Circle Formation of Weak Mobile Robots

YOANN DIEUDONNÉ, OUIDDAD LABBANI-IGBIDA, and FRANCK PETIT
MIS Laboratory, Université de Picardie Jules Verne

We consider distributed systems made of *weak mobile* robots, that is, mobile devices, equipped with sensors, that are *anonymous*, *autonomous*, *disoriented*, and *oblivious*. The *Circle Formation Problem* (CFP) consists of the design of a protocol insuring that, starting from an initial arbitrary configuration where no two robots are at the same position, all the robots eventually form a *regular n-gon*—the robots take place on the circumference of a circle C with equal spacing between any two adjacent robots on C.

CFP is known to be unsolvable by arranging the robots evenly along the circumference of a circle C without leaving C—that is, starting from a configuration where the robots are on the boundary of C. We circumvent this impossibility result by designing a scheme based on *concentric circles*. This is the first scheme that deterministically solves CFP. We present our method with two different implementations working in the semi-synchronous system (SSM) for any number $n \geq 5$ of robots.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Algorithms

Additional Key Words and Phrases: Distributed computing, formation of geometric patterns, mobile robot networks, self-deployment

ACM Reference Format:

Dieudonné, Y., Labbani-Igbida, O., and Petit, F. 2008. Circle formation of weak mobile robots. ACM Trans. Autonom. Adapt. Syst. 3, 4, Article 16 (November 2008), 20 pages. DOI = 10.1145/1452001.1452006 http://doi.acm.org/10.1145/1452001.1452006

1. INTRODUCTION

Consider a distributed system where the computing units are *weak mobile robots* (*sensors* or *agents*), that is, devices equipped with sensors and designed to move in a two-dimensional plane. By weak, we mean that the robots are *anonymous*, *autonomous*, *disoriented*, and *oblivious*, that is, devoid of (1) any local parameter (such as an identity) allowing one to differentiate any of them,

Early versions of this work were published in Dieudonné et al. [2006] and Dieudonné and Petit [2007c].

Author's address: F. Petit, Université de Picardie Jules Verne, Amiens, France; email: franck.petit@u-picardie.fr.

(2) any central coordination mechanism or scheduler, (3) any common coordinate mechanism or common sense of direction, and (4) any way to remember any previous observation nor computation performed in any previous step. Furthermore, all the robots follow the same program (*uniform* or *homogeneous*), and there is no kind of explicit communication medium. The robots implicitly "communicate" by observing the position of the others robots in the plane, and by executing a part of their program accordingly.

The motivation behind such a weak and unrealistic model is the study of the minimal level of ability the robots are required to have in the accomplishment of some basic cooperative tasks in a deterministic way. So far, the studied tasks are geometric problems, such as *pattern formation*, *line formation*, *gathering*, and *circle formation* [Suzuki and Yamashita 1999; Flocchini et al. 2000; Prencipe 2002; Prencipe and Santoro 2006]. Basically, every protocol for one of these problems aims to be *quiescent*, that is, in every execution, all the robots are eventually *motionless* in the desired configuration, and remain motionless thereafter. Note that quiescence is a very desirable property in order to be able to sequentially combine algorithms together to achieve a global task. As a matter of fact, each of these algorithms is often easier to solve than the whole task. Consider for instance, a solution $\mathcal{S}$ for a given task $\mathcal{T}$ working in successive phases. Assume that the second phase of $\mathcal{S}$ requires a distinguished robot as a *leader* (necessary and sufficient conditions to deterministically elect a leader are given in Dieudonné and Petit [2007b]). So, without any extra assumption on the model, the first phase must be a leader election phase. The second phase can start only when the leader robot is unambiguously distinguishable by every robot. Thus, such a first leader election phase must be quiescent in order to enable the second phase.

The *Circle Formation Problem* (CFP) belongs to the class of pattern formations. It consists of the design of a protocol insuring that, starting from an initial arbitrary configuration where no two robots are at the same position, all the robots eventually form a circle with equal spacing between any two adjacent robots (this problem is sometimes referred to as the *uniform circle formation Problem* [Défago and Konagaya 2002]). In other words, the system is expected to converge to a configuration where the robots are quiescent, forming a *regular n-gon*.

Related Works. An informal CFP algorithm was first given in Debest [1995]. Several CFP protocols were subsequently proposed. In Sugihara and Suzuki [1996], an algorithm based on heuristics is proposed for the formation of a circle approximation. A CFP protocol is given in Suzuki and Yamashita [1999] for nonoblivious robots with an unbounded memory. Three deterministic algorithms are provided in Défago and Konagaya [2002], Chatzigiannakis et al. [2004], and Défago and Souissi [2008]. The former is based on an elegant Voronoi Diagram construction to lead the robots on the circumference of the circle C. The second work makes an extra assumption on the initial position of the robots. The latter improves the space complexity of Défago and Konagaya [2002] by avoiding the Voronoi construction to lead the robots on C. All of these solutions work in the semi-synchronous model (SSM) [Suzuki and Yamashita 1999] in which the cycles of all the robots are synchronized and their actions are atomic. Furthermore, they all guaranteed only asymptotical convergence toward a configuration in which the robots are uniformly distributed on the boundary of a circle. In other words, these solutions are not quiescent—the robots move infinitely often and never reach the desired final configuration.

The first (quiescent) solution leading n robots in a regular n-gon in finite time is proposed in Katreniak [2005]. The proposed protocol works in CORDA [Prencipe 2001], a fully asynchronous model where, by contrast to SSM, the robot cycles are not required to be synchronized. CORDA being weaker than SSM, solutions designed in CORDA also work in SSM [Prencipe 2002]. The solution in Katreniak [2005] works if $n \geq 5$ only. Moreover, if n is even, the robots may form a *biangular circle* in the final configuration, that is, the distance between two adjacent robots is alternatively either α or β.

A common strategy in order to solve a nontrivial problem such as CFP is to combine subproblems that are easier to solve. The first subproblem consists in placing the robots along the boundary of a circle C, without considering their relative positions. The second subproblem, called the *uniform transformation problem* (UTP), consists in starting from there, and arranging the robots, without leaving the circle C, evenly along the boundary of C. In Défago and Konagaya [2002], the authors present an algorithm for the second subproblem that converges toward a homogeneous distribution of robots, but it does not terminate deterministically. In Flocchini et al. [2006], the authors propose two deterministic solutions for UTP by assuming an extra assumption: the robots agree on a clockwise direction of the circle. The first solution solves UTP by assuming that the desired final distance d between two robots is known to them. The second solution does not require that the robots know d. This solution leads the system in an ϵ-approximate regular n-gon, that is, the actual distance between the robots is eventually equal to d' such that $|d - d'| \leq \epsilon$ for a given $\epsilon > 0$. In Dieudonné and Petit [2007a], properties on Lyndon words are used to achieve a (quiescent) n-gon in SSM. However, the solution in Dieudonné and Petit [2007a] works for a prime number of robots only. Also, it assumes that every robot reaches its destination atomically—that is, no robot stops before reaching its destination. There in no doubt that UTP is definitively the “hard” part of CFP to be solved. In fact, in Défago and Konagaya [2002], the authors conjecture that there is no deterministic solution solving UTP in finite time in SSM. The validity of the conjecture is proven in Flocchini et al. [2006].

Contribution. In this article, we circumvent the previously described impossibility result by designing a scheme based on *concentric circles*. Combined with the solution in Katreniak [2005], the proposed scheme is quiescent and deterministic. Our solution works for any number $n \geq 5$ of robots. We shortly present a first attempt that works, provided that $n \neq 6$ and $n \neq 8$. This first attempt assumes that no robot can stop before reaching its destination. Next, we develop another approach dealing with the two previously mentioned drawbacks. It uses an original technique, based on *swing words*, also introduced in this article. We apply intrinsic properties of swing words over the *convex hull* formed by the robots.

This second approach does not assume that every robot is required to reach its computed destination in one step. Furthermore, we show that, starting from a biangular configuration, this second approach solves CFP in CORDA. To the best of our knowledge, it is the first CFP protocol for SSM that is compatible with CORDA.

Outline of the Article. In the next section (Section 2), we describe the distributed systems and the model we consider in this article. In the same section, we present the problem considered in this article. Both techniques to achieve the CFP are shown in Section 3. Finally, we conclude in Section 4.

2. PRELIMINARIES

In this section, we introduce the distributed system, some basic definitions, and the problem considered in this article.

2.1 Distributed Model

We adopt the model introduced in Suzuki and Yamashita [1996], in the remainder referred to as *SSM*. The *distributed system* considered in this article consists of n robots $r_1, r_2, \cdots, r_n$—the subscripts $1, \ldots, n$ are used for notational purposes only. Each robot r_i, viewed as a point in the Euclidean plane, moves on this two-dimensional space unbounded and devoid of any landmark. When no ambiguity arises, r_i also denotes the point in the plane occupied by that robot. Any robot can observe, compute, and move with an infinite decimal precision. The robots are equipped with sensors allowing them to detect the instantaneous positions of the other robots in the plane. Each robot has its own local coordinate system and measure unit. The robots do not agree on the orientation of the axes of their local coordinate system, nor on the measure unit. They are *uniform* and *anonymous*, that is, they all have the same program, using no local parameter (such as an identity) allowing one to differentiate any of them. They communicate only by observing the positions of the others and they are *oblivious*, that is, none of them can remember any previous observation nor computation performed in any previous step.

Time is represented as an infinite sequence of time instants $t_0, t_1, \ldots, t_j, \ldots$ Let $P(t_j)$ be the multiset of the positions in the plane occupied by the n robots at time t_j ($j \geq 0$). For every t_j, $P(t_j)$ is called the *configuration* of the distributed system in t_j. $P(t_j)$, expressed in the local coordinate system of any robot r_i, is called a *view*, denoted $v_i(t_j)$. At each time instant t_j ($j \geq 0$), each robot r_i is either *active* or *inactive*. The former means that, during the computation step (t_j, t_{j+1}), using a given algorithm, r_i computes in its local coordinate system a position $p_i(t_{j+1})$ depending only on the system configuration at t_j, and moves towards $p_i(t_{j+1})$—$p_i(t_{j+1})$ can be equal to $p_i(t_j)$, making the location of r_i unchanged. In other words, if a robot r_i is activated (at t_j), then r_i observes, computes, and moves in one atomic step (t_j, t_{j+1}). In every single activation, the distance traveling by any robot r is bounded by an arbitrary $\sigma_r > 0$. So, if the destination point computed by r is farther than σ_r, then r moves toward a point of at most σ_r.

The concurrent activation of robots is modeled by the interleaving model in which the robot activations are driven by a *fair scheduler*. At each instant t_j ($j \geq 0$), the scheduler arbitrarily activates a (nonempty) set of robots. Fairness means that every robot is infinitely often activated by the scheduler.

2.2 The Circle Formation Problem

In this article, the term *circle* refers to a circle having a radius strictly greater than zero. Consider a configuration at time t_k ($k \geq 0$) in which the positions of the n robots are located at distinct positions on the circumference of a circle C. At time t_k, the *successor* r_j, $j \in 1 \dots n$, of any robot r_i, $i \in 1 \dots n$ and $i \neq j$, is the single robot such that no robot exists between r_i and r_j on C in the clockwise direction. Given a robot r_i and its successor r_j on C centered in O:

(1) r_i is said to be the *predecessor* of r_j;
(2) r_i and r_j are said to be *adjacent*;
(3) $\widehat{r_i O r_j}$ denotes the angle centered in O and with sides the half-lines $[O, r_i)$ and $[O, r_j)$ such that no other robot (than r_i and r_j) is on C inside $\widehat{r_i O r_j}$.

Definition 2.1. *Regular n-gon*. A cohort of n robots ($n \geq 2$) forms (or is arranged in) a regular n-gon if the robots take place on the circumference of a circle C centered in O such that, for every pair r_i, r_j of robots, if r_j is the successor of r_i on C, then $\widehat{r_i O r_j} = \delta$, where $\delta = \frac{2\pi}{n}$. The angle δ is called the *characteristic angle* of the n-gon.

The problem considered in this article, called CFP (*Circle Formation Problem*), consists of the design of a distributed protocol that arranges a group of n ($n > 2$) mobile robots with initial distinct positions into a *regular n-gon* in finite time.

3. CIRCLE FORMATION PROTOCOL

We first need the following definition:

Definition 3.1. *UTP Algorithm*. A distributed algorithm A solves the uniform transformation problem (UTP) if and only if, starting from a configuration where the robots are arbitrarily located along the circumference of a circle C (no two robots being located at the same position), (i) none of the robots leaves the circumference of C during the execution of A and, (ii) all the robots eventually form a regular n-gon.

THEOREM 3.2. [Flocchini et al. 2006]. *There exists no deterministic algorithm A solving UTP in SSM for an arbitrary number of robots.*

In this section, we present our method to deterministically circumvent Theorem 3.2. It is based on concentric cycles defined as follows:

Definition 3.3. *Concentric Circles*. Let C_1 and C_2 be two circles having their radius (strictly) greater than 0. C_1 and C_2 are said to be *concentric* if they share the same center but their radii are different.

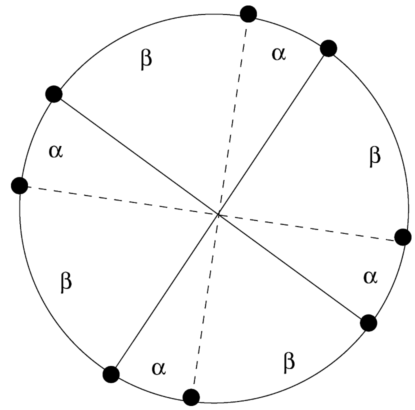


Fig. 1. An example showing a strict biangular circle ($\alpha \neq \beta$).

The proposed scheme uses the protocol presented in Katreniak [2005] as a subroutine. The protocol in Katreniak [2005] leads a cohort of n robots from an arbitrary into a *biangular configuration*, formally defined as follows:

Definition 3.4. *Biangular Circle.* A cohort of n robots ($n \geq 2$) forms (or is arranged in) a biangular circle if the robots take place on the circumference of a circle C centered in O and there exist two nonzero angles α, β such that for every pair r_i, r_j of robots, if r_j is the successor of r_i on C, then $\widehat{r_i O r_j} \in \{\alpha, \beta\}$ and α and β alternate in the clockwise direction.

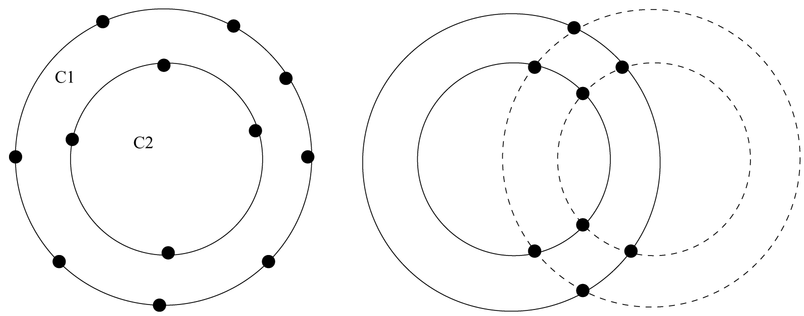
In the remainder of the article, without loss of generality, we assume that $\alpha \leq \beta$. Obviously, if $\alpha = \beta$ then, for any n value, the n robots form a regular n-gon. If $\alpha \neq \beta$, then n must be even ($n = 2p,\ p > 1$). In that case, the biangular circle is called a *strict* biangular circle—refer to Figure 1.

The solution proposed in Katreniak [2005] requires $n \geq 5$. In the remainder of this section, we assume $n \geq 5$, and refer to the protocol in Katreniak [2005] as Procedure $\langle A \rightsquigarrow B \rangle$, which stands for "from an *Arbitrary* configuration to a *Biangular* configuration."

THEOREM 3.5. [Prencipe 2002]. *Any algorithm that correctly solves a problem P in Corda, correctly solves P in SSM.*

This result means that Procedure $\langle A \rightsquigarrow B \rangle$ can be used in *SSM*. Obviously, Procedure $\langle A \rightsquigarrow B \rangle$ trivially solves the CFP if the number of robots n is odd. So, to solve CFP for any number of robots, it remains to deal with a system in a strict biangular configuration when n is even.

In the next subsection, we informally present our first attempt. We show why it assumes that no robot can stop before reaching its destination. This assumption means that, for every robot r, no bound exists on σ_r, the distance traveled by r. We also show that this first solution also requires $n \neq 6$ and $n \neq 8$ ($n \geq 5$). Next, we present our second attempt and main result. The protocol is presented with its correctness. It works for any number $n \geq 5$ of robots. Moreover, for every robot r, no assumption is made on σ_r, except $\sigma_r > 0$.



(a) An example of a concentric configuration with $n = 12$.

(b) An example showing that the pair of concentric circles may not be unique with $n \leq 8$.

Fig. 2. Examples of concentric configurations.

3.1 First Attempt

Without loss of generality, in the sequel, given a pair (C_1, C_2) of concentric circles, C_1 (respectively, C_2) indicates the circle with the greatest (respectively, smallest) radius.

Definition 3.6. *Concentric Configuration.* The system is said to be in a concentric configuration if there exists a pair of concentric circles (C_1, C_2) and a partition of the n robots into two subsets A and B such that every robot of A (respectively B) is located on C_1 (respectively, C_2)—refer to Figure 2, Case (a).

Remark 3.7. If $n \leq 8$, then the pair (C_1, C_2) may not be unique.

An example illustrating Remark 3.7 is given in Figure 2, Case (b).

LEMMA 3.8. *If the system is in a concentric configuration and $n \geq 9$, then there exists a single pair (C_1, C_2).*

PROOF. From Remark 3.7, if there exists a single pair (C_1, C_2), then n must be greater than or equal to 9. Since two distinct circles may share at most two points, 2 distinct pairs of concentric circle may share at most 8 points. □

Lemma 3.8 explains why our first attempt works for any $n \geq 9$ only. Given a concentric configuration and a robot r, $proj(r)$ denotes the projection of r on C_1, that is, the intersection between the half-line $[c, r)$ and C_1, where c is the center of (C_1, C_2). Obviously, if r is located on C_1, then $proj(r) = r$. We denote by Π the projection set of the n robots. In a concentric configuration, if $|\Pi| = n$, then the radii passing through the robots on C_1 split up the disk bounded by C_1 into *sectors*. In other words, we consider that if the robots are in a concentric configuration, then the disk can be divided into sectors iff $|\Pi| = n$ (refer to Figure 3), otherwise the configuration is considered as an arbitrary configuration.

A cohort of k robots $(0 < k \leq n)$ forms a *regular* (k, n)*-gon* if their positions coincide with a regular n-gon, where $n - k$ robots are missing. An example of a (k, n)-gon is given in Figure 4.

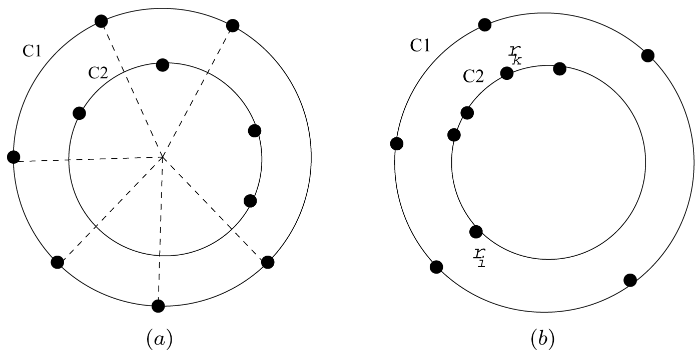
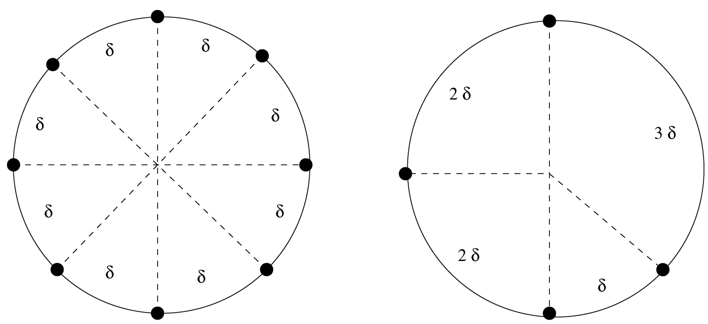


Fig. 3. The concentric configuration shown in Case (a) is split up into sectors ($|\Pi| = n$). The one in Case (b) is not because some robots on $C1$ are located on the projections of r_i and r_k ($|\Pi| = n - 2$).



(a) A 8-gon ($\delta = \frac{\pi}{4}$).

(b) A $(4, 8)$-gon obtained removing 4 robots from the corresponding 8-gon.

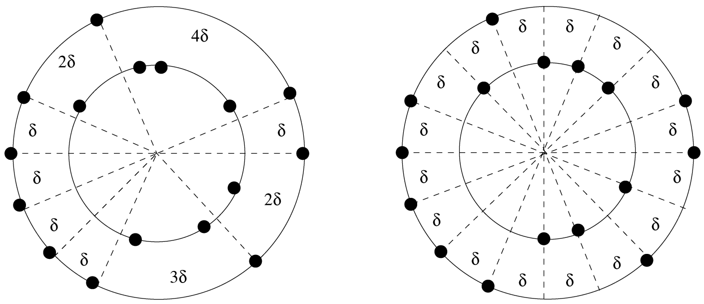
Fig. 4. An example showing a (k, n)-gon.

A cohort of n robots ($n \geq 9$) forms an (*arbitrary*) *quasi n-gon* if the three following conditions hold:

(1) The robots form a concentric configuration divided into sectors;
(2) The robots on C_1 form a regular (k, n)-gon;
(3) In each sector, if p robots are missing on C_1 to form a regular n-gon, then p robots are located on C_2 in the same sector.

A quasi n-gon is said to be *aligned* if Π coincides with a regular n-gon. Two quasi n-gon, are shown in Figure 5, the former is arbitrary, the latter is aligned.

We are now ready to informally present the overall scheme of this first attempt. It is mainly based on the particular configurations presented previously. Starting from an arbitrary configuration, by the repeated execution of Procedure $\langle A \rightsquigarrow B \rangle$, the system eventually forms a biangular circle. If n is odd, then the robots form a regular n-gon and the system is done. Otherwise (n is even), the robots form either a regular n-gon—again, the system is done—or a strict biangular circle. Starting from this latter case, the robots try to from a regular n-gon in one step. Either they move synchronously, and succeed, or they form a quasi n-gon. If the robots achieve a quasi n-gon, they aim to eventually



(a) An arbitrary quasi n-gon. (b) An aligned quasi n-gon.

Fig. 5. Two quasi n-gons with $n = 16$.

form an aligned quasi n-gon. Starting from an aligned quasi n-gon, the robots eventually form a regular n-gon. We discuss the details of each case as follows.

Starting from a strict biangular circle. In such a configuration, every active robot r_i achieves the following scheme:

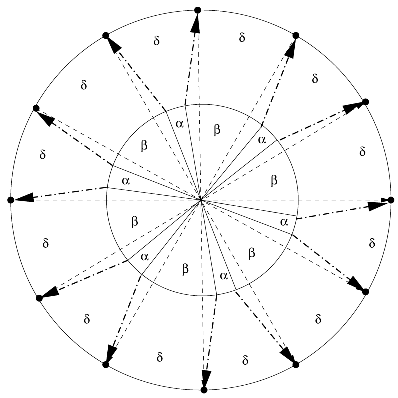
(1) Robot r_i computes the concentric circle C' whose the radius is twice the radius of the strict biangular circle C;
(2) Robot r_i considers its neighbor $r_{i'}$ such that $\widehat{r_i O r_{i'}} = \alpha$ and r_i moves away from r_i' to the position $p_i(t_{j+1})$ on C' with an angle equal to $\frac{\pi}{n} - \frac{\alpha}{2}$. More precisely, $(p_i(\widehat{t_{j+1})Op}_i(t_j)) = \frac{\pi}{n} - \frac{\alpha}{2}$ and $(p_i(\widehat{t_{j+1})Op}_{i'}(t_j)) = \frac{\pi}{n} + \frac{\alpha}{2}$ —refer to Figure 6.

Let us consider two possible behaviors depending on the synchrony of the robots.

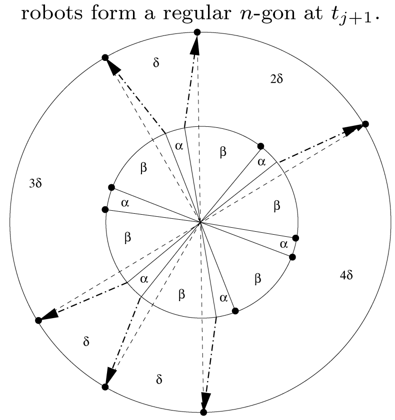
(1) Assume that every robot in the strict biangular circle is active at time t_j. In that case, since each robot reaches its destination in one atomic step, at t_{j+1}, the robots form a regular n-gon—refer to Case (a) in Figure 6. Note that (1) the trajectories of the robots do not cross between them, and (2) all the angles α (respectively, β) increase up (respectively, decrease down) to $\frac{2\pi}{n}$.
(2) Assume that some robots, in the strict biangular circle, are not active at time t_j. In that case, only a subset of robots moves toward C' from t_j to t_{j+1}. Then, the robots form a quasi n-gon at time t_{j+1}—refer to Case (b) in Figure 6.

Starting from an arbitrary quasi n-gon. Denote a quasi n-gon by the corresponding pair of concentric circles (C_1, C_2)—the radius of C_1 is greater than the radius of C_2. The idea is to change the arbitrary quasi n-gon into an aligned one by arranging the robots on the smallest concentric circle, C_2, in each sector—refer to Figure 5.

We define two quasi n-gons (C_1, C_2) and (C_1', C_2') to be *equivalent* if $C_1 = C_1'$, $C_2 = C_2'$ and the positions of the robots on C_1 and C_1' are the same ones. In other



(a) If all the robots are active at t_j, then the robots form a regular n-gon at t_{j+1}.



(b) If some robots are inactive at t_j, then the robots form a quasi n-gon at t_{j+1}.

Fig. 6. An example showing the principle of the algorithm starting from a biangular circle.

words, the only allowed possible difference between two equivalent quasi n-gons (C_1, C_2) and (C_1', C_2') is different positions of robots between C_2 and C_2' in each sector.

Let S be a sector. Let $p_1, \ldots, p_s$ $(s \leq n-1)$ be the final positions on C_2 in S in order to form the aligned quasi n-gon. Let B_1, B_2 be the two points located on C_2 at the boundaries of S. Of course, if only one robot is located on C_1 (i.e., there exists only one sector), then $B_1 = B_2$ and $s = n-1$. For each $i \in 1 \ldots s$, p_i denotes the point on C_2 in S such that $\widehat{B_1 O p_i} = \frac{2i\pi}{n}$, $p_i \neq B_1$, and $p_i \neq B_2$. For each $i \in 1 \ldots s$, p_i is the point on C_2 in S such that $\exists k > 0, \widehat{B_1 O p_i} = \frac{2k\pi}{n}$, $p_i \neq B_1$ and $p_i \neq B_2$. While the distributed system remains in an equivalent quasi n-gon, all the final positions remain unchanged for every robot. A final position p_i, $i \in 1 \ldots s$, is said to be *free* if no robot takes place at p_i. Similarly, a robot r_i on C_2 in S is called a free robot if its current position does not belong to $\{p_1, \ldots, p_s\}$.

Clearly, every active robot in S computes the same set, PS, of final positions. Let FRS be the set of free robots. Basically, if $FRS = \emptyset$, then every robot occupies a final position in S. Otherwise ($FRS \neq \emptyset$), let EFR be the set of robots in S that are the closest free robots to B_1 and B_2. Clearly, the number of robots in EFR is at most equal to two, one for each point B_1 and B_2. Note that it can be equal to one when there is only one free robot, that is, when only one robot in S did not reach the last free position. There are two cases:

(1) Assume that EFR contains a pair of robots $\{r_i, r_{i'}\}$ ($r_i \neq r_{i'}$). Then, without loss of generality, assume that r_i (respectively, $r_{i'}$) is the closest robot to B_1 (respectively, B_2). Then, r_i (respectively, $r_{i'}$) moves to the closest free position among $\{p_1, \ldots, p_s\}$ to B_1 (respectively, B_2).
(2) Assume that EFR contains only one robot r_i, then r_i moves to the single free remaining position p_i in its sector.

Starting from an aligned quasi n-gon. In such a configuration, we are almost done: each robot on C_2 moves toward its projection on C_1, whereas any robot on C_1 remains at the same position. Once all the robots belong to C_1, the system forms a regular n-gon.

3.2 Second Attempt

Clearly, the method presented in the previous subsection is based on the ability of a robot to distinguish whether the configuration is concentric (Definition 3.6) or not. Moreover, the robots need to move from a concentric configuration to another. This explains why the robots need to reach their positions in one step. We now present our second method, which is still based on concentric circles, but in which the robots do not have to reach their expected positions in one step. We first recall some basic definitions on words [Lothaire 1983]. Next, we introduce the swing words. Next, we provide basic geometric definitions, followed by particular configurations of the system built using properties of swing words. The particular configurations are used in the design and the proofs of the protocol. The protocol is then formally presented, with the correctness proof. Finally, we show that, starting from a biangular configuration, the proposed protocol works in CORDA.

3.2.1 *Swing words.* Let an ordered nonempty alphabet A be a finite set of letters. Denote by $\prec$ an order on A. A nonempty *word* w over A is a finite sequence of letters $a_0, \ldots, a_i, \ldots, a_{l-1}$, $l > 0$. The *concatenation* of two words u and v, denoted $u \circ v$ or simply uv, is equal to the word $a_0, \ldots, a_i, \ldots, a_{k-1}, b_0, \ldots, b_j, \ldots, b_{l-1}$ such that $u = a_0, \ldots, a_i, \ldots, a_{k-1}$ and $v = b_1, \ldots, b_j, \ldots, b_{l-1}$. Let ϵ be the *empty word* such that, for every word w, $w\epsilon = \epsilon w = w$. The *length* of a word w, denoted by $|w|$, is equal to the number of letters of w—$|\epsilon| = 0$. The *mirror word* of a word $w = a_0, \ldots, a_{l-1}$, denoted by $\overline{w}$, is equal to the word $a_{l-1}, \ldots, a_0$—$\overline{\epsilon} = \epsilon$.

A word u is *lexicographically* smaller than or equal to a word v, denoted $u \preceq v$, if and only if there exists either a word w such that $v = uw$ or three words r, s, t and two letters a, b such that $u = ras$, $v = rbt$, and $a \prec b$.

Let k and j be two positive integers. The *kth power* of a word w is the word denoted s^k such that $s^0 = \epsilon$, and $s^k = s^{k-1}s$. The *jth rotation* of a word w, denoted $R_j(w)$, is defined by:

$$R_j(w) \stackrel{\text{def}}{=} \begin{cases} \epsilon & \text{if } w = \epsilon \\ a_j, \ldots, a_{l-1}, a_0, \ldots, a_{j-1} & \text{if } w = a_0, \ldots, a_{l-1}, 0 \leq j \leq l-1). \end{cases}$$

Let $u = a_0 a_1 \ldots a_{l-1}$ ($l \geq 2$) be a finite word over A. Denote by Λ_u the subset of words $v = b_0 b_1 \ldots b_{l'-1}$ over $\{0, 1\}$ such that (1) $l' = l$, and (2) for every $i \in 0 \ldots l' - 1$:

$$b_i = \begin{cases} 0 & \text{if } a_{i \bmod l} \leq a_{(i+1) \bmod l} \\ 1 & \text{if } a_{i \bmod l} \geq a_{(i+1) \bmod l}. \end{cases}$$

Remark that the words in Λ_u are built on the cyclic representation of u. Let us consider the two following examples:

Example 1. Assume that $A = \{1, 2\}$ $(1 < 2)$. Then, $\Lambda_{11} = \Lambda_{22} = \{00, 01, 10, 11\}$, $\Lambda_{12} = \{01\}$, $\Lambda_{112} = \{001, 101\}$, $\Lambda_{1112} = \{0001, 1001, 0101, 1101\}$, and $\Lambda_{1221} = \{0110, 0111, 0010, 0011\}$. For instance, since both $1 \geq 1$ and $1 \leq 1$ are true, $\Lambda_{112} = \{001, 101\}$ because both $a_0 \leq a_1 \leq a_2 \geq a_0$ and $a_0 \geq a_1 \leq a_2 \geq a_0$ are true.

Example 2. Assume that $A = \{1, 2, 3\}$ $(1 < 2 < 3)$. Then: $\Lambda_{11231} = \{00010, 00011, 10010, 10011\}$, and $\Lambda_{311122} = \{101000, 100000, 110000, 111000, 101010, 100010, 110010, 111010\}$.

We are now ready to introduce the word class we are using in our method. Informally, a finite nonempty word w over an ordered alphabet A ($|A| \geq 2$) is a *swing* word if and only if the five following conditions are true:

1. The length of w is even;
2. The word w contains at least two different letters;
3. Each odd letter of w is greater than or equal to (respectively, lower than or equal to) its following letter;
4. Each even letter of w but the last one is lower than or equal to (respectively, greater than or equal to) its following letter;
5. The last letter of w is lower than or equal to (respectively, greater than or equal to) the first letter of w.

The formal definition is as follows:

Definition 3.9. *Swing word*. A finite nonempty word $w = a_0 a_1 \ldots a_{l-1}$ ($l \geq 1$) made over A is a *swing* word if and only if the following two conditions are true: (1) $w \neq a_0^l$, and (2) there exists $u \in \Lambda_w$ such that $u \in \{(01)^{\frac{l}{2}}, (10)^{\frac{l}{2}}\}$—$u$ is called an *associate swing word* of w.

For instance, in Example 1 (respectively, Example 2), 1112 (respectively, 311122) is a swing word—$0101 \in \Lambda_{1112}$ (respectively, $101010 \in \Lambda_{311122}$). Note that, even if Λ_{11} and Λ_{22} contain 01, both 11 and 22 are not swing words because they are equal to 1^2 and 2^2, respectively.

The following lemma directly follows from Definition 3.9:

LEMMA 3.10. *If a word $w = a_0 a_1 \ldots a_{l-1}$ $(l \geq 1)$ is a swing word, then:*
1. $l = 2p$ $(p \geq 1)$;
2. $\overline{w}$ *is a swing word;*
3. *For every* $j \in 0 \ldots l-1$, $R_j(w)$ *is a swing word.*

LEMMA 3.11. *If a word $w = a_0 a_1 \ldots a_{l-1}$ $(l \geq 1)$ is a swing word, then w has a unique associate swing word.*

PROOF. Let Λ_w^S be the subset of Λ_w such that $u \in \{(01)^{\frac{l}{2}}, (10)^{\frac{l}{2}}\}$. Since the length of w is finite, Λ_w^S contains at most two words. From Definition 3.9, Λ_w^S contains at least one word. Assume by contradiction that $\Lambda_w^S = \{u_1, u_2\}$. Without lost of generality, $u_1 = (01)^{\frac{l}{2}}$, and $u_2 = (10)^{\frac{l}{2}}$. So, from Definition 3.9, we have both:

(1) $a_0 \leq a_1 \geq \ldots \leq a_{l-1} \geq a_0$, and
(2) $a_0 \geq a_1 \leq \ldots \geq a_{l-1} \leq a_0$.

So, $a_0 = a_1 = \ldots = a_{l-1} = a_0$. This contradicts the first condition of Definition 3.9. □

3.2.2 *Definitions and Basic Properties.*

Definition 3.12. *Convex hull*. Given a set P of $n \geq 2$ points $p_1, p_2, \cdots, p_n$ in a plane, the Convex Hull $CH(P)$ (CH for short, if no ambiguity arises) is the smallest polygon such that the n points of P are on its edges or inside it.

Now, we introduce a definition of adjacent, predecessor and successor more general than the one presented in Section 2. Given a team R of n robots located at distinct positions on $CH(R)$, we define that two robots are *CH-adjacent* if and only if they are connected by an edge belonging to $CH(R)$. We also consider that a robot r' is the *CH-successor* (respectively, *CH-predecessor*) of r if and only if r' is the *CH*-adjacent robot in clockwise direction (respectively, counterclockwise direction) on CH.

Observation 3.13. We can associate a unique regular $2k$-gon to a regular k-gon ($k \geq 3$) centered in O, by applying the following construction (refer to Figure 7):

(1) Consider one edge of CH edge $[p_1, p_2]$ of the regular k-gon, and place two points x_1, x_2 on this edge such that $\widehat{x_1 O x_2} = \frac{\pi}{k}$ and the distance between x_1 and p_1 is equal to the distance between x_2 and p_2.
(2) Reiterate with the other edges of CH.

These adding points form a regular $2k$-gon. ◇

Observation 3.14. Given a k-gon ($k \geq 3$) and its associated $2k$-gon (Observation 3.13), let C*1* and C*2*, be the smallest enclosing circles containing the k vertices of the k-gon and the $2k$ vertices of the *2k*-gon, respectively. Then, C_1 and C_2 are concentric circles. ◇

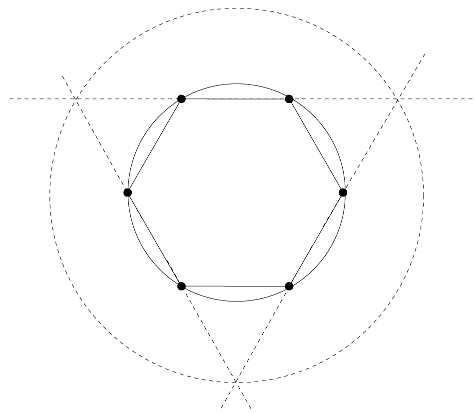


Fig. 7. An example showing a regular $2k$-gon associated to a regular k-gon—$k = 3$.

3.2.2.1 *String of Edges.* We use the subscript i in the notation of a robot r_i, $i \in 1 \ldots n$, to denote the order of the robots in an arbitrary clockwise direction on the Convex Hull CH. We proceed as follows: a robot is arbitrarily chosen as r_1 on CH. Next, for any $i \in 1 \ldots n-1$, r_{i+1} denotes the CH-successor of r_i on CH (in the clockwise direction). Finally, the successor of r_n is r_1.

Let the alphabet A be the set of k ($k \leq n$) strictly positive reals $x_1, x_2, \ldots, x_k$ such that $\forall i \in 1 \ldots n$, there exists $j \in 1 \ldots k$ such that x_j is equal to the length of one of the edges of CH.

The order on A is the natural order ($<$) on the reals. So, the lexicographic order $\preceq$ on the words made over A is defined as follows:

$$u \preceq v \stackrel{\text{def}}{\equiv} (\exists w|\ v = uw) \vee \big(\exists r, s, t,\ \exists a, b \in A|\ (u = ras) \wedge (v = rbt) \wedge (a < b)\big).$$

For instance, if $A = \{1, 2\}$, then $1 \preceq 11 \preceq 12 \preceq 122 \preceq 2$.

For each robot r_i, let us define the word $SE(r_i)$ (respectively, $\overline{SE(r_i)}$) over A (SE stands for "string of edges") as follows:

$$\begin{aligned} SE(r_i) &= a_i a_{i+1} \ldots a_{n-1} a_0 \ldots a_{i-1} \\ \text{(respectively, } \overline{SE(r_i)} &= a_{i-1} \ldots a_0 a_{n-1} a_{n-2} \ldots a_i). \end{aligned}$$

An example showing a string of edges is drawn in Figure 8. Note that, if k robots are on CH, then for every robot r_i, $|SE(r_i)| = |\overline{SE(r_i)}| = k$. Moreover, if the configuration is a regular n-gon, then for every robot r_i, $SE(r_i) = \overline{SE(r_i)} = u^n$, where u is the common length of all the edges of CH.

3.2.2.2 *Swing and Perfect Convex Hulls*

Definition 3.15. *Swing convex hull*. A Convex Hull CH is said to be a swing convex hull (notation, *Swing-CH*) if and only if there exists a robot r_i on CH such that $SE(r_i)$ is a swing word.

The following lemma directly follows from Lemmas 3.10 and 3.11:

LEMMA 3.16. *If $SE(r_i)$ is a swing word, then for every $r_{i'}$ on CH, $SE(r_{i'})$ and $\overline{SE(r_{i'})}$ are swing words.*

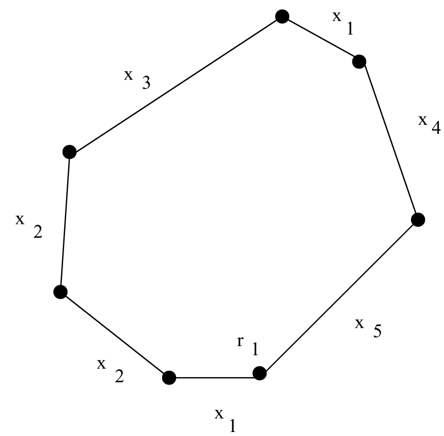


Fig. 8. $SE(r_1) = x_1(x_2)^2x_3x_1x_4x_5$.

COROLLARY 3.17. *If a Convex Hull CH is a swing convex hull, then each robot r_i on CH can determine that CH is a Swing convex hull by locally computing its string of edges, regardless of the local clockwise direction of r_i.*

Let $SE(r_0) = s_0s_1 \cdots s_{l-1}$ be a swing word built on a *Swing-CH* and $u = u_0u_1 \cdots u_{l-1}$ its associate swing word. The edge $[r_i, r_{i+1}]$ is an *up-edge* (respectively, a *down-edge*) iff $u_i = 1$ (respectively, $u_i = 0$). From Lemmas 3.10 and 3.11 again, we can easily deduce that the up-edges and down-edges are the same for all the robots on a *Swing-CH*.

Definition 3.18. *Couple of robots*. Given two robots r_i and $r_{i'}$ such that r_i and $r_{i'}$ are *CH*-adjacent on a *Swing-CH*, we say that they form a *couple* if and only if the *CH* edge, linking r_i and $r_{i'}$ is a down-edge.

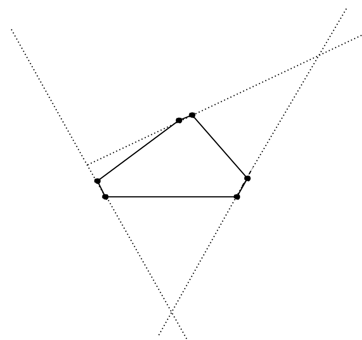
In a *Swing-CH*, denote by *SetLines* the set of lines (r_i, r_{i+1}) passing through both robots of the same couple. $IntersectionLines$ (*IL*) is the set of intersection points between all lines $l_1 = (r_i, r_{i+1})$ and $l_2 = (r_{i+2}, r_{i+3})$ such that $l_1 \in SetLines, l_2 \in SetLines$ and r_{i+1} is *CH*-adjacent to r_{i+2}.

Definition 3.19. *Perfect Convex Hull*. Let a team of $n = 2k$ robots on the convex hull *CH*. *CH* is *perfect* if the following four conditions hold:

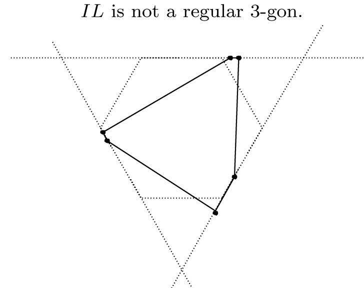
(1) $n = 2k$ and $k \geq 3$.
(2) *CH* is a *Swing*-CH.
(3) *IL* is a regular k-gon.
(4) All the robots are on the edges of the regular $2k$-gon associated to *IL*.

Note that Condition (1) in the previous definition excludes the case $k < 3$ because, in this case, *IL* does not exist.

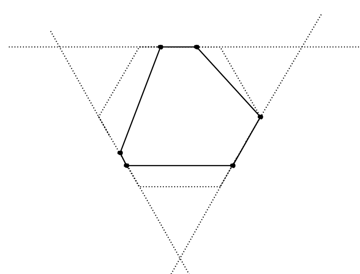
We say that the cohort R is in *perfect convex hull* if $CH(R)$ is perfect and all the robots are located at distinct positions on it—refer to Figure 9 where non-perfect and perfect convex hulls are shown.



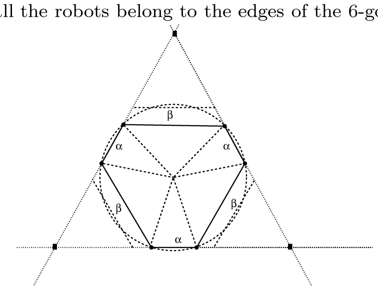
(a) A Swing convex hull which is not perfect: IL is not a regular 3-gon.



(b) A Swing convex hull which is not perfect: there exist robots that are not on the edges of the regular 6-gon associated to IL.



(c) A Perfect convex hull: all the robots belong to the edges of the 6-gon.



(d) A strict biangular circle is also a Perfect convex hull, illustrating Observation 3.20.

Fig. 9. Examples showing Swing convex hulls.

Algorithm 1. Procedure $\langle N \rightsquigarrow \texttt{gon}\rangle$ for Robot r_i.

n := the number of robots;
if n is even
then **if** the robots do not form a regular n-gon
  then **if** the robots form a perfect convex hull
    then Execute $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$;
    else Execute $\langle A \rightsquigarrow B\rangle$;
else Execute $\langle A \rightsquigarrow B\rangle$;

Observation 3.20. If all the robots are in strict biangular circle then they are in perfect convex hull.

Definition 3.21. *Equivalence*. Two perfect convex hulls are said to be *equivalent* if they share the same regular k-gon IL in a system configuration.

The only possible difference between two equivalent perfect convex hulls is different positions of the robots on the regular n-gon associated to IL.

3.2.3 *Protocol.* The protocol is sketched in Algorithm 1. As for the first attempt, it uses Procedure $\langle A \rightsquigarrow B\rangle$ [Katreniak 2005]. Procedure $\langle A \rightsquigarrow B\rangle$ excluded, the protocol mainly consists of one procedure called Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$, which is used when the system forms a perfect convex hull. It leads the system into a regular n-gon. Let us explain how the procedures are used by giving the overall scheme of Algorithm 1. Starting from an arbitrary configuration, using Procedure $\langle A \rightsquigarrow B\rangle$, the system is eventually in a biangular circle. If n is odd, then the robots form a regular n-gon and the system is done. Otherwise (n is even), the robots form either a regular n-gon or a strict biangular circle. Starting from the latter case, each robot executes Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$ (from Observation 3.20, we know that when the robots form a strict biangular circle, they form a perfect convex hull too). The resulting configuration of the execution of Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$ is a regular n-gon.

THEOREM 3.22. *Assuming that, initially, no two robots are located at the same position, Procedure $\langle N \rightsquigarrow N\texttt{gon}\rangle$ is a deterministic circle formation protocol for any number $n \geq 5$ of robots in SSM.*

This theorem follows from Procedure $\langle N \rightsquigarrow N\texttt{gon}\rangle$, Algorithm 1, [Katreniak 2005], and Lemmas 3.24. This last lemma shows that, starting from a perfect convex hull, Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$, described as follows, deterministically solves CFP.

3.2.3.1 *Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$.* Let us assume that in the initial configuration, the robots form a perfect convex hull. In such a configuration, every active robot r_i executes Algorithm 2.

LEMMA 3.23. *In SSM, using Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$, if all the robots are in a perfect convex hull at time t_j, then at time t_{j+1}, either the configuration is an equivalent perfect convex hull, or the n-gon is formed.*

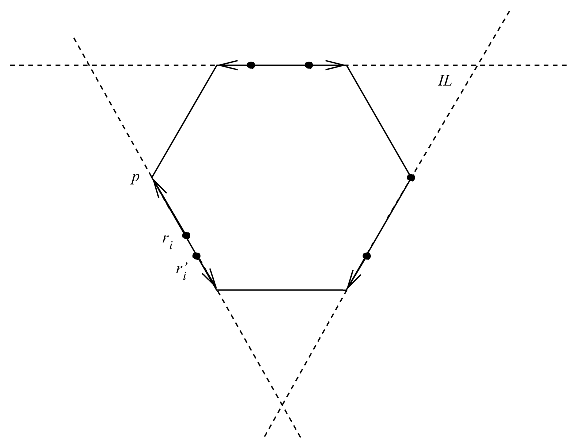


Fig. 10. An example describing Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$.

Algorithm 2. Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$ for Robot r_i.

Let *CH* be the convex hull;
Compute *IL*, the set of intersection lines (forming the $\frac{n}{2}$-gon);
Let r_i' be r_i's neighbor such that $\{r_i, r_i'\}$ is a couple of *CH*;
Let p be the position on the associated n-gon on the opposite direction
of r_i' on the line (r_i, r_i')—refer to Figure 10;
Move toward p;

Proof. From Lemma 3.16 and Corollary 3.17, all the perfect convex hulls computed by the robots are equivalent. Let *Unit* denote the distance between two adjacent vertices on the associated n-gon. Using Procedure $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$, at time t_{j+1}, the distance between two *CH*-adjacent robots forming a couple is lower than or equal to *Unit*, whereas the distance between two *CH*-adjacent robots, not forming a couple, is greater than or equal to *Unit*. So, at time t_{j+1}, if for every pair of adjacent robots (r_i, r_i'), the distance between (r_i, r_i') is equal to *Unit*, then the n-gon is formed. Otherwise, the up-edge and the down-edge are the same ones as t_j, because $|\,down\ edge\,| \leq Unit \leq |\,up\ edge\,|$. Furthermore, each robot moves only on the edge of the associated regular n-gon without collision. So, each robot can recompute the same *IL* and the same associated regular n-gon. □

The following lemma follows from Lemma 3.23 and fairness:

Lemma 3.24. *In SSM, Procedure* $\langle \texttt{PCH} \rightsquigarrow N\texttt{gon}\rangle$ *is a deterministic algorithm transforming a perfect convex hull into a regular n-gon if $n \geq 5$.*

3.2.3.2 *Procedure* $\langle N \rightsquigarrow \texttt{gon}\rangle$ *in CORDA.* Assume the fully asynchronous model CORDA, that is, one or more robots can move while some others are either observing or computing—the case of waiting robots is beyond the scope of this discussion. Let us assume that the robots form a perfect convex hull. Lemma 3.23 shows that, while the regular n-gon is not formed, Procedure $\langle \mathtt{PCH} \rightsquigarrow N\mathtt{gon} \rangle$ guarantees that the regular $\frac{n}{2}$-gon IL and the associated regular n-gon are preserved for every robot. So, even assuming CORDA, the perfect convex hull computed by the robots remains equivalent for every pair of robots, while the regular n-gon is not formed. Thus Lemma 3.23 also holds in CORDA. As a result, the following theorem holds:

THEOREM 3.25. *Starting from a biangular configuration, Procedure $\langle N \rightsquigarrow \mathtt{gon} \rangle$ is a deterministic circle formation protocol for any number $n \geq 5$ of robots in CORDA.*

4. CONCLUSION

In this article, we studied the problem of forming a regular n-gon with a cohort of n robots (circle formation problem). We presented a new approach for this problem based on concentric circles formed by the robots. Combined with the biangular circle formation protocol given in Katreniak [2005], our method leads to circle formation patterns for any number of robots $n \geq 5$. In the case of $n < 5$ robots, our approach could not be applied because of the non uniqueness of the computed concentric circles.

This article provides two different implementations in the semi-synchronous model (SSM). While the first protocol assumed the robots reaching their final positions in one step, and used concentric circles only, the second protocol outperforms the former by using intrinsic properties on particular words introduced in this work, so called swing words. Furthermore, we show that, starting from a biangular configuration, this second approach solves CFP in CORDA. This can be seen as a first step to solving CFP in CORDA. Indeed, our second approach works in CORDA starting from a *biangular configuration* only. It is not clear whether it can be combined in CORDA with the solution proposed in Katreniak [2005], as we did in SSM. Actually, the solvability of CFP in CORDA remains an open question.

ACKNOWLEDGMENTS

We are grateful for the many comments provided by the anonymous reviewers. Their suggestions helped us greatly in improving the presentation of these results.

REFERENCES

CHATZIGIANNAKIS, I., MARKOU, M., AND NIKOLETSEAS, S. 2004. Distributed circle formation for anonymous oblivious robots. In *Proceedings of the 3rd International Workshop on Experimental and Efficient Algorithms (WEA)*. Lecture Notes in Computer Science, vol. 3059, Springer, Berlin, Germany. 159–174.

DEBEST, X. A. 1995. Remark about self-stabilizing systems. *Commun. ACM*, *38*, 2, 115–117.

DÉFAGO, X. AND KONAGAYA, A. 2002. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *Proceedings of the 2nd ACM International Annual Workshop on Principles of Mobile Computing (POMC)*. ACM, New York, NY, 97–104.

DÉFAGO, X. AND SOUISSI, S. 2008. Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. *Theor. Comput. Sci. 396*, 1-3, 97–112.

DIEUDONNÉ, Y. LABBANI-IGBIDA, O., AND PETIT, F. 2006. Circle formation of weak mobile robots. In *Proceedings of the 8th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*. Lecture Notes in Computer Science, vol. 4280. Springer, Berlin, Heidelberg, New York, 262–275.

DIEUDONNÉ, Y. AND PETIT, F. 2007a. Circle formation of weak robots and Lyndon words. *Inform. Proc. Lett. 104*, 4, 156–162.

DIEUDONNÉ, Y. AND PETIT, F. 2007b. Deterministic leader election in anonymous sensor networks without common coordinated system. In *Proceedings of the 11th International Conference On Principles of Distributed Systems (OPODIS)*. Lecture Notes in Computer Science, vol. 4878. Springer, Berlin, Germany, 132–142.

DIEUDONNÉ, Y. AND PETIT, F. 2007c. Swing words to make circle formation quiescent. In *Proceedings of the 14th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*. Lecture Notes in Computer Science, vol. 4474. Springer, Berlin, Germany, 169–179.

FLOCCHINI, P., PRENCIPE, G., AND SANTORO, N. 2006. Self-deployment algorithms for mobile sensors on a ring. In *Proceedings of the 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors)*. Lecture Notes in Computer Science. Springer, Berlin, Germany, 59–70.

FLOCCHINI, P., PRENCIPE, G., SANTORO, N., AND WIDMAYER, P. 2000. Distributed coordination of a set of autonomous mobile robots. In *Proceedings of the IEEE Intelligent Vehicle Symposium (IV)*. IEEE Computer Society, Los Alamitos, CA, 480–485.

KATRENIAK, B. 2005. Biangular circle formation by asynchronous mobile robots. In *Proceedings of the 12th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*. Lecture Notes in Computer Science. vol. 5058, Springer, Berlin, Germany, 185–199.

LOTHAIRE, M. 1983. *Combinatorics on Words*. Addison-Wesley.

PRENCIPE, G. 2001. Corda: distributed coordination of a set of autonomous mobile robots. In *Proceedings of the 4th European Research Seminar on Advances in Distributed Systems (ERSADS)*. 185–190.

PRENCIPE, G. 2002. Distributed coordination of a set of autonomous mobile robots. Ph.D. thesis, Dipartimento di Informatica, University of Pisa.

PRENCIPE, G. AND SANTORO, N. 2006. Distributed algorithms for autonomous mobile robots. In *Proceedings of the IFIP International Conference on Theoretical Computer Science (TCS)*. Springer, Berlin, Germany, 47–62.

SUGIHARA, K. AND SUZUKI, I. 1996. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Robot. Syst. 3*, 13, 127–139.

SUZUKI, I. AND YAMASHITA, M. 1996. Agreement on a common x-y coordinate system by a group of mobile robots. *Intelligent Robots: Sensing, Modeling and Planning*. World Scientific Press, 305–321.

SUZUKI, I. AND YAMASHITA, M. 1999. Distributed anonymous mobile robots—formation of geometric patterns. *SIAM J. Comput. 28*, 4, 1347–1363.

Received February 2007; revised July 2008; accepted September 2008