solve the circle formation problem in general.

Structure of the paper. The rest of the paper is structured as follows. In Section 2, we introduce the system model and the terminology used in the paper. In Section 3, we describe our algorithm, and in Section 4, we prove its correctness. Finally, in Section 5, we conclude the paper.

2 System Model and Definitions

2.1 System Model

In this paper, we consider the system model of Suzuki and Yamashita [17], which is defined as follows. The system consists of a set of autonomous mobile robots $\mathcal{R} = \{r_1, \cdots, r_n\}$ 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. However, the robots share neither knowledge of the coordinate systems of the other robots nor of a global one.

During its observation, a robot obtains the position of all robots according to its own local coordinate system. In this paper, we consider the case when visibility is not limited and robots do not obstruct the view from each other.

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.[5] Thus, we assume that all robots occupy distinct locations initially, and let the algorithm ensure that it remains so.

The time is represented as an infinite sequence of discrete time instants $t_0, t_1, t_2, \ldots$, during which each robot can be either *active* or *inactive*. When a robot becomes active, it observes the environment,

[3]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 [14].

[4]In a biangular circle, the number of robots must be even.

[5]Consider two robots that happen to have the same coordinate system and that are always activated together. It is impossible to separate them deterministically. Allowing this result in that, the problem becomes trivially impossible to solve deterministically.

computes a new location, and moves toward it. This behavior constitutes its cycle of observing, computing, moving and being inactive. The sequence *Observe–Compute–Move* is called the *cycle* of a robot. The model states that the robots execute their cycles *atomically*.[6]

The activation of robots is determined by an activation schedule, unpredictable 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,[7] 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 δ_{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,[8] 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 φ that is executed by every robot r_i each time it becomes active. The arguments of φ 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 φ is the new destination for r_i.

2.2 Difficulty of Coordination

In the model described above, it is difficult for the robots to coordinate their actions. This is largely because of the dilemma caused by the unpredictability of the activation schedule. Let us illustrate this point by a simple example.

In a system with two robots r_1 and r_2 that initially occupy distinct positions, consider the problem of having them eventually move to the same location. We can see the dilemma that r_1 faces (and also r_2 by symmetry) by considering the naive solution that follows.

When it is activated, robot r_1, assuming that the other robot is inactive, moves directly to the position occupied by r_2. In this case, the problem is solved in one step if r_2 indeed remains stationary. However, if r_2 happens to be active simultaneously, it takes the same action as r_1, and hence moves to occupy the position that r_1 has just left. Consequently, if the activation schedule is such that the two robots are always activated at the same time, the system remains caught in a livelock with the two robots endlessly swapping positions.

2.3 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 1 (UNIFORM CIRCLE FORMATION) *Given a group of n robots $r_1, r_2, \ldots, r_n$ 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 more rigorously as follows.

[6] The complete rationale for this assumption is given by Suzuki and Yamashita [17].

[7] 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.

[8] By deterministic, we mean that any two independent executions of the algorithm with identical input values always yield the same output. In particular, this rules out the use of randomization.

Problem 2 (CIRCLE FORMATION) *Given a group of n robots $r_1, r_2, \ldots, r_n$ 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 1 cannot be solved deterministically with oblivious robots, we show that Problem 2 can. In fact, we show that our algorithm solves Problem 2 within a finite number of steps, and converges toward a uniform solution (Prob. 1).

2.4 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 [18]. 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 ρ_p and θ_p, where ρ_p is the length of the segment $\overline{op}$, and θ_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 π.

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).

3 Circle Formation for Oblivious Robots

3.1 Algorithm Intuition

Given the Suzuki and Yamashita [17] model (see Section 2.1) 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. 2) deterministically, and (2) converges toward a solution to the *Uniform Circle Formation* problem (Prob. 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 3.2). 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).[9]

3.2 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 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 2 *Let $P_c(t)$ be the set of robots on the boundary of 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 π, or else C would no longer the smallest circle enclosing all robots.

3.3 Algorithm Description

We now describe the algorithm in more details, and give a pseudo-code description (see Algorithm 1).[10] As already mentioned, the robots use the smallest circle enclosing all robots 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 C reach its boundary in a finite number of activations (Prob. 2), and that the robots located on the boundary converge to a situation where they are evenly spread on this boundary (Prob. 1). In fact, the algorithm can be seen as a combination of two algorithms that solve the two problems simultaneously.

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 C, based on the observed position of the robots, and changes its coordinate system to a polar one, with the origin located at point o; the center of 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. END.

[9] 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.

[10] 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.

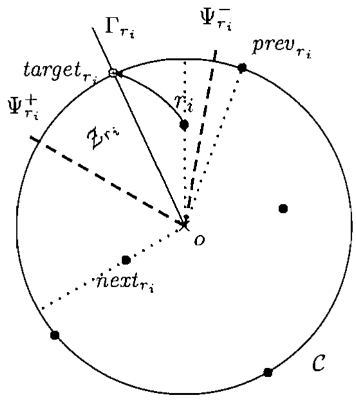

Figure 1. Principle of the algorithm.

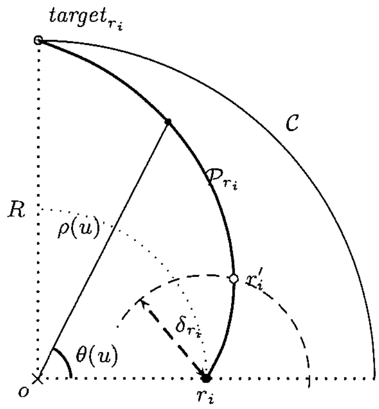

Figure 2. Parametric path $\mathcal{P}_{r_i}$ computed by robot r_i.

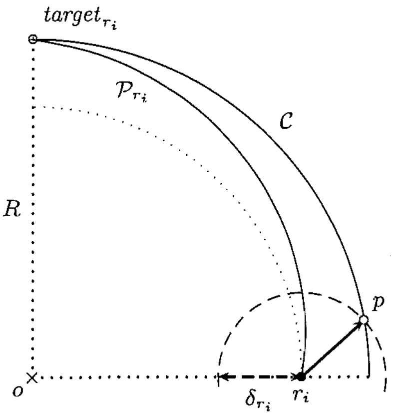

Figure 3. $target_{r_i}$ is out of reach, while C is not; r_i joins C at point p.

3. Otherwise, r_i locates two robots $prev_{r_i}$ and $next_{r_i}$, according to the description of the virtual ring in Sect. 2.4.

4. If $prev_{r_i}$, r_i, and $next_{r_i}$ are together *aligned with the origin*, then r_i does nothing. END.

5. If not, then r_i computes three rays starting from o, called $\Psi^-_{r_i}$, $\Psi^+_{r_i}$, and Γ_{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 $next_{r_i}$. Γ_{r_i} is the bisector of the angle formed by $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$.

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:

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

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 Γ_{r_i}, r_i computes a target location $target_{r_i}$, as the intersection of Γ_{r_i} with C. Notice that, by definition, $target_{r_i}$ is always located in $\mathcal{Z}_{r_i}$.

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 C, then it moves[11] to the reachable point on 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. END.

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 δ_{r_i}) along this path (see Fig. 2). END.

[11]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 C (see Lemma 10). 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 C in finite time due to the Zeno paradox.

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

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

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

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

- $\mathcal{C}$:= smallest circle enclosing all points in P;
- **if** (r_i = center of $\mathcal{C}(P)$) **then**
  - r_i moves to an arbitrary location by some radius ρ_{r_i} less than the minimum radius of all other robots;
- **else**
  - Compute $prev_{r_i}$ and $next_{r_i}$ (see Sect. 2.4)
  - **if** ($prev_{r_i}, r_i, next_{r_i}$) are *aligned with the origin* **then**
    - stay still;
  - **else**
    - $\alpha_{prev_{r_i}}$:= angular distance between r_i and $prev_{r_i}$ in clockwise orientation;
    - $\alpha_{next_{r_i}}$:= angular distance between r_i and $next_{r_i}$ in anticlockwise orientation;
    - $\Psi^-_{r_i}$:= bisector of the angle $\alpha_{prev_{r_i}}$;
    - $\Psi^+_{r_i}$:= bisector of the angle $\alpha_{next_{r_i}}$;
    - Γ_{r_i} := bisector of the angle formed by $\Psi^-_{r_i}$ and $\Psi^+_{r_i}$;
    - $target_{r_i}$:= $\Gamma_{r_i} \cap \mathcal{C}$;
    - Compute path $\mathcal{P}_{r_i}$ from r_i to $target_{r_i}$ (Eq. (2));
    - **if** $dist(r_i, \mathcal{C}) \leq \delta_{r_i}$ **then**
      - Move to $\mathcal{C}$;
    - **else**
      - Move along $\mathcal{P}_{r_i}$ toward $target_{r_i}$ by δ_{r_i};
    - **end if**
  - **end if**
- **end if**

4 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 1). Second, we prove that the smallest enclosing circle remains invariant (Theorem 2). Then, we show that all robots reach the boundary of the circle in finite time (Theorem 4). Finally, we prove that the algorithm converges toward a configuration wherein all robots are located at regular intervals on the circle (Theorem 5).

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

Lemma 1 *No robot ever moves beyond the boundary of the smallest circle enclosing all robots.*

Lemma 2 *All robots located on the boundary of the smallest enclosing circle remain on that boundary.*

4.1 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 angular position of robot $prev_{r_a}$ (see Fig. 4). Let θ_1, θ_2, and θ_3 denote the angles of robots r_a, r_b, and $next_{r_b}$, respectively. We also consider the bisectors

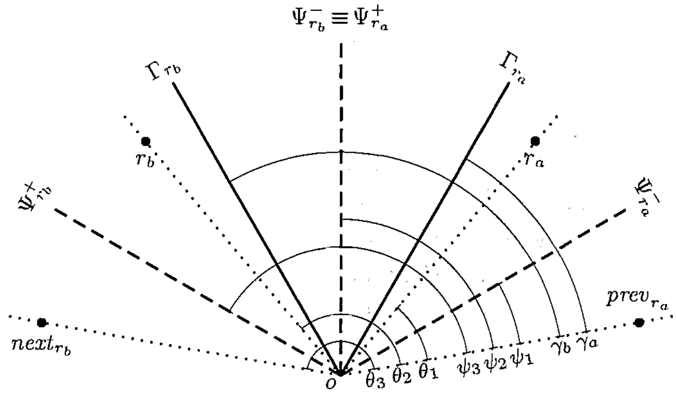

Figure 4. Invariance of virtual ring: consecutive robots r_a and r_b.

$\Psi^-_{r_a}$, $\Psi^+_{r_a}$, $\Psi^+_{r_b}$, used in the definition of the movement. Notice that $\Psi^-_{r_b} \equiv \Psi^+_{r_a}$ because $r_a = prev_{r_b}$. Let ψ_1, ψ_2, and ψ_3 denote the angles of $\Psi^-_{r_a}$, $\Psi^+_{r_a}$, and $\Psi^+_{r_b}$, respectively. Finally, we consider the two second-order bisectors Γ_{r_a} and Γ_{r_b}, and let γ_a and γ_b denote their respective angles. Remind that the respective targets of r_a and r_b are located on Γ_{r_a} and Γ_{r_b}.

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

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

Lemma 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, θ_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 $\theta_1 = \psi_2$.

$\square_{\text{Lemma 3}}$

Lemma 4 *There is no overlap between the exclusive zones of any two robots.*

PROOF. The proof is a generalization of Lemma 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 ρ_r, there can be no intersection with the zone of r_o. $\square_{\text{Lemma 4}}$

Theorem 1 *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 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, Γ_{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 1

4.2 Invariance of the Smallest Enclosing Circle

From Lemma 1, Lemma 2, Restriction 1-2 and Theorem 1, we obtain the following theorem:

Theorem 2 *The smallest enclosing circle C is invariant.*

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

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

1. *(2 robots)* The smallest enclosing circle $C(t)$ is defined by only two robots. Those robots cannot move by Restriction 1 and hence $C(t+1) = C(t)$.
2. *(3 robots; one quits the circle)* The smallest enclosing circle $C(t)$ is defined by three robots, one of which moves outside the boundary of $C(t)$. This is a contradiction with Lemma 1.
3. *(3 robots; two distinct points)* The smallest enclosing circle $C(t)$ is defined by three robots, two of which move to the same location. This is in contradiction with Theorem 1.
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 $C(t)$ 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 2, the movement of two consecutive robots cannot lead them further away from each other than π, regardless of their activation schedule.

When there are more than three robots on the boundary of $C(t)$, the situation can always be reduced to one of the four cases mentioned above. It follows that $C(t)$ and $C(t+1)$ cannot be different; a contradiction. □Theorem 2

The following lemma is obtained easily from the algorithm.

Lemma 5 *For any robot r_i, its radial position $\rho_{r_i}(t)$ is nondecreasing.*

Lemma 6 *There is a time since which no robot is on the center of 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, the radial position is nondecreasing, and thus no robot can be located at the center of $\mathcal{C}$ after time t. $\Box_{\text{Lemma 6}}$

4.3 Invariance of the Virtual Ring

Theorem 3 *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 4.1, 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 Γ_{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.
$$\begin{array}{ccccccccccccc} 0 & \leq & \psi_1 & \leq & \theta_1 & < & \psi_2 & < & \theta_2 & \leq & \psi_3 & \leq & \theta_3 \\ & & \parallel & & & & \parallel & & & & \parallel & & \\ & & \psi_1 & < & \gamma_a & < & \psi_2 & < & \gamma_b & < & \psi_3 & & \end{array}$$
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 2). 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.

$\Box_{\text{Theorem 3}}$

4.4 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 7 *The angle that separates a robot r_i from its target $target_{r_i}$ is at most $\frac{\pi}{4}$.*

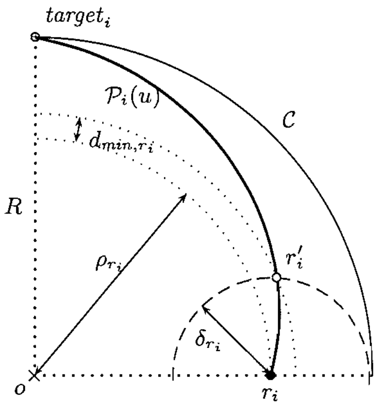

Figure 5. The minimum distance of progress of r_i toward the boundary of the circle is d_{min,r_i}.

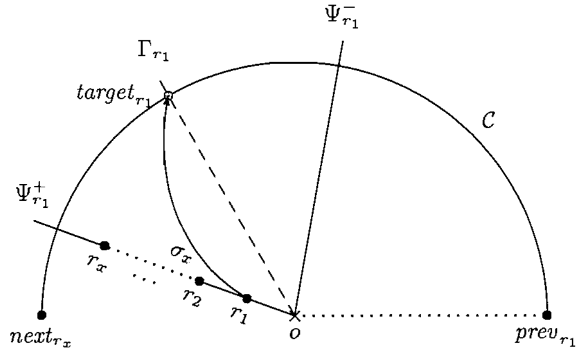

Figure 6. String of robots *aligned with the origin.*

PROOF. By Restriction 2, the maximum angular distance that can separate any two consecutive robots is π. 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 π.

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 $\Psi^+_{r_i}$, Γ_{r_i} is at angle $\frac{\pi}{4}$. Since $target_{r_i}$ is located on Γ_{r_i}, this proves the lemma. $\Box_{\text{Lemma 7}}$

Lemma 8 *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 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 7) 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 (δ_{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 ρ_{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. $\Box_{\text{Lemma 8}}$

Lemma 9 *By the algorithm, 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 $\sigma_x = r_1, \cdots, r_x$ 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 σ_x (see proof of Theorem 1), and then the rest of the proof is by induction on x, the number of robots at σ_x.
Basis: ($x = 1$). The lemma holds trivially.
Induction Step: Assume that the lemma holds for any string σ_y shorter than x ($y < x$), and let us prove that the lemma holds for a string σ_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 Γ_{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 δ_{r_1} is greater than zero, the destination r'_1 of r_1 cannot be aligned with the robots of σ, regardless of the test in line 16. Thus, after its move, r_1 no longer belongs to the string σ, thus decreasing its length by one. This proves the induction step. $\square_{\text{Lemma 9}}$

Lemma 10 *All robots located in the interior of C reach its circumference in finite time.*

PROOF. By Lemma 9, 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 8, at each activation step, a robot r_i, not located on the boundary of 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 C is bounded above by $\frac{R}{d_{min,r_i}}$. Thus, due to the fairness of the activation schedule, the boundary of C is reached in finite time and the lemma holds. $\square_{\text{Lemma 10}}$

Lemma 11 *The global predicate that all robots are located on the boundary of 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 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 C. Because r_i is also on C, the entire path $\mathcal{P}_{r_i}$ is located on C. Thus, r_i can only move to a location on the boundary of C. It follows that configuration c' is in C_{circle}. $\square_{\text{Lemma 11}}$

Theorem 4 *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 10), and this situation is stable (Lemma 11). $\square_{\text{Theorem 4}}$

4.5 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 1 *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 2 *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 12 *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:

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

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, $\alpha_{max}(t+1) \leq \alpha_{max}(t)$. $\Box_{\text{Lemma 12}}$

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

Lemma 13 *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. $\Box_{\text{Lemma 13}}$

Lemma 14 *The function $\Delta(t) = \alpha_{max}(t) - \alpha_{min}(t)$ is monotonically decreasing and converges to zero.*

PROOF. First of all, from Lemma 12, we can deduce that $\Delta(t)$ is nonincreasing. 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,

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

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,

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

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 12, 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 12, $\alpha_{max}(t+1) \leq \alpha_{max}(t)$, 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. □Lemma 14

Theorem 5 *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 13 and Lemma 14.

4.6 Discussion: Difference with Earlier Algorithm

Défago and Konagaya [5] used a different algorithm. In short, the earlier algorithm was a composition of two independent algorithms. The first one, solving Problem 2, relied also on the definition of exclusive movement zones. However, the zones were defined using the Voronoi cell[12] of each robot, and was executed by all robots until they all reached the boundary of $\mathcal{C}$. The second algorithm, converging toward Problem 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 $O(n)$, whereas computing the Voronoi diagram is normally done in $O(n \log n)$. Finally, since the two problems are being solved simultaneously, it is reasonable to assume that Algorithm 1 converges faster than its predecessor.

5 Conclusion

In this paper, we have presented a distributed algorithm whereby a team of oblivious mobile robots self-organize to form a circle. The algorithm allows the robots to deterministically form the circle within a finite number of activation steps, and asymptotically converges toward a uniform distribution of the

[12]The *Voronoi diagram* Voronoi(P) of a set of points $P = \{p_1, p_2, \ldots, p_n\}$ 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].

robots along the circumference of the circle. Our algorithm is arguably simple and also it has a low computation cost. Moreover, it is intrinsically self-stabilizing, due to the assumption that robots are oblivious. In particular, a self-stabilizing system is able to tolerate any number of transient faults and the state immediately after the occurrence of an error can be regarded as initial state.

For systems in which robots do have a memory, the algorithm can be further optimized by relying on the invariance of the smallest enclosing circle. Then, the self-stabilizing properties can still be preserved, provided that the validity of the smallest enclosing circle cached in memory is verified before each activation.

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.

References

[1] F. Aurenhammer. Voronoi diagrams–a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, Sept. 1991.

[2] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. In *Proc. of IEEE Transaction on Robotics and Automation*, volume 14, pages 926–939, 1998.

[3] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. On the average and worst-case efficiency of some new distributed communication and control algorithms for ad-hoc mobile networks. In *Proc. 1st ACM Int'l Workshop on Principles of Mobile Computing (POMC'01)*, pages 1–19, Newport, RI, USA, Aug. 2001.

[4] X. A. Debest. Remark about self-stabilizing systems. *Commun. ACM*, 38(2):115–117, Feb. 1995. Technical correspondence.

[5] X. Défago and A. Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *Proc. 2nd ACM Int'l Workshop on Principles of Mobile Computing (POMC'02)*, pages 97–104, Toulouse, France, Oct. 2002.

[6] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *Proc. of the 6th Int'l Conf. on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, Venice, Italy, July 2000.

[7] Y. Dieudonne, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. In *Proc. of 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'06) LNCS.*, volume 4280, pages 262–275, USA, 2006.

[8] Y. Dieudonne and F. Petit. Circle formation of weak robots and lyndon words. In *Information Processing Letters*, volume 104, pages 156–162, 2007.

[9] S. Dolev. *Self-Stabilization.* MIT Press, 2000.

[10] V. Gervasi and G. Prencipe. Flocking by a set of autonomous mobile robots. Technical Report TR-01-24, Dipartimento di Informatica, Università di Pisa, Italy, Oct. 2001.

[11] I.Chatzigiannakis, M. Markou, and S. Nikoletseas. Distributed circle formation for anonymous oblivious robots. In *Proc. of the 3th Workshop on Efficient and Experimental Algorithms.(WEA-3)*, pages 159–174, Angra dos Reis, Rio de Janeiro, Brazil, May 2004.

[12] B. Katreniak. Biangular circle formation by asynchronous mobile robots. In *Proc. 12th Intl. Colloquium on Structural Information and Communication Complexity (SIROCCO'05)*, volume 3499, pages 185–199, France, 2005.

[13] M. J. B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995, Aug. 2000.

[14] W. Research. Mathworld. In *URL. http://mathworld.wolfram.com.*

[15] S. Souissi. On distributed cooperative mobile robotics: Decomposition of basic problems and study of a self-stabilizing circle formation algorithm. Master's thesis, Japan Advanced Institute of Science and Technology, School of Information Science, Sept. 2004.

[16] K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal on Robotics Systems*, 3(13):127–139, Mar. 1996.

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

[18] W.Emo. Smallest enclosing disks (balls and ellipsoids). (555):359–370, 1991.