

Leader Election Problem Versus Pattern Formation Problem

Yoann Dieudonné, Franck Petit, Vincent Villain

► **To cite this version:**

Yoann Dieudonné, Franck Petit, Vincent Villain. Leader Election Problem Versus Pattern Formation Problem. [Research Report] 2009. <inria-00361916v2>

HAL Id: inria-00361916

<https://hal.inria.fr/inria-00361916v2>

Submitted on 20 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Leader Election Problem Versus Pattern Formation Problem

Yoann Dieudonné[†], Franck Petit[‡] and Vincent Villain[†]

[†] MIS Lab., Université de Picardie Jules Verne, France

[‡] LIP6/Regal, Université Pierre et Marie Curie, INRIA, CNRS, France

Abstract

Leader election and arbitrary pattern formation are fundamental tasks for a set of autonomous mobile robots. The former consists in distinguishing a unique robot, called the leader. The latter aims in arranging the robots in the plane to form any given pattern. The solvability of both these tasks turns out to be necessary in order to achieve more complex tasks.

In this paper, we study the relationship between these two tasks in a model, called *CORDA*, wherein the robots are weak in several aspects. In particular, they are *fully asynchronous* and they have no direct means of communication. They cannot remember any previous observation nor computation performed in any previous step. Such robots are said to be *oblivious*. The robots are also *uniform* and *anonymous*, i.e, they all have the same program using no global parameter (such that an identity) allowing to differentiate any of them. Moreover, none of them share any kind of common coordinate mechanism or common sense of direction, except that they agree on a common handedness (*chirality*).

In such a system, Flochini *et al.* proved in [9] that it is possible to solve the leader election problem for $n \geq 3$ robots if the arbitrary pattern formation is solvable for $n \geq 3$. In this paper, we show that the converse is true for $n \geq 4$ and thus, we deduce that both problems are equivalent for $n \geq 4$ in *CORDA* provided the robots share the same chirality. The possible equivalence for $n = 3$ remains an open problem in *CORDA*.

Keywords: Mobile Robot Networks, Pattern Formation Problem, Leader Election Problem.

Contact Author:

Franck Petit

Address: 104 avenue du Président Kennedy 75016 Paris France

Email: Franck.Petit@lip6.fr

Phone: +33 681 949 729

1 Introduction

Mobile robots working together to perform cooperative tasks in a given environment is an important, open area of research. *Teams* (or, *swarms*) of *mobile robots* provide the ability to measure properties, collect information and act in a given physical environment. Numerous potential applications exist for such multi-robot systems, to name only a very few: environmental monitoring, large-scale construction, risky area surrounding or surveillance, and exploration of awkward environments.

In a given environment, the ability for the swarm of robots to succeed in the accomplishment of the assigned task greatly depends on (1) *global* properties assigned to the swarm, and (2) *individual* capabilities each robot has. Examples of such global properties are the ability to distinguish among themselves at least one (or, more) robots (*leader*), to agree on a common global direction (*sense of direction*), or to agree on a common handedness (*chirality*). The individual capacities of a robot are its moving capacities and its sensory organs.

To deal with cost, flexibility, resilience to dysfunction, and autonomy, many problems arise for handling the distributed coordination of swarms of robots in a *deterministic* manner. This issue was first studied in [14, 15], mainly motivated by the minimal level of ability the robots are required to have in the accomplishment of basic cooperative tasks. In other words, the feasibility of some given tasks is addressed assuming swarm of autonomous robots either devoid or not of capabilities like (observable) identifiers, direct means of communication, means of storing previous observations, sense of direction, chirality, etc. So far, except the “classical” *leader election* problem [1, 5, 9, 12], most of the studied tasks are geometric problems, so that *pattern formation*, *line formation*, *gathering*, and *circle formation*—refer to [3, 4, 6, 8, 9, 10, 15] for these problems.

In this paper, we concentrate on two of the aforementioned problems: leader election and pattern formation. The former consists in moving the system from an initial configuration where all entities are in the same state into a final configuration where all entities are in the same state, except one, the leader. The latter consists in the design of protocols allowing autonomous mobile robots to form any (arbitrary) geometric pattern.

The issue of whether the pattern formation problem can be solved or not according to some capabilities of the robots is addressed in [9]. Assuming that every robot is able to observe all its pairs, the authors consider sense of direction and chirality. They show by providing an algorithm that, if the robots have sense of direction and chirality, then they can form any arbitrary pattern. They refine their result by showing that with the lack of chirality (*i.e.*, assuming that they have sense of direction only), the problem can be solved in general with an odd number of robots only. They also show that, assuming robots having no sense of direction, then, in general, the robots cannot form an arbitrary pattern, even with chirality. As a matter of fact, the idea of proof relies on the fact that if it is possible to solve the pattern formation problem, then the robots can form an asymmetric configuration in order to distinguish a unique robot. That means that, the ability to (deterministically) form a particular type of patterns implies the ability to (deterministically) elect a robot in the system as the leader. In other words, if it is not possible to solve the leader election problem, then it is not possible to solve the pattern formation problem. However, in [5], assuming anonymous robots (possibly motionless) with chirality only (without sense of direction), the authors provide a complete characterization (necessary and sufficient conditions) on the robots positions to deterministically elect a leader. An interesting question arises from the above facts: “*With robots devoid of sense of direction, does the (arbitrary) pattern formation problem becomes solvable if the robots have the possibility to distinguish a unique leader?*” In [17], the authors provide a positive answer to this question assuming that robots have the chirality property. Their result holds in the

semi-synchronous model (SSM) introduced in [14], *a.k.a.* Model SYm in the literature. In this paper, we show that this results also holds for $n \geq 4$ robots in a fully asynchronous model, called CORDA.

Combined with the result in [9] and provided the robots have the chirality property, we deduce that Leader Election and Pattern Formation are two *equivalent* problems in CORDA for $n \geq 4$ robots, in the precise sense that, the former problem is solvable if and only if the latter problem is solvable. The possible equivalence for the case $n = 3$ remains an open problem in CORDA.

The rest of the paper is organized as follows: In Section 2, we describe the distributed systems and state the problems considered in this paper. The proof of equivalence is given in Section 3 for any $n \geq 4$ by providing an algorithm working in CORDA. (Due to lack of space, technical proofs have been moved in the appendix.) Finally, we make concluding remarks in Section 4.

2 Preliminaries

In this section, we define the distributed system and the problems considered in this paper.

2.1 Distributed Model.

We adopt the model CORDA introduced in [13]. The *distributed system* considered in this paper consists of n robots r_1, r_2, \dots, r_n —the subscripts $1, \dots, n$ are used for notational purpose only. Each robot r_i is viewed as a point in a two-dimensional space unbounded and devoid of any landmark. When no ambiguity arises, r_i also denotes the position in the plane occupied by that robot. Each robot has its own local coordinate system and unit measure. The robots do not agree on the orientation of the axes of their local coordinate system, nor on the unit measure.

Definition 1 (Sense of Direction) *A set of n robots has sense of direction if the n robots agree on a common direction of one axis (x or y) and its orientation. The sense of direction is said to be partial if the agreement relates to the direction only —i.e. they are not required to agree on the orientation.*

In Figure 1, the robots have sense of direction in the cases (a) and (b), whereas they have no sense of direction in the cases (c) and (d).

Given an x - y Cartesian coordinate system, the *handedness* is the way in which the orientation of the y axis (respectively, the x axis) is inferred according to the orientation of the x axis (resp., the y axis).

Definition 2 (Chirality) *A set of n robots has chirality if the n robots share the same handedness.*

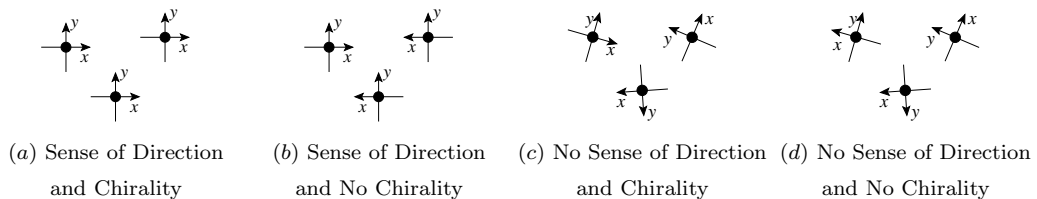


Figure 1: Four examples showing the relationship between Sense of Direction and Chirality

In Figure 1, the robots have chirality in the cases (a) and (c), whereas they have no chirality in the cases (b) and (d). In the sequel, we assume that all the robots have the chirality property.

The robot's life is viewed as an infinite sequence of cycles. Each cycle is a sequence of four states Wait-Observe-Compute-Move characterized as follows.

Life cycle. Initially, a robot is in the waiting state (**Wait**). Asynchronously and independently from other robots, it observes its surroundings (**Observe**) by using its sensors. The latter ones return a set of all the positions occupied by at least one robot, with respect to its own coordinate system. Then, from its new observations the robot computes its next location (**Compute**) according to a given protocol which is the same one for all the robots. Once the computation is done, the robot moves towards its new location (**Move**). However, it is assumed the distance travelled by a robot in a cycle is bounded by a constant σ . So, if the destination point is too far, the robot may stop its motion before reaching the computed location. Finally, the robot returns to the waiting state. It is assumed that the amount of time spent in each phase of a cycle is finite but unpredictable and may be different for each cycle and for each robot. That is why the robots are considered to be fully asynchronous.

Finally we assume that the robots are *uniform* and *anonymous*, i.e., they all have the same program using no local parameter (such that an identity) allowing to differentiate any of them. Moreover, they have no direct means of communication and they are *oblivious*, i.e., none of them can remember any previous observation nor computation performed in any previous cycles.

2.2 Leader Election Problem

The *leader election* problem considered in this paper is stated as follows: Given the positions of n robots in the plane, the n robots are able to deterministically agree on the same robot L called the leader. Initially, the robots are in arbitrary positions, with the only requirement that no two robots are in the same position.

2.3 Arbitrary Pattern Formation Problem

In the *Arbitrary Pattern Formation Problem*, the robots have in input the same pattern, called the target pattern \mathcal{P} , described as a set of positions in the plane given in lexicographic order (each robot sees the same pattern according to the direction and orientation of its local coordinate system). They are required to form the pattern: at the end of the computation, the positions of the robots coincide, in everybody's local view, with the positions of \mathcal{P} , where \mathcal{P} may be translated, rotated, and scaled in each local coordinate system. Initially, the robots are in arbitrary positions, with the only requirement that no two robots are in the same position, and that, of course, the number of positions prescribed in the pattern and the number of robots are the same.

3 Equivalence for $n \geq 4$

In this section we prove the following theorem:

Theorem 3 *In CORDA, assuming a cohort of $n \geq 4$ robots having chirality and devoid of any kind of sense of direction, if the leader election problem is solvable, then the pattern formation problem is solvable.*

In a recent paper [9], the authors prove the following result:

Theorem 4 [9] *In $CORDA$, assuming a cohort of $n \geq 3$ robots devoid of any kind of sense of direction, if it is possible to solve the pattern formation problem, then the leader election problem is solvable too.*

So, to prove Theorem 3, from Theorem 4, it remains to show the following lemma:

Lemma 5 *In $CORDA$, assuming a cohort of $n \geq 4$ robots having chirality and devoid of any kind of sense of direction, if the leader election problem is solvable, then the pattern formation problem is solvable.*

The proof of Lemma 5 is mainly based on the existence of a protocol allowing to form an arbitrary target pattern if initially the robots are in a configuration allowing the robots to deterministically elect a leader. Such a configuration is called a *leader configuration*.

Definition 6 (Leader configuration) *A configuration allowing the robots to deterministically elect a leader is called a leader configuration.*

3.1 Definitions and Basic Properties

In the rest of this paper, we assume the set of all the positions \mathcal{Q} occupied by the robots in the plane is the set of all the coordinates expressed in a cartesian coordinate system \mathcal{S} which is unknown for all the robots. However, all the coordinates \mathcal{Q} expressed in \mathcal{S} coincide with all the coordinates \mathcal{Q} expressed in everybody's local system where \mathcal{Q} may be translated, rotated or scaled.

Definition 7 (Smallest enclosing circle) [4] *Given a set \mathcal{Q} of $n \geq 2$ positions p_1, p_2, \dots, p_n on the plane, the smallest enclosing circle of \mathcal{Q} , called $SEC(\mathcal{Q})$, is the smallest circle enclosing all the positions in \mathcal{Q} .*

When no ambiguity arises, $SEC(\mathcal{Q})$ is shortly denoted by SEC and $SEC(\mathcal{Q}) \cap \mathcal{Q}$ indicates the set of all the positions both on $SEC(\mathcal{Q})$ and \mathcal{Q} . Besides, we say that a robot r is inside SEC if, and only if, r is not located on the circumference of SEC . In any configuration \mathcal{Q} , SEC is unique and can be computed in linear time [11]. Note that since the robots have the ability of chirality, they are able to agree on a common orientation of SEC , denoted \circlearrowright , in the sequel referred to as the clockwise direction.

The following property contains some details about the smallest enclosing circle:

Property 8 [16] *SEC passes either through two of the positions that are on the same diameter (opposite positions), or through at least three positions. SEC does not change by eliminating or adding positions that are inside it. SEC does not change by adding positions on its boundary. However, it may be possible that SEC changes by either eliminating or moving positions on its circumference.*

Examples showing the latter assertion of Property 8 are proposed in Figure 2.

Definition 9 (Critical position) [10] *Given a set \mathcal{Q} of distinct positions. We say that a position p is critical iff $SEC(\mathcal{Q}) \neq SEC(\mathcal{Q} \setminus \{p\})$.*

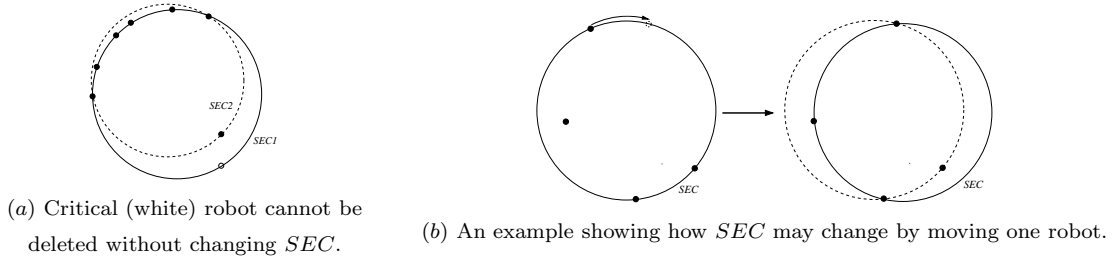


Figure 2: Examples illustrating Property 8

An example of such a critical robot is given by Figure 2, Case (a). According to Property 8, a critical position cannot be inside SEC . So, we have the following corollary:

Corollary 10 *Let \mathcal{Q} be a configuration. If there exists a critical position p in \mathcal{Q} , then p is on the circumference of $SEC(\mathcal{Q})$.*

Before giving other properties about critical positions, we need to define extra notions.

Definition 11 ($adjacent(r, C, \circlearrowright)$) *Given a circle C and a group of robots located on it, we say that $r' = adjacent(r, C, \circlearrowright)$ if r' is the next robot on C just after r in the clockwise direction.*

In the same way, we can define $adjacent(r, \circlearrowleft)$ in the counterclockwise direction. When no ambiguity arises, $adjacent(r, C, \circlearrowleft)$ is shortly denoted by $adjacent(r, \circlearrowleft)$. Sometimes, if $r' = adjacent(r, \circlearrowleft)$, we simply say that r' and r are *adjacent*.

Definition 12 ($angle(p, c, p', \circlearrowright)$) *Given a circle C centered at c and two points p and p' located on it, $angle(p, c, p', \circlearrowright)$ is the angle centered at c from p to p' in the clockwise direction.*

In the same way, we can define $angle(p, c, p', \circlearrowleft)$ in the counterclockwise direction.

The following properties are fundamental results about smallest enclosing circles:

Lemma 13 [2] *Let r_i, r_j and r_k be three consecutive robots on SEC centered at c such that $r_j = adjacent(r_i, \circlearrowright)$ and $r_k = adjacent(r_j, \circlearrowright)$. If $angle(r_i, c, r_k, \circlearrowright) \leq 180^\circ$, then r_j is non-critical and SEC does not change by eliminating r_j .*

Corollary 14 *Let $SEC(\mathcal{Q})$ be the smallest circle enclosing all the positions in \mathcal{Q} . For all couple of positions r_i and r_j in $SEC(\mathcal{Q}) \cap \mathcal{Q}$ such that $r_j = adjacent(r_i, \circlearrowright)$, we have $angle(r_i, c, r_j, \circlearrowright) \leq 180^\circ$.*

Lemma 15 [2] *Given a smallest enclosing circle with at least four robots on it. There exists at least one robot which is not critical.*

Definition 16 (Concentric Enclosing Circle) *Given a set P of distinct positions. We say that C^P is a concentric enclosing circle if and only if it is centered at the center c of SEC , has a radius strictly greater than zero and it passes through at least one position in P .*

In the following, \mathcal{SC}^P and $|\mathcal{SC}^P|$ respectively denote the set of all the concentric enclosing circle in P and its cardinality. For some k such that $1 \leq k \leq |\mathcal{SC}^P|$, C_k^P indicates the k^{th} greatest concentric enclosing circle in P and $\bigcup_{i=1}^k C_i^P$ is the set of the k first greatest enclosing circles in P . Moreover, we assume that a position (or robot) located inside a concentric enclosing circle C_k^P is not on the circumference of C_k^P . $C_i^P \cap P$ indicate the set of all the positions both on C_i^P and P .

Remark 17 *From Definition 16, SEC is the greatest concentric enclosing circle of SC (i.e., $SEC = C_1$) and the center of SEC cannot be a concentric enclosing circle.*

From Definition 16, we can introduce the notion of *agreement configuration*:

Definition 18 (Agreement Configuration) *A configuration \mathcal{Q} is an agreement configuration if, and only if both following conditions hold:*

1. *There exists a robot r_l in \mathcal{Q} such that r_l is the unique robot located on the smallest concentric enclosing circle $C_{|\mathcal{SC}^{\mathcal{Q}}|}^{\mathcal{Q}}$,*
2. *There is no robot at the center of $SEC(\mathcal{Q})$.*

In an agreement configuration, r_l is called the *leader*

Definition 19 (Equivalent agreement configuration) *Two agreement configuration \mathcal{Q}_1 and \mathcal{Q}_2 is said to be equivalent if, and only if, both following conditions hold:*

1. *$SEC(\mathcal{Q}_1)$ and $SEC(\mathcal{Q}_2)$ are superimposed.*
2. *Let c_1 and c_2 be respectively the center of $SEC(\mathcal{Q}_1)$ and the center of $SEC(\mathcal{Q}_2)$. Let r_{l1} and r_{l2} be respectively the leader in \mathcal{Q}_1 and the leader in \mathcal{Q}_2 . $[c_1, r_{l1})$ and $[c_2, r_{l2})$ are superimposed.*

We end with the three following definitions:

Definition 20 ($Map(\mathcal{Q}, \mathcal{P})$) *Let \mathcal{Q} and \mathcal{P} be respectively an agreement configuration formed by the robots in the plane and a target pattern.*

$Map(\mathcal{Q}, \mathcal{P})$ is the set of all the final positions \mathcal{P} expressed in the plane where the robots currently lies and computed as follows:

1. *First, the center of $SEC(\mathcal{P})$ is translated to the center of $SEC(\mathcal{Q})$.*
2. *Then, let o, c, r_l and s be respectively the center of $SEC(\mathcal{Q})$, the center of $SEC(\mathcal{P})$, the leader in \mathcal{Q} and the first non-critical position (in the lexicographic order) located on the smallest concentric enclosing circle of \mathcal{P} . \mathcal{P} is rotated so that the half-line $[o, r_l)$ is viewed as the half-line $[c, s)$.*
3. *Finally, \mathcal{P} is scaled with respect to the radius of $SEC(\mathcal{Q})$ in order that all the distances are expressed according to the radius of $SEC(\mathcal{Q})$. In particular $SEC(\mathcal{Q}) = SEC(\mathcal{P})$.*

An example showing the construction of Definition 20 is given in Figure 3.

Definition 21 ((k, \mathcal{P}) -partial pattern) *Let \mathcal{Q} and \mathcal{P} be respectively an agreement configuration formed by the robots in the plane and a target pattern. We say that:*

1. *\mathcal{Q} is a $(0, \mathcal{P})$ -partial pattern if the leader in \mathcal{Q} is inside the smallest concentric enclosing circle of $Map(\mathcal{Q}, \mathcal{P})$.*
2. *\mathcal{Q} is a (k, \mathcal{P}) -partial pattern with $1 \leq k \leq \text{Min}(|\mathcal{SC}^{\mathcal{Q}}|, |\mathcal{SC}^{\mathcal{P}}|)$ if the three following properties holds:*

- a. *\mathcal{Q} is a $(0, \mathcal{P})$ -partial pattern.*
- b. *$C_k^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P}) \subseteq C_k^{\mathcal{Q}} \cap \mathcal{Q}$.*
- c. *$\bigcup_{i=1}^{k-1} C_i^{\mathcal{Q}} \cap \mathcal{Q} = \bigcup_{i=1}^{k-1} C_i^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$.*

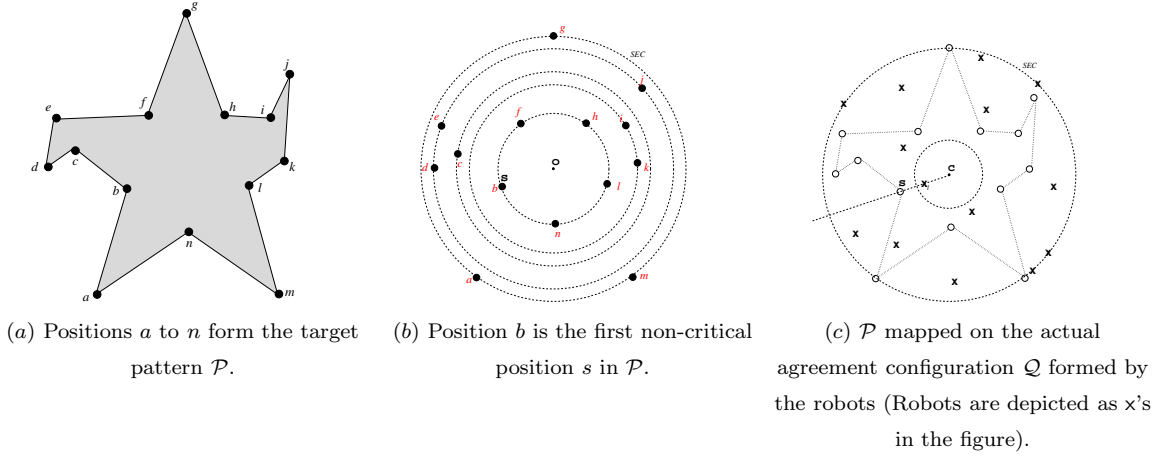


Figure 3: An example showing a pattern \mathcal{P} mapped on an agreement configuration \mathcal{Q} —Definition 20.

In the sequel, we say that \mathcal{Q} is a *maximal* (k, \mathcal{P}) -partial pattern if \mathcal{Q} is a (k, \mathcal{P}) -partial pattern and not a $(k + 1, \mathcal{P})$ -partial pattern.

Definition 22 (Extra robots) Let \mathcal{P} and \mathcal{Q} be respectively a target pattern and a configuration formed by the robots in the plane such that \mathcal{Q} is a maximal (k, \mathcal{P}) -partial pattern. We say that a robot r is an extra robot if one of the two following properties holds:

1. $k = 0$, r is inside $SEC(\mathcal{Q})$, and r is not the leader in \mathcal{Q} ;
2. $k \geq 1$ and
 - (a) either r is inside the enclosing circle $C_k^{Map(\mathcal{Q}, \mathcal{P})}$ and r is not the leader in \mathcal{Q} ;
 - (b) or r is on the circumference of $C_k^{Map(\mathcal{Q}, \mathcal{P})}$ and r does not occupy a position in $C_k^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$.

3.2 The protocol

Starting from a leader configuration, the protocol, shown in Algorithm 1, allows to form any target pattern \mathcal{P} . It is a compound of two procedures presented in the two following subsections:

1. Protocol $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$ transforms an arbitrary leader configuration into an agreement configuration.
2. Protocol $\langle \text{Agreement} \rightsquigarrow \text{Pattern} \rangle$ transforms an agreement configuration into a pattern \mathcal{P} .

Algorithm 1 Form an arbitrary pattern starting from a leader configuration ($n \geq 4$).

$\mathcal{P} :=$ the target pattern ;
if the robots do not form the target pattern
then **if** the robots do not form an agreement configuration
 then Execute $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$;
 else Execute $\langle \text{Agreement} \rightsquigarrow \text{Pattern} \rangle$;

3.2.1 Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$.

Algorithm 2 Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$ for any robot r_i in an arbitrary leader configuration

$\mathcal{Q} :=$ the configuration where the robots currently lies;
 $r_l := \text{Leader}(\mathcal{Q})$;
 $c :=$ center of $\text{SEC}(\mathcal{Q})$
if r_l is located at c
then $r_k :=$ the closest robot to $c \in \mathcal{Q} \setminus \{r_l\}$;
 $p :=$ the middle of the segment $[r_l; r_k]$;
if I am r_l
then $\text{MoveTo}(p, \rightarrow)$;
endif
else **if** r_l is not critical
then $p :=$ the middle of the segment $[r_l; c]$;
if I am r_l
then $\text{MoveTo}(p, \rightarrow)$;
endif
else /* r_l is critical and r_l is on SEC^* /
 $r_k :=$ the first non-critical robot starting from r_l on SEC in clockwise.
if I am r_k
then $p :=$ the middle of the segment $[r_k; c]$;
 $\text{MoveTo}(p, \rightarrow)$;

In a leader configuration, we have the following corollary:

Corollary 23 [5] *If the robots are in a leader configuration, then they can distinguish a unique leader which is one of the closest robot to the center of the smallest enclosing circle of the configuration, provided that they share the property of chirality.*

So, from Corollary 23, we know that we can distinguish a unique robot r_l , called the leader, which is one of the robots closest to the center c of $\text{SEC}(\mathcal{Q})$. However, according to Definition 18, if r_l is at the center of $\text{SEC}(\mathcal{Q})$ or if r_l is not the unique robot closest to the center of $\text{SEC}(\mathcal{Q})$, \mathcal{Q} is not an agreement configuration. In that case, Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$ allows to transform the leader configuration into an agreement configuration. Algorithm 2 describes Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$.

In Algorithm 2, we use two subsoutines: $\text{Leader}(\mathcal{Q})$ and $\text{MoveTo}(p, \rightarrow)$. The former returns the unique leader from a leader configuration \mathcal{Q} . The latter allows a robot r to move towards the point p , using a straight movement.

3.2.2 Procedure $\langle \text{Agreement} \rightsquigarrow \text{Pattern} \rangle$.

Procedure $\langle \text{Agreement} \rightsquigarrow \text{Pattern} \rangle$ is shown in Algorithm 3.

The routine $\text{Nearest_extra_robot}(C_{k+1}^{\text{Map}(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, \text{Map}(\mathcal{Q}, \mathcal{P}))$ returns an extra robot r such that r is the closest extra robot to $C_{k+1}^{\text{Map}(\mathcal{Q}, \mathcal{P})}$ which is not located on $C_{k+1}^{\text{Map}(\mathcal{Q}, \mathcal{P})}$. If several candidates exist, then the extra robots inside $C_{k+1}^{\text{Map}(\mathcal{Q}, \mathcal{P})}$ have priority. Finally, if there is again several candidates then these latter ones are located on the same concentric circle C centered at the center c of $C_{k+1}^{\text{Map}(\mathcal{Q}, \mathcal{P})}$ and

Algorithm 3 Procedure $\langle \text{Agreement} \rightsquigarrow \text{Pattern} \rangle$ for any robot r_i in an agreement configuration

\mathcal{Q} := the configuration where the robots currently lies;
 \mathcal{P} := the target pattern; /* \mathcal{P} is the same for all the robots */
 r_l := $Leader(\mathcal{Q})$;
 s := the first non-critical position located on the smallest concentric enclosing circle of $Map(\mathcal{Q}, \mathcal{P})$;
if the robots do not form any (k, \mathcal{P}) -partial pattern
then /* r_l is not inside the smallest concentric enclosing circle of $Map(\mathcal{Q}, \mathcal{P})$ */
 p := the middle of the segment $[c; s]$;
 if I am r_l
 then $MoveTo(p, \rightarrow)$;
 endif
else /* the robots form a (k, \mathcal{P}) -partial pattern */
 if the center of $SEC(\mathcal{Q}) \in Map(\mathcal{Q}, \mathcal{P})$
 then x := the center of $SEC(\mathcal{Q})$;
 else x := s ;
 endif
 $Final_Positions$:= $Map(\mathcal{Q}, \mathcal{P}) \setminus \{x\}$;
 if all the positions in $Final_Positions$ are occupied
 then **if** I am r_l
 then $MoveTo(x, \rightarrow)$;
 endif
 else k := the maximal k for which \mathcal{Q} is a (k, \mathcal{P}) -partial pattern;
 if there is at least one extra robot not located on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$
 then r := $Nearest_extra_robot(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, Map(\mathcal{Q}, \mathcal{P}))$;
 p := $Nearest_free_point(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, r)$;
 if I am r
 then $MoveTo(p, \rightarrow)$;
 endif
 else $Arrange(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, Final_Positions)$

the routine returns the extra robot, located on C , which is the closest in clockwise to the intersection between C and the half line $[c, r_l)$ (with r_l the leader in \mathcal{Q}).

$Nearest_free_point(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, r)$ returns the nearest position from r which is located on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ and not occupied by any robot belonging to \mathcal{Q} . If there is two nearest positions then the routines returns the position which is the closest in clockwise to the intersection between $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ and the half line $[c, r_l)$ (with c the center of $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ and r_l the leader).

$MoveTo(p, C, \odot)$ allows a robot to move toward a position p located on the circle C by moving along the boundary of C in clockwise. $MoveTo(p, C, \ominus)$ is similar but in counterclockwise.

$Arrange(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, Final_Positions)$ allows all the robots on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ to occupy all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Final_Positions$. The function is described by Algorithm 4 in which we use the following notions:

Definition 24 ($arc(p, p', C, \odot)$) Given a circle C and two points p and p' located on it, $arc(p, p', C, \odot)$ is the arc of circle C from p to p' in the clockwise direction, p being excluded (p' being included).

Definition 25 ($P\text{-}arc(p_i, p_{i+1}, C, \odot)$) Given a target pattern \mathcal{P} and an agreement configuration \mathcal{Q} , we say that $arc(p_i, p_{i+1}, C, \odot)$ is a $P\text{-}arc(p_i, p_{i+1}, C, \odot)$ if, and only if the three following properties holds:

1. C is one of the concentric enclosing circle of $\text{Map}(\mathcal{P}, \mathcal{Q})$
2. p_i and p_{i+1} belong to Final_Positions
3. $p_{i+1} = \text{adjacent}(p_i, C, \odot)$

Remark 26 From Definition 24, we know that p_i is not located on $P\text{-arc}(p_i, p_{i+1}, C, \odot)$.

In the remainder, we say that a P -arc is *free* if there is no robot located on it. In Figure 4, the circles denote the positions to achieve. The crosses depict the robots. The P -arc starting after f (f excluded) and finishing at a is free.

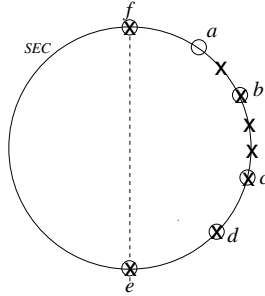


Figure 4: An example showing a Deadlock Chain and a Deadlock Breaker.

Definition 27 (Deadlock Chain) A *Deadlock Chain* is a consecutive sequence of P -arc starting from a free P -arc P_0 and followed in the counterclockwise direction by a P -arc P_1 such that:

1. P_1 is a $P\text{-arc}(p, p', C, \odot)$ such that $\text{angle}(p, c, p', \odot) = 180^\circ$ and there is only one robot r on it and r is located at p' ,
2. and P_1 is followed in counterclockwise by a consecutive sequence (possibly empty) of $P\text{-arc}(p, p', C, \odot)$ such that there is only one robot r on each of them and r is located at p' , and that consecutive sequence (possibly empty) is followed by a $P\text{-arc}(p, p', C, \odot)$ such that there is at least two robots on it and one of them is located at p' . This P -arc is called the last P -arc of the deadlock chain.

In Figure 4, the segment starting from Position a (a included) to Position b (b excluded) forms a deadlock chain.

Definition 28 (Deadlock Breaker) Let $P\text{-arc}(p, p', C, \odot)$ be the last P -arc of a deadlock chain. The *deadlock breaker* is the robot located at p' .

In Figure 4, the robot located at Position c is the deadlock breaker.

3.2.3 Sketch of Correctness Proof of Algorithm 1

We first show that by executing Algorithm 1, the smallest enclosing circle $\text{SEC}(\mathcal{Q})$ remains invariant—Lemma 30. Next, we prove that if the robots form a leader configuration which is not a final pattern \mathcal{P} and not an agreement configuration, they eventually form an agreement configuration—Lemma 31.

Algorithm 4 $Arrange(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, Final_Positions)$ executed by robot r_i on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$

```

/* I am  $r_i$  */
 $p :=$  the closest position in  $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Final\_Positions \setminus \{r_i\}$  to  $r_i$  in clockwise;
if  $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} = SEC(\mathcal{Q})$ 
then if there is no robot in  $arc(r_i, p, C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \odot)$  or I am a deadlock breaker
then if I am a deadlock breaker
then  $t :=$  the position such that  $angle(r_i, c, t, \odot) = \frac{1}{2}angle(r_i, c, p, \odot)$ ;
 $p := t$ ;
endif
 $r_{i-1} := adjacent(r_i, SEC, \odot)$ ;
 $p' :=$  the position such that  $angle(r_{i-1}, c, p', \odot) = 180^\circ$ ;
 $p'' :=$  the closest point to  $r_i$  in clockwise in  $\{p; p'\}$ ;
if  $r_i$  is not located at  $p''$ 
then  $MoveTo(p'', SEC, \odot)$ ;
endif
endif
else if there is no robot in  $arc(r_i, p, C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \odot)$ 
then  $MoveTo(p, C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \odot)$ ;
endif
endif

```

Starting from such a configuration, $Map(\mathcal{Q}, \mathcal{P})$ remains invariant or the target pattern \mathcal{P} is formed—Corollary 33. It follows that from an agreement configuration which is not a (k, \mathcal{P}) -partial pattern, the robots eventually form a $(0, \mathcal{P})$ -partial pattern—Lemma 34.

From this point on, note that according to Algorithm 3, $Final_Positions$ is equal to all the positions in $Map(\mathcal{P}, \mathcal{Q})$ except:

1. either the center c of $SEC(\mathcal{Q})$ if $c \in Map(\mathcal{P}, \mathcal{Q})$,
2. or the first non critical position located on the smallest concentric enclosing circle of $Map(\mathcal{P}, \mathcal{Q})$ if $c \notin Map(\mathcal{P}, \mathcal{Q})$

Next, we show by induction that, from a configuration being a maximal (k, \mathcal{P}) -partial pattern, the robots eventually form a $(k+1, \mathcal{P})$ -partial pattern or the target pattern \mathcal{P} is formed—Lemmas 35 to 37. From Lemma 37 and by induction we deduce the following theorem:

Theorem 29 *Starting from a leader configuration, Algorithm 1 allows to solve the pattern formation problem in $CORDA$ among a cohort of $n \geq 4$ robots having chirality and devoid of any kind of sense direction.*

4 Conclusion

We studied the relationship between the arbitrary formation problem and the leader election problem among robots having the chirality in $CORDA$. We gave an algorithm allowing to form an arbitrary pattern starting from any geometric configuration wherein the leader election is possible. Combined with the result in [9], we deduce that arbitrary pattern formation problem and Leader election are equivalent, *i.e.*, it is possible to solve the pattern formation problem for $n \geq 4$ if and only if the leader election is solvable too.

In a future work, we would like to investigate the same equivalence (1) for the case $n = 3$ and (2) in the case without chirality.

References

- [1] D Canepa and M Gradinariu Potop-Butucaru. Stabilizing flocking via leader election in robot networks. In *Ninth International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS '07)*, pages 52–66, 2007.
- [2] M Cieliebak and G Prencipe. Gathering autonomous mobile robots. In *9th International Colloquium on Structural Information and Communication Complexity (SIROCCO 9)*, pages 57–72, 2002.
- [3] Reuven Cohen and David Peleg. Local spreading algorithms for autonomous robot systems. *Theor. Comput. Sci.*, 399(1-2):71–82, 2008.
- [4] X Defago and A Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *2nd ACM International Annual Workshop on Principles of Mobile Computing (POMC 2002)*, pages 97–104, 2002.
- [5] Y Dieudonné and F Petit. Deterministic leader election in anonymous sensor networks without common coordinated system. In *Eleventh International Conference On Principles Of Distributed Systems (OPODIS 2007)*, 2007.
- [6] Yoann Dieudonné, Ouiddad Labbani-Igbida, and Franck Petit. Circle formation of weak mobile robots. *TAAS*, 3(4), 2008.
- [7] Yoann Dieudonné and Franck Petit. Squaring the circle with weak mobile robots. In *ISAAC*, pages 354–365, 2008.
- [8] P Flocchini, G Prencipe, N Santoro, and P Widmayer. Distributed coordination of a set of autonomous mobile robots. In *IEEE Intelligent Vehicle Symposium (IV 2000)*, pages 480–485, 2000.
- [9] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theor. Comput. Sci.*, 407(1-3):412–447, 2008.
- [10] B Katreniak. Biangular circle formation by asynchronous mobile robots. In *12th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, pages 185–199, 2005.
- [11] N Megiddo. Linear-time algorithms for linear programming in \mathbb{R}^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.
- [12] G Prencipe. Distributed coordination of a set of autonomous mobile robots. Technical Report TD-4/02, Dipartimento di Informatica, University of Pisa, 2002.
- [13] Giuseppe Prencipe. Instantaneous actions vs. full asynchronicity : Controlling and coordinating a set of autonomous mobile robots. In *ICTCS*, pages 154–171, 2001.
- [14] I Suzuki and M Yamashita. Agreement on a common x - y coordinate system by a group of mobile robots. *Intelligent Robots: Sensing, Modeling and Planning*, pages 305–321, 1996.
- [15] I Suzuki and M Yamashita. Distributed anonymous mobile robots - formation of geometric patterns. *SIAM Journal of Computing*, 28(4):1347–1363, 1999.
- [16] E Welzl. Smallest enclosing disks (balls and ellipsoids). In *Lecture Notes in Computer Science.*, volume 555, pages 359–370, 1991.
- [17] M Yamashita and I Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. Technical report, 2008.

Missing Proofs.

Lemma 30 *According to Algorithm 1, the smallest enclosing circle $SEC(\mathcal{Q})$ remains invariant.*

Proof. Assume by contradiction $SEC(\mathcal{Q})$ does not remain invariant. From Corollaries 10 and 14 and Property 8, we deduce that can occurs if, and only if:

- Either a robot r moves outside $SEC(\mathcal{Q})$. However, according to Algorithm 1, no robot moves outside $SEC(\mathcal{Q})$. That is a contradiction.
- Or an angle strictly greater than 180° appears between two adjacent robots r_{i-1} and r_i , i.e., $angle(r_{i-1}, c, r_i, \odot) > 180^\circ$ with $r_{i-1} = adjacent(r_i, SEC, \odot)$. This subcase can occur if, and only if
 - Either a critical robot leaves $SEC(\mathcal{Q})$. However, according to Algorithm 2 no critical robot leaves $SEC(\mathcal{Q})$ (only the first non-critical robot on SEC in clockwise is sometimes allowed to move). Furthermore, according to Algorithm 3 some robots are allowed to leave $SEC(\mathcal{Q})$ only if these latter ones are extra robots and \mathcal{Q} is a $(1, \mathcal{P})$ -partial pattern. That implies some robots are allowed to leave $SEC(\mathcal{Q}) = C_1^{\mathcal{Q}}$ only if these latter ones do not occupy a position $\in Map(\mathcal{Q}, \mathcal{P}) \cap SEC(\mathcal{Q})$ and all the positions in $Map(\mathcal{Q}, \mathcal{P}) \cap SEC(\mathcal{Q})$ are occupied by some robots. However, from Corollary 14 we know that for all couple of positions r_i and r_j on $Map(\mathcal{Q}, \mathcal{P}) \cap SEC(\mathcal{Q})$ such that $r_j = adjacent(r_i, \odot)$, we have $angle(r_i, c, r_j, \odot) \leq 180^\circ$. Consequently, when extra robots leaves $SEC(\mathcal{Q})$, $SEC(\mathcal{Q})$ is not changed. So, no critical robot leaves $SEC(\mathcal{Q})$. That is a contradiction.
 - Or two adjacent robots r_{i-1} and r_i , such that $r_{i-1} = adjacent(r_i, SEC, \odot)$, move along $SEC(\mathcal{Q})$ so that $angle(r_{i-1}, c, r_i, \odot) > 180^\circ$. That might occur only by applying Algorithm 4. However, if r_i is allowed to move, it can only move in clockwise towards a position p such that $angle(r_{i-1}, c, p, \odot) \leq 180^\circ$. Furthermore, r_{i-1} is never allowed to move in counterclockwise. So, $angle(r_{i-1}, c, r_i, \odot)$ is always less than or equal to 180° . That is a contradiction.

□

Lemma 31 *If the robots form a leader configuration which is not a final pattern \mathcal{P} and not an agreement configuration, they form an agreement configuration in a finite number of cycles.*

Proof. If the robots form a leader configuration which is not a final pattern \mathcal{P} and not an agreement configuration, then from Corollary 23 we have two cases to consider: either (1) the leader r_l is at the center c of SEC or (2) r_l is not the unique robot closest to c .

- **Case 1.** r_l is at the center of SEC . According to Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$, r_l moves away from c towards a position which is closer to the center than the second robot closer to the center. Furthermore, from Lemma 30, the center c of SEC remains invariant even if r_l moves. So, r_l remains the unique leader and, by fairness, we deduce that an agreement configuration is formed in a finite number of cycles.
- **Case 2.** r_l is not the unique robot closest to c . In that case, we have two subcases to consider:

- **Case 2.1.** r_l is not a critical robot. In this subcase, r_l moves towards a position which is located between c and itself (except c and itself). From Lemma 30, the center c of SEC remains invariant even if r_l moves. So, by fairness we know that an agreement configuration is formed in a finite number of cycles.
- **Case 2.2.** r_l is a critical robot. From Corollary 10, r_l is on the circumference of SEC . However, by assumption r_l is also one of the robots closest to the center of SEC . So, we deduce that all the robots are on SEC . Hence, by Lemma 15, we deduce there is at least one non-critical robot on SEC because there are at least four robots on it (recall that we assume the number of robot is greater than or equal to 4).

According to Procedure $\langle \text{Leader} \rightsquigarrow \text{Agreement} \rangle$ the first non-critical robot r_k starting from r_l on SEC in clockwise is allowed to move toward a position located between itself and c (except c and itself). From Lemma 30, the center c of SEC remains invariant even if r_k moves. So, by fairness r_k becomes the unique robot closest to c and it is not located at c . So, the robots form an agreement configuration in a finite number of cycles.

□

Lemma 32 *Starting from an agreement configuration \mathcal{Q} , the robots remains in an equivalent agreement configuration or the target pattern \mathcal{P} is formed in a finite number of cycles.*

Proof. According to Lemma 30, $SEC(\mathcal{Q})$ and its center c remain invariant. Moreover, according to Algorithm 1 and more precisely Algorithm 3 no robot is allowed to pass r_l .

So, if r_l is not allowed to move then, according to Definition 19 all the robots remain in an equivalent agreement configuration.

If r_l is allowed to move then, according to Algorithm 3 that can occur only in three cases:

- Case 1. The robots do not form any (k, \mathcal{P}) -partial pattern. In that case, r_l moves in straight line towards the middle p of the segment $[c, s]$ in order to get closer to the center c . However, from Definition 20, we know that s is on the half line $[c, r_l)$. So, during the motion of r_l , all the robots clearly remain in an equivalent agreement configuration.
- Case 2. The center c of $SEC(\mathcal{Q})$ is in $Map(\mathcal{Q}, \mathcal{P})$ and all the positions in $Map(\mathcal{Q}, \mathcal{P})$ are occupied except c . In that case, r_l chooses to move towards c in straight line (i.e., along $[c, r_l)$) in order to occupy the last free position in $Map(\mathcal{Q}, \mathcal{P})$. Until r_l has not reached c , the robots remain in an equivalent agreement configuration because r_l is still on the same half line $[c, r_l)$ and it remains the unique robot closest to c . So by fairness, it reaches c in a finite number of cycle and the pattern \mathcal{P} is formed.
- Case 3. The center c of $SEC(\mathcal{Q})$ is not in $Map(\mathcal{Q}, \mathcal{P})$ and all the positions in $Map(\mathcal{Q}, \mathcal{P})$ are occupied except the first non critical position s located on the smallest concentric enclosing circle. In that case, r_l chooses to move towards s in order to occupy the last free position in $Map(\mathcal{Q}, \mathcal{P})$. From Definition 20, we know that s is on the half line $[c, r_l)$ and thus, until r_l has not reached s the robots remain in an equivalent agreement configuration because r_l is still on the same half line $[c, r_l)$ and it remains the unique robot closest to c . So by fairness, it reaches s in a finite number of cycle and the pattern \mathcal{P} is formed.

□

Corollary 33 *From an agreement configuration, $Map(\mathcal{Q}, \mathcal{P})$ remains invariant or the target pattern \mathcal{P} is formed.*

Lemma 34 *From an agreement configuration which is not a (k, \mathcal{P}) -partial pattern, the robots form a $(0, \mathcal{P})$ -partial pattern in a finite number of cycles.*

Proof. From Definition 21, we know that if an agreement configuration is not a $(0, \mathcal{P})$ -partial pattern then, the leader r_l is not inside the smallest concentric enclosing circle of $Map(\mathcal{Q}, \mathcal{P})$. From Corollary 33 and according to Algorithm 3, r_l is inside the smallest concentric enclosing circle of $Map(\mathcal{Q}, \mathcal{P})$ in a finite number of cycles. \square

Lemma 35 *Let \mathcal{P} be a target pattern and let \mathcal{Q} be a configuration which is a maximal (k, \mathcal{P}) -partial pattern such that $1 \leq k < |\mathcal{SC}^{\mathcal{P}}|$. If all the extra robots are on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ then, all the positions in $Final_Positions \cap C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ are occupied in a finite number of cycles.*

Proof. If all the extra robots are on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ and there exists at least one position in $Final_Positions \cap C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ which is not occupied then the robots apply the routine $Arrange(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, Final_Positions)$ (refer to Algorithm 4). Remark that by applying this routine, no robot can collide another robot since any robot can move only in clockwise and any move of a robot on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ is only allowed in arc of circle containing no robot. Moreover, since $k \geq 1$, $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \neq SEC(\mathcal{Q})$ and thus, it is no need to prevent from creating an angle strictly greater than 180° between two adjacent robots. In the remainder of this proof, we denote by α the number of extra robots located on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$, β the number of P -arc on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ and γ the number of free P -arc on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$. According to Algorithm 2, the acute reader noticed that the number α of extra robots is greater than or equal to the number β of P -arc on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$.

We consider two cases.

- **All the P -arcs are not free.** According to Algorithm 4, each last robot on each P -arc (p_i, p_{i+1}, C, \odot) is allowed to move to p_{i+1} if it is not yet at this position. At the end of these motions, all the positions in $Final_Positions \cap C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ are occupied and remains occupied.
- **At least one P -arc is free.** In that case we have $1 \leq \gamma < \beta$. According to Algorithm 4, if a robot moves from a P -arc to another one then γ does not decrease because if robot r chooses to move from a P_1 -arc to a P_2 -arc, that implies that P_2 -arc is free. However, if P_1 becomes free when r reaches P_2 then the number of free P -arc remains unchanged.

We now assume by contradiction that γ never reaches the value β . So γ eventually remains unchanged. From this point on, no robot of any P -arc containing more than one robot will move towards a free P -arc. Following the algorithm, that implies that every P -arc P with more than one robot is followed in clockwise by a non free P -arc infinitely often (at least each time the last robot of P is awaked). Since the robots cannot move in counterclockwise, that also implies that every P -arc with more than one robot is always followed by a non free P -arc P' . If this second P -arc P' also contains more than one robot then it is also followed by a non free P -arc. However, if P' contains only one robot then it is also followed by a non free P -arc

P'' , since on the contrary, the robot of P' will eventually move to P'' and the last robot of P will eventually move to P' . A contradiction. So, step by step, it is clear that no P -arc can be free and $\gamma = \beta$ which contradicts our assumption. So γ will eventually reach the value β .

When $\gamma = \beta$ we retrieve the case where all the P -arcs are not free and the lemma holds. □

Lemma 36 *Let \mathcal{P} be a target pattern and let \mathcal{Q} be a configuration which is a maximal $(0, \mathcal{P})$ -partial pattern. If all the extra robots are on $SEC(Map(\mathcal{Q}, \mathcal{P}))$ then, all the positions in $Final_Positions \cap C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ are occupied in a finite number of cycles.*

Proof. If all the extra robots are on $SEC(Map(\mathcal{Q}, \mathcal{P}))$ and there exists at least one position in $Final_Positions \cap SEC(Map(\mathcal{Q}, \mathcal{P}))$ which is not occupied then the robots apply the routine $Arrange(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, Final_Positions)$ (refer to Algorithm 4) for $k = 1$. Despite a more complicated code, the case $k = 1$ can be seen as the case $k > 1$ with an additional constraint on the angles and a particular statement for a deadlock configuration removal. We show (refer to last item of this proof) that the deadlock removal generates a behavior that can finally be generated by Algorithm 4 for a concentric enclosing circle which is not $SEC(\mathcal{Q})$. So the aim of the proof is to show that Algorithm 4 has no deadlock. In the rest of this proof we say that a point p is a P -point if $p \in Final_Positions \cap SEC(Map(\mathcal{Q}, \mathcal{P}))$. So assume by contradiction that there exists a deadlock and we consider the two following cases:

1. No P – arc is free but there exists at least one P -point which is not occupied by a robot. Again, we distinguish two cases:
 - (a) At least one P -point is occupied by a robot. Let p_i be one these P -point such that its successor in clockwise $p_{(i+1)}$ is free. Clearly, $angle(p_i, c, p_{(i+1)}, \odot) \leq 180$ (even if the first non critical s does not belong to $Final_Positions$ because, due to the fact s is not critical, from Lemma13 its absence cannot create an angle > 180). So the last robot of the $p_{(i+1)}$ P – arc can move to $p_{(i+1)}$. A contradiction.
 - (b) No P -point is occupied by a robot. Since there are at least three robots on SEQ , at least one of them has a predecessor with an angle less than 180° . So it can move. A contradiction.
2. There exists at least one free P – arc. Let i ($0 \leq i \leq \alpha - 1$) be an integer such that the i th P – arc is free and its predecessor (the $((i - 1) \bmod \alpha)^{th}$ P – arc) is not. Let us call them A and A' , respectively. We distinguish two cases:
 - (a) There exists (A', A) such that A' contains at least two robots. We call r the last robot of A' and r' the predecessor of r on A' . In this case r can move to A (since $angle(r', c, r, \odot) < angle(p_{(i-2) \bmod \alpha}, c, p_{(i-1) \bmod \alpha}, \odot) \leq 180^\circ$). A contradiction.
 - (b) Every couple (A', A) is such that A' contains one robot only. In that case there exists at least a couple (A', A) such that the predecessor A'' of A' contains at least one robot since there are at least as many robots as P – points on SEQ . Because the deadlock assumption, the robot r on A' cannot move to A so $angle(r', c, r, \odot) = 180^\circ$ where r' is the last robot on A'' . Again, we distinguish two cases:

- i. r is not on $p_{(i-1) \bmod \alpha}$. In this case r' also is not on $p_{(i-2) \bmod \alpha}$ since $\text{angle}(p_{(i-2) \bmod \alpha}, c, r, \odot) < 180$. Since there is at least a third robot on SEQ , this robot r'' is such that $\text{angle}(r'', c, r', \odot) < 180$ so r' can move toward $p_{(i-2) \bmod \alpha}$. A contradiction.
- ii. r is on $p_{(i-1) \bmod \alpha}$. In this case r' is also on $p_{(i-2) \bmod \alpha}$ and $\text{angle}(p_{(i-2) \bmod \alpha}, c, p_{(i-1) \bmod \alpha}, \odot) = 180$. Since no robot can move, we can see that the configuration on SEQ is as follows: A is the first P -arc of a chain starting from A in the counterclockwise such that any P -arc of this chain but A contains a robot at its P -point, we call this chain PC . The last P -arc of PC is followed by a free P -arc (A if there exists no other free P -arc). Since no robot can move we can deduce that between this free P -arc and A (in the counterclockwise) all the P -arcs are free. So all the robots are on PC and there exists at least one P -arc of PC which contains at least two robots. Let B be the first P -arc of the chain (starting from A in counterclockwise) such that B contains at least two robots. Then the chain starting from A and ending to B is a deadlock chain. By definition, the robot on the P -point of B is a deadlock breaker and can move. A contradiction.

Now we just focus on the behavior of the successive deadlock breakers. The aim of their behavior is to allow r to move toward the next P -point. It is easy to see that this part of the algorithm just reverses the order of the deadlock breakers and r , but once one of these robots has started to move their behavior is the same as in the internal circle part (still with angle constraint).

□

Lemma 37 *Let \mathcal{P} be a target pattern and let \mathcal{Q} be a configuration which is a maximal (k, \mathcal{P}) -partial pattern. The robots form a $(k+1, \mathcal{P})$ -partial pattern or the target pattern is formed, in a finite number of cycles.*

Proof. We have to consider three cases.

- $k = |\mathcal{SC}^{Map(\mathcal{Q}, \mathcal{P})}|$. In that case, $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ does not exist and $\bigcup_{i=1}^{|\mathcal{SC}^{Map(\mathcal{Q}, \mathcal{P})}|} C_i^{\mathcal{Q}} \cap \mathcal{Q} = \bigcup_{i=1}^{|\mathcal{SC}^{Map(\mathcal{Q}, \mathcal{P})}|} C_i^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$. That implies that it remains only one position p to occupy and p is inevitably at the center of $SEC(\mathcal{Q})$ (otherwise $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ would exist). According to Algorithm 2, leader r_l moves toward c . From Corollary 33 and by fairness, we deduce that the target pattern is formed in a finite number of cycles.
- $k = |\mathcal{SC}^{Map(\mathcal{Q}, \mathcal{P})}| - 1$. In that case, we distinguish two subcases:
 1. The center c of $SEC(\mathcal{Q})$ is in $Map(\mathcal{Q}, \mathcal{P})$. In that subcase, all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ must be occupied by all the extra robots even the first non critical position. According to Algorithm 2, the extra robots move to the boundary of $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ by using subroutines $Nearest_extra_robot(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, Map(\mathcal{Q}, \mathcal{P}))$ and $Nearest_free_point(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, r)$. These subroutines assure us that the extra robots moves one by one toward a position on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ which is not occupied by any robot. Of course, if we are lucky, a $(k+1, \mathcal{P})$ -partial pattern is formed during this step. Otherwise, the robots apply Algorithm 4 and, from Lemma 35 and 36, the $(k+1, \mathcal{P})$ -partial pattern is formed in a finite number of cycles.

2. The center c of $SEC(\mathcal{Q})$ is not in $Map(\mathcal{Q}, \mathcal{P})$. In that subcase, all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ must be occupied by all the extra robots except the first non critical position on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ which is booked for the leader. According to Algorithm 2, the extra robots move to the boundary of $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ by using subroutines $Nearest_extra_robot(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, Map(\mathcal{Q}, \mathcal{P}))$ and $Nearest_free_point(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, r)$. During this step, if we are lucky, all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ are occupied by all the extra robots except the first non critical position on $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$. Otherwise, the robots apply Algorithm 4 and, from Lemmas 35 and 36 all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ are eventually occupied except the first non critical position. From this point now, according to Algorithm 2 leader r_l moves towards the first non critical position in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ and from Corollary 33 and fairness we deduce that the target pattern is formed in a finite number of cycles.
- $k < |\mathcal{S}C^{Map(\mathcal{Q}, \mathcal{P})}| - 1$. In that subcase, all the positions in $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})} \cap Map(\mathcal{Q}, \mathcal{P})$ must be occupied by all the extra robots. According to Algorithm 2, the extra robots move to the boundary of $C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}$ by using subroutines $Nearest_extra_robot(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, Map(\mathcal{Q}, \mathcal{P}))$ and $Nearest_free_point(C_{k+1}^{Map(\mathcal{Q}, \mathcal{P})}, \mathcal{Q}, r)$. If we are lucky, a $(k + 1, \mathcal{P})$ -partial pattern is formed during this step. Otherwise, the robots apply Algorithm 4 and, from Lemma 35 and 36, the $(k + 1, \mathcal{P})$ -partial pattern is formed in a finite number of cycles.

□