# Leader Election Problem Versus Pattern Formation Problem

Yoann Dieudonné, Franck Petit, Vincent Villain

# Leader Election Problem Versus Pattern Formation Problem

Yoann Dieudonné[†], Franck Petit[‡] and Vincent Villain[†]
[†] MIS Lab., University of Picardie Jules Verne, France
[‡] INRIA, LIP Lab., University of Lyon, France

**Abstract**

Leader election and arbitrary pattern formation are fundammental 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 the semi-synchronous model (SSM), wherein the robots are weak in several aspects. In particular, 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, we show that both problems are equivalent, that is it is possible to solve the pattern formation problem for $n \neq 2$ if and only if the leader election is solvable too.

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

## 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 of 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 have. Examples of such global properties are the ability to distinguish between them 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 faisibility 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 [12, 9, 1, 6], most of the studied tasks are geometric problems, so that *pattern formation*, *line formation*, *gathering*, and *circle formation*—refer to [15, 9, 8, 4, 10, 5, 3] for this 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 were all entities are in the same state into a final configuration were 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 rafine 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 [6], 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 this paper, we provide a positive answer to this question assuming that robots have the chirality property and the semi-synchronous model (SSM). We show that Leader Election and Pattern Formation are two *equivalent* problems, in the precise sense that, the former problem is solvable if and only the latter problem is solvable. We show that the equivalence is true for any number $n$ of robots, except if $n = 2$.

The result is mainly based on two pattern formation protocols. The first one is a non-trivial protocol working for any $n \geq 4$. The difficulty in the design of our protocol comes from the following facts: (1) The protocol is based on the result in [6] that provides a leader among the robots. (2) We then built a global coordinate system based on the leader position. (3) Both the leader and the coordinate system must be kept as an invariant until the desired pattern is eventually formed. The second protocol works for the special case $n = 3$, only.

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 $n \geq 4$. Section 4 tackles the case $n = 3$. Finally, we make concluding remarks in Section 5.

## 2 Preliminaries

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

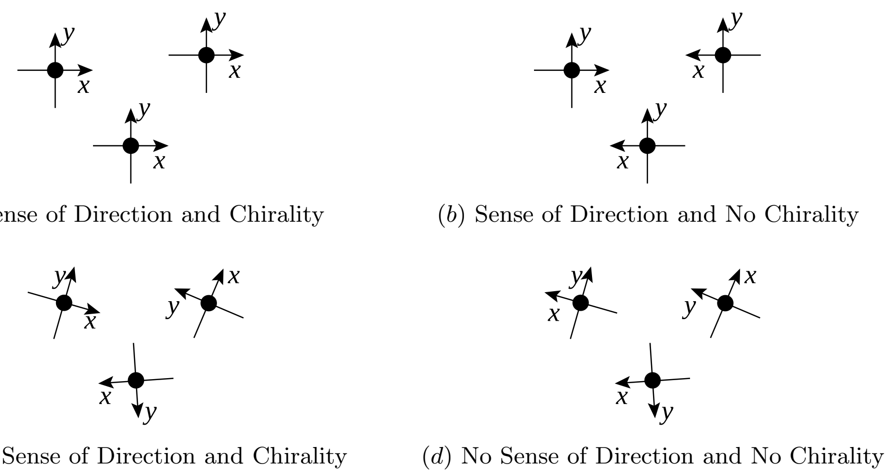
**Distributed Model.** We adopt the Semi-Synchronous Model (SSM) of [14]. The *distributed system* considered in this paper consists of $n$ robots $r_1, r_2, \cdots, r_n$—the subscripts $1, \ldots, n$ are used for notational purpose only. Each robot $r_i$, viewed as a point in the Euclidean plane, move 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. It is assumed that the robots never collide and that two or more robots may simultaneously occupy the same physical location. Any robot can observe, compute and move with infinite decimal precision. The robots are equipped with sensors allowing to detect the instantaneous position of the other robots in the plane. 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 —ie. 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.*



$(a)$ Sense of Direction and Chirality

$(b)$ Sense of Direction and No Chirality

$(c)$ No Sense of Direction and Chirality

$(d)$ No Sense of Direction and No Chirality

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 $(d)$.

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. They communicate only by observing the position of the others and they are *oblivious*, i.e., none of them can remember any previous observation nor computation performed in any previous step.

Time is represented as an infinite sequence of time instant $t_0, t_1, \ldots, t_j, \ldots$ 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 the latter case, $r_i$ does not perform any local computation and remains at the same position.

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 (non empty) set of robots. Fairness means that every robot is infinitely often activated by the scheduler.

**The 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.

**The Arbitrary Pattern Formation Problem** In the *Arbitrary Pattern Formation Problem*, the robots are given in input the same pattern, described as a set of points in the plane (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 points of the pattern, where the input pattern may be translated, rotated, scaled, and flipped into its mirror position 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 points prescribed in the pattern and the number of robots are the same.

## 3 Proof of Equivalence

In this section, we present the main result of this paper:

**Theorem 3** *In SSM, assuming a cohort of $n \neq 2$ anonymous, oblivious robots having chirality and devoid of any kind of sense of direction, the leader election problem is solvable if and only if the pattern formation problem is solvable.*

To prove Theorem 3, we first borrow the following theorem from [12]:

**Theorem 4** *[12] If it is possible to solve the pattern formation problem for $n \geq 3$ robots, then the leader election problem is solvable too.*

The proof of Theorems 4 is given in a model, called Corda [12], which allows more asynchrony among the robots than the model SSM used in this paper. However, we borrow the following result from [12]:

**Theorem 5** *[13] Any algorithm that correctly solves a problem P in Corda, correctly solves P in SSM.*

It follows from Theorem 5 that Theorem 4 is also true in SSM. Thus, it remains to show the following theorem:

**Theorem 6** *In SSM, assuming a cohort of $n \neq 2$ anonymous, oblivious 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 the remainder of this section, we provide a constructive proof of Theorem 6 for $n \geq 4$. The case $n = 3$ is left in the next section. We first provide definitions and basic properties related to both problems. Next, we present the protocol with its proof.

## 3.1 Definitions and Basics properties

Given a configuration $P(t_j)$, $SEC(t_j)$ (or $SEC$ when no ambiguity arises) denotes the *smallest enclosing circle* of the positions in $P(t_j)$ at time $t_j$. The center of $SEC$ is denoted $c$. In any configuration $P(t_j)$, $SEC$ is unique and can be computed in linear time [11].

**Remark 7** *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 points that are inside it. SEC does not change by adding points on its boundary, however it may be possible that SEC change by eliminating points on its circumference. Note that if $n = 2$, then SEC passes both positions and no robot can be located inside SEC, in particular at $c$.*

**Remark 8** *Since the robots have the ability of chirality, they are able to agree on a common orientation of SEC, denoted $\circlearrowright$.*

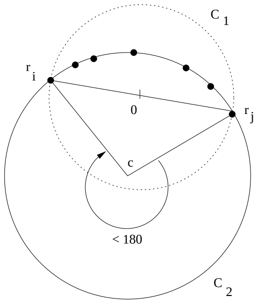
Consider a configuration at time $t_k$ ($k \geq 0$) in which the positions of the $n$ robots are located at distinct positions on the circumference of a circle $C$—the radius of $C$ is greater than *zero*. At time $t_k$, the *successor* $r_j$, $j \in 1 \dots n$, of any robot $r_i$, $i \in 1 \dots n$ 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. $\widehat{r_iOr_j}$ denotes the angle centered in $O$ and with sides the half-lines $[O, r_i)$ and $[O, r_j)$ such that no robots (other than $r_i$ and $r_j$) is on $C$ inside $\widehat{r_iOr_j}$.

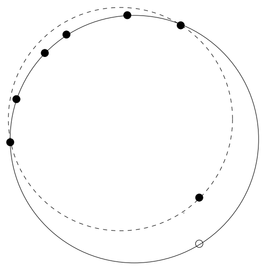
The following properties are fundammental results about smallest enclosing circles:

**Definition 9** *[2] A robot (or point) is critical if its deletion from SEC modifies this one.*

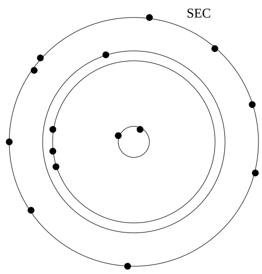
An example of any critical robot is given by Figure 2, Case $(b)$.




($a$) The circle $C_1$ is an enclosing circle smaller than $C_2$.



($b$) Critical (white) robot cannot be deleted without changing $SEC$.




($c$) A set of concentric enclosing circles.

Figure 2: Some examples illustrating Corollary 11, Definition 9, and Definition 13.

**Lemma 10** *[12] Let $r_i, r_j$ and $r_k$ be three robots on SEC such that $r_j$ is adjacent to $r_i$ and $r_k$. If $r_i$ and $r_k$ become adjacent by eliminating $r_j$ and $\widehat{r_i O r_k} \leq 180$, then $r_j$ is non-critical and SEC does not change.*

**Corollary 11** *Let SEC be the smallest circle enclosing n points on the plane, and Rad(SEC) its radius. For all adjacent points (or robots) $(r_i, r_j)$ on C, we have $\widehat{r_i O r_j} \leq 180$*

An example illustrating Corollary 11 is shown in Figure 2, Case ($a$).

**Lemma 12** *[2] Given a smallest enclosing circle with at least four robots on it. There exists at least one robot which is not critical.*

**Definition 13 (Concentric Enclosing Circle)** *Given a set S of distinct points. We say that C is a concentric enclosing circle if and only if C is centered at the center c of SEC, has a radius strictly greater than zero and C passing through at least one point in S.*

An example of concentric circle is given by Figure 2, Case ($c$).

Remark that from Definition 13, $SEC$ is the greatest concentric enclosing circle of $S$.

We say that the robots have the possibility to elect a leader if they are in a *leader configuration.* In such a configuration, we have the following corollary:

**Corollary 14** *[6] If the robots are in a leader configuration, then they can distinguish a unique robot which is one of the robots the closest to the center of the smallest enclosing circle of the configuration, provided that they share the property of chirality.*

We now define a description scheme of a pattern. This description scheme is specialy adapted to the algorithm we give in the next section. The reader can easily see that for any pattern given in any classical description scheme (for exemple as a set of vertices with cartesian coordonates) there exists at least one description in our scheme.

**Definition 15 (Description scheme)** *A pattern $P$ is the 4-uple $(VS, CS, v_0, VA)$ where:*

- *$VS$ is the set of vertices of $P$*
- *$CS$ is the pair $(VCS, RS)$ where:*
  - *$VCS$ is the set of all the concentric enclosing circles $C_0 \dots C_\alpha$ such that $C_0 = SEC$ and $VCS$ is a partition of $VS$ if $O \notin VS$, or $VS \setminus \{O\}$, otherwise;*
  - *$RS$ is the set of radii $\rho_0 \dots \rho_\alpha$, one for each $C_i$, such that $1 = \rho_0 > \dots > \rho_\alpha > 0$;*
- *$v_0$ is an arbitrary vertex of $C_\alpha$;*
- *$VA$ is the set of the $\sharp VS - 1$ angles $\widehat{v_0Ov}$ where $v$ is any vertex $\neq v_0$ of $VS$.*

**Definition 16 ($(k, P)$-sub pattern)** *Given a pattern $P = (VS, CS, v_0, VA)$, a $(k, P)$-sub pattern $(VS', CS', v'_0, VA')$ is the pattern defined as following:*

- *$CS'$ is the pair $(VCS', RS')$ where:*
  - *$VCS'$ is the set of $C_0, C_1 \dots C_k$;*
  - *$RS'$ is the set of corresponding radii $\rho_0$, $\rho_1 \dots \rho_k$;*
- *$VS' = \bigcup_{i=0}^{k} C_i$;*
- *$v'_0$ is an arbitrary vertex of $C_k$;*
- *$VA'$ is the subset of $VA$ reduced to the vertices of $VS'$.*

**Definition 17 ($(k, P)$-partial pattern)** *Given a pattern $P = (VS, CS, v_0, VA)$, we say that a pattern $Q = (VS', CS', v_0, VA')$ is a $(k, P)$-partial pattern ($0 \leq k \leq \alpha$) iff $\exists v \in C_k$ such that the $(k, P)$-sub pattern $(VS'', CS'', v, VA'')$ is equal to $Q$.*

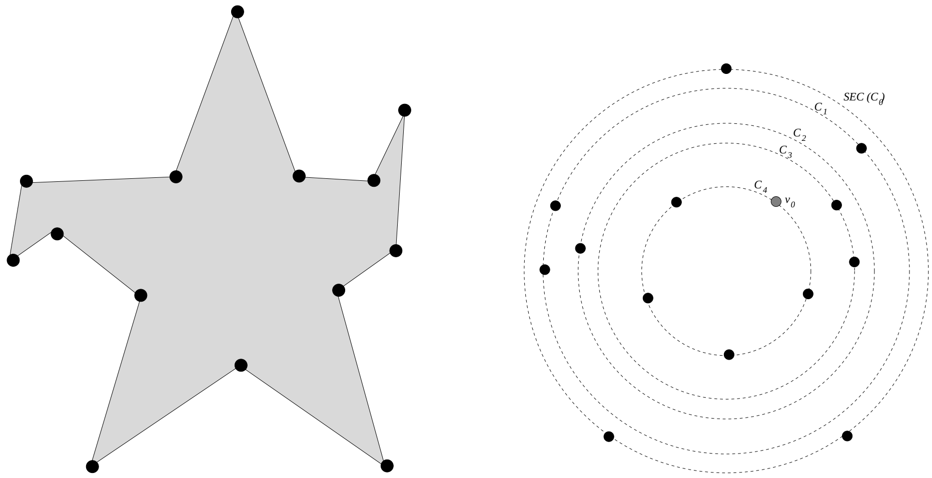
*If for any $k$ ($0 \leq k \leq \alpha$) $Q$ is not a $(k, P)$-partial pattern, we say that $Q$ is a $(-1, P)$-partial pattern.*

**Definition 18 (Equality of two patterns)** *Let $P$ and $Q$ two patterns, $P = (VS, CS, v_0, VA)$ and $Q = (VS', CS', v'_0, VA')$. We say that $P = Q$ iff the three following properties hold:*

- *$\sharp VS = \sharp VS'$;*
- *$Q$ is a $(\alpha, P)$-partial pattern configuration;*
- *$(O \notin VS) \Leftrightarrow (O \notin VS')$.*

In the sequel, we consider that $P$ is the pattern that any robot has to form and $Q$ is the current pattern or configuration formed by the robots. Moreover if we say that a group of robots or $Q$ forms a $(k, P)$-partial pattern configuration, then we assume that $k$ is the greatest possible.

We call $Q_k$ the subset of robots which are at the positions of the $(k, P)$-partial pattern.




($a$) Pattern target. ($b$) The pattern described using concentric enclosing circles.

Figure 3: An example showing our method to describe pattern.

### 3.2 The Protocol for $n \geq 4$ robots

In this subsection, we present an algorithm that leads the robots into a (arbitrary) given pattern by starting from a leader configuration. The main idea consists in forming the desired pattern by placing the robots to the final positions, concentric enclosing circle by concentric enclosing circle, starting from the greatest to the smallest. In order to achieve this strategy, the robots implicitly construct a commun cooordinate system, using $SEC$ and the leader, that we must maintain as invariant until the desired pattern is formed. The difficulty arises from the fact that it's not trivial to maintain $SEC$ as invariant due to the motions and the obliviousness of all the robots.

Before anything else, we say that the system is in a *Leader* configuration if the configuration allows the election of a leader. The algorithm we present statisfies the following property: starting from any *Leader* configuration, any reachable configuration is a *Leader* configuration or the system forms the pattern (in this last case the configuration can be totally symetric).

We first define some special robot configurations which will be used for building the steps of our algorithm.

**Definition 19 (*Leader*-$P$ configuration)** *We say that a robot configuration $Q$ is a Leader-$P$ configuration iff the leader $r_l$ is the single robot closest to $c$ and $0 < d(r_l, c) < \rho_\alpha$ where $d(x, y)$ denotes the distance between $x$ and $y$.*

**Definition 20 (*Pre*-$(k, P)$ configuration)** *Let $P = (VS, CS, v_0, VA)$ be a given pattern and $Q$ be a robot configuration. $Q$ is a Pre-$(k, P)$ configuration with $0 \leq k \leq \alpha$ iff $Q$ is a $(k-1, P)$-partial pattern configuration and all the robots $\notin Q_k \cup \{r_l\}$ are at a distance $k$ to $c$.*

**Definition 21 ($(k, P)$ configuration)** *Let $P = (VS, CS, v_0, VA)$ be a given pattern and $Q$ be a robot configuration. $Q$ is a $(k, P)$ configuration with $0 \leq k \leq \alpha$ iff $Q$ is a Pre-$(k-1, P)$ configuration and a robot is on each position of $C_k \setminus \{v_0\}$.*

In the presentation of the algorithm we first omit the case $v_0$ is a critical point of $P$. We will just add a comment at the end of this section. So, the algorithm is as follows.

$Q$ **is not a** *Leader*-$P$ **configuration.** First, we know that, in a *Leader* configuration, we can distinguish a unique robot $r_l$, called the leader, which is one of the robots closest to the center $c$ of $SEC$ (refer to Corollary 14). If the leader is not the single robot closest to $c$ or $d(r_l, c) \notin ]0, \rho_\alpha[$ then

- if the leader is not a critical robot, then it moves toward $c$ to a position located between $c$ and itself (except $c$ and itself) at a distance of $c < \rho_\alpha$. From Remark 7, $SEC$ remains invariant and the leader is eventually the single robot closest to $c$.
- else all the robots are closest to $c$ (i.e. all the robots are on $SEC$) and the leader is a critical robot. If it moves, the $SEC$ will change. So in that case, the first non critical robot $r_i$ starting from $r_l$ according to the common orientation, refer to Remark 8, of $SEC$ moves toward $c$ to a position located between $c$ and itself (except $c$ and itself). Note that from Lemma 12, $r_i$ exists. So, $r_i$ eventually becomes the new leader $r_l$ which is the closest robot to $c$ at a distance less than $\rho_\alpha$.

When the leader $r_l$ is the single robot the closest to $c$, if $d(r_l, c) = 0$ or $d(r_l, c) = \rho_\alpha$ then $r_l$ moves along a radius to a distance $= \frac{\rho_\alpha}{2}$ of $c$.

So starting from there, all the robots will consider $r_l$ as the robot which corresponds to $v_0$ and will able to compute coherent positions for pattern $P$.

$Q$ **is a** $(k, P)$ **configuration but not a** *Pre*-$(k+1, P)$ **configuration** $(-1 \leq k < \alpha)$**.** In this case all the robots which are not in $Q_k \cup \{r_l\}$ move at a distance $\rho_k$ of $c$, by executing an algorithm similar to Algorithm $\phi_{circle}$. The correctness of this algorithm is shown in [4]. The only difficulty we can meet is when the robots have to move toward $SEC$ $(k = -1)$. But in that case the distance between each robot and the center of $SEC$ never decreases. So, during this step, $SEC$ remains invariant since only the robots inside it are allowed to move (see Remark 7). Furthermore, $r_l$ remains the single robot closest to $c$ and thus, the leader.

In any case, the robots eventually form a *Pre*-$(k+1, P)$ configuration.

$Q$ **is a** *Pre*-$(k, P)$ **configuration but not a** $(k, P)$ **configuration** $(0 \leq k \leq \alpha)$**.** We distinguish the case $k = 0$ from the other ones because we need to maintain SEC as an invariant. We call ${C^Q}_k$ the circle of radius $\rho_k$ and also the set of robots which are on ${C^Q}_k$. $C_k$ still is the set of positions of $P$ which are at a distance $\rho_k$ of the center.

1. $k = 0$. In this case ${C^Q}_0$ is $SEC$. Let $X$ be the first robot from the intersection of $SEC$ with the half line $[c, r_l)$ and $Y$ the opposite end of the $SEC$ diameter from $X$. We now call $C_X$ (resp. $C_Y$) the half circle $[X, Y[$ (resp. $[Y, X[$) of $SEC$ following the chirality which contains X (resp. $Y$) but not Y (resp. $X$). We call $p_X$ (resp. $p_Y$) the nearest position of $P$ from $X$ (resp. $Y$). We call $C_{kX}$ (resp. $C_{kY}$) the set of positions of $C_k$ reduced to the half circle $C_X$ (resp. $C_Y$) but $p_X$ (resp. $p_Y$).
   (a) The number of robots on each half circle is at least equal to the number of the corresponding positions of each half circle of ${C^Q}_0$. In this case, the robots on $C_X$ (resp. $C_Y$) apply $\phi_{deploy}(C_X, C_{kX})$ (resp. $\phi_{deploy}(C_Y, C_{kY})$). At the end of these two executions of Algorithm $\phi_{deploy}$ all the positions of $C_k$ but $p_X$ and $p_Y$ are occupied by a robot.

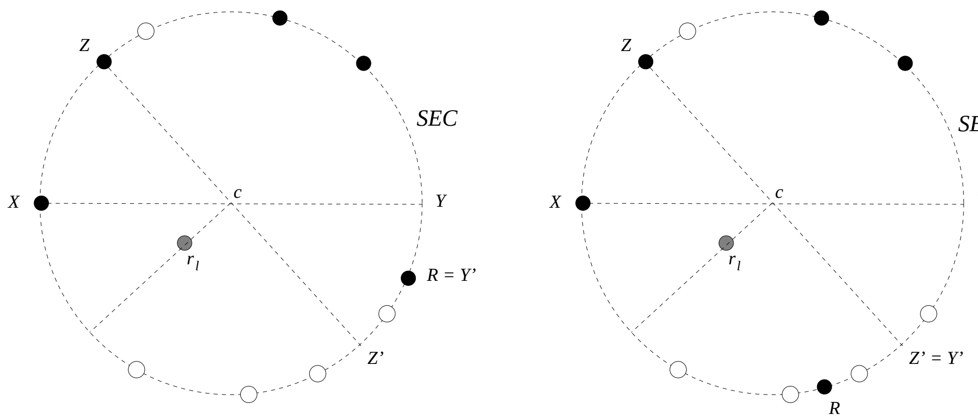



Figure 4: Two examples of $Pre$-$(0, P)$ configurations considered in Case 1-$(b)$.

(b) The number of robots of $C_Y$ is less than the number of the positions of the corresponding half circle of ${C^Q}_0$. Let $g$ be the gap between the number of robots and the number of positions in $C_Y$. Let $Z$ be the first robot of the $g$ last robots on $C_X$ after $X$. Clearly, $Z \neq X$ since by definition of $SEC$, there exists at least a position of $P$ on $C_X$. Let $Z'$ be the second end of the diameter from $Z$ and $R$ be the robot at the minimal distance $> 0$ from $Y$. Let $Y'$ be the position on $C_Y$ corresponding to the nearest of $Z'$ and $R$ from $Y$—refer to Figure 4. If a robot is on $Y$, it moves to a position which is at the distance $\frac{d(Y,Y')}{2^2}$ from $Y$ and this position is designed as the new $Y'$. The $g-1$ last robots of $C - X$ move sequentially along $SEC$ to a position equals to $\frac{d(Y,Y')}{2}$ for the last one, then to $\frac{d(Y,Y')}{2^2}$ for the previous one and so on. Finally $Z$ moves to $Y$.

During these motions $SEC$ is invariant since the motions never create any angle $> 180$. Finally the robots apply $\phi_{deploy}(C_X, C_{kX})$ and $\phi_{deploy}(C_Y, C_{kY})$ as in the first case.

(c) The number of robots of $C_X$ is less than the number of the positions of the corresponding half circle of ${C^Q}_0$. This case is similar to the previous one by moving the robots in the counter direction from $C_Y$ to $C_X$.

Now, at least the positions of $C_k \setminus \{p_X, p_Y\}$ are occupied by a robot. We have three cases:

(a) $p_X$ and $p_Y$ are both occupied by a robot. So $Q$ is a $(k, P)$ configuration.

(b) either $p_X$ or $p_Y$ is occupied by a robot. If $p_X$ (resp. $p_Y$) is free then the robot at $X$ (resp. $Y$) moves to $p_X$ (resp. $p_Y$) along the circle and following the clockwise direction.

(c) $p_X$ and $p_Y$ are both free. The robots at $X$ and $Y$ can both move to their respective positions ($p_X$ and $p_Y$) except if $p_X$ (resp. $p_Y$) is the alone position of $P$ on $C_X$ (resp. $C_Y$). In this case, to avoid the creation of an angle $> 180$, the robot at $Y$ (resp. $X$) is allowed to move only if $X$ (resp. $Y$) has reached $p_X$ (resp. $p_Y$).

2. $0 < k \leq \alpha$. In this case we do not need to maintain any property except that the robots do not leave the circle during their motions. The robots can directly apply $\phi_{deploy}({C^Q}_k, C_k)$ on the whole circle.

At the end of this step, $Q$ is a $(k, P)$ configuration.

$Q$ **is a** $(\alpha, P)$ **configuration.** In this case all the robots but $r_l$ are at the right positions. By construction of the current $Q$ the final position $v_0$ of $r_l$ is $c$ or the intersection of the half line $[c, r_l)$ with ${C^Q}_\alpha$. So $r_l$ moves to $v_0$ along the $[c, r_l)$. During this motion, the property "$Q$ is a $(\alpha, P)$-config" is invariant until $r_l$ reaches $v_0$, in that case $Q = P$ and the algorithm terminates.

**Algorithm** $\phi_{deploy}(C_Q, C_P)$**.** Given in input:

- $C_Q$ the circle or the half circle where the robots are located
- $C_P$ the positions which have to be occupied by a part of the robots on $C_Q$.

Let $O$ be the origin of $C_Q$ ($O$ is the intersection between the virtual circle described by $C_Q$ and the half line $[c, r_l)$ if $C_Q$ is a whole circle, $X$ (resp. $Y$) if $C_Q$ is the half circle $C_X$ (resp. $C_Y$). Let $p$ be the farest position of $C_P$ from $O$ according to the clockwise direction. We distinguish two cases:

- If some robots are after $p$, they all move but the last along $C_Q$ in the counter clockwise direction such that they finally are between $O$ and $p$. Then the last one moves along $C_Q$ to $p$.
- All the robots are before $p$, so the last one moves along $C_Q$ to $p$.

This process is repeated with all the robots but the last and the previous position of $C_P$. The process terminates when all the positions of $C_P$ are occupied.

### 3.3 $v_0$ is a critical point of $P$

In this case all the vertices of $P$ are on one circle only, so $\alpha = 0$. The algorithm above can be apply after $v_0$ has been remplaced in $P$ by the first position on $C_0$ which is not critical. Since the system contains at least 4 robots, Lemma 12 proves that such a position exists.

### 3.4 correctness

Until the system forms the given pattern each step of the algorithm keeps the property for a the configuration to be a *Leader* configuration then to be a *Leader-P* configuration after the first step and once $Q$ is a $(k, P)$-configuration then $Q_k - \{r_l\} = P_k - \{r_l\}$ for ever.

## 4 Equivalence with three robots

In this section, we show that the equivalence holds even if we have only three robots. Contrary to the general case $n \geq 4$, we do not consider the smallest enclosing circle because it is more difficult to maintain this latter invariant when we have only three robots. So, we are going to consider the three edges' triangle formed by the robots in the plane and the three edges' triangle formed by the points in the pattern to form. Note that, we assume that the robots (resp. points) form a triangle even if these three latter ones are collinear: However, we will obtain a triangle for which its edges are superimposed. For more convenience in the discussion, we will denote by $\mathcal{T}$ the triangle initially formed by the robots and $\mathcal{Z}$ the triangle formed by the three points in the pattern. If we assume that the leader election is possible in $\mathcal{T}$, we know that $\mathcal{T}$ is not an equilateral triangle otherwise the leader election would be impossible [12, 1]. So, in $\mathcal{T}$ there exists an edge which is the unique smallest one or the unique farthest one.

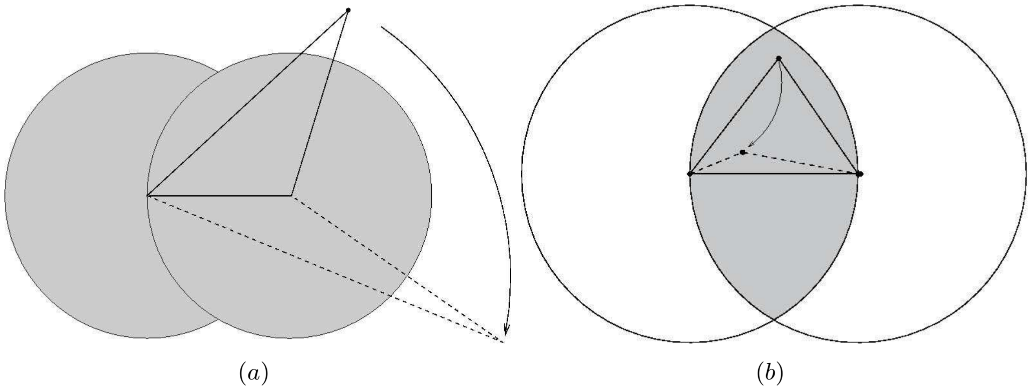


Figure 5: Two examples illustrating the strategy to form an arbitrary triangle.

**There exists an edge $e$ in $\mathcal{T}$ which is the unique smallest one.** We consider three cases depending on the shape of $\mathcal{Z}$

1. **There exists an edge in $\mathcal{Z}$ which is the unique smallest one**. In this case, each robot consider that the smallest edge $e$ in $\mathcal{T}$ corresponds to the smallest edge in $\mathcal{Z}$. So, it remains to construct both other edges by moving only one robot denoted $r$. Since the two remaining edges must to be more greater than the smallest edge, to form $\mathcal{Z}$ no need that $r$ passes inside the area formed by the two disks centered in one vertex of $e$ and whose the radius is equal to the lenght of $e$ (see Figure 5$a$). Thus, $e$ and $r$ remains the same ones during the motions of $r$.

2. **There does not exist an edge in $\mathcal{Z}$ which is the unique smallest one but there exists an edge in $\mathcal{Z}$ which is the unique greatest one**.

   - If there exists an edge $f$ in $\mathcal{T}$ which is the unique greatest one then each robot consider that the greatest edge $f$ in $\mathcal{T}$ corresponds to the greatest edge in $\mathcal{Z}$. So, it remains to construct both other edges by moving only one robot denoted $r$. Since the two remaining edges must to be more smaller than the greatest edge, to form $\mathcal{Z}$ no need that $r$ passes outside the area formed by the intersection formed by both disks centered in one vertex of $f$ and whose the radius is equal to the lenght of $f$ (see Figure 5$b$). Thus, $f$ and $r$ remains the same ones during the motions of $r$.
   - If there does not exists an edge $f$ in $\mathcal{T}$ which is the unique greatest one then the leader moves in order to form a scalene triangle. So, at the following time there exists an edge in $\mathcal{Z}$ which is the unique smallest one, and the robots are in the first case.

3. **There does not exist neither an edge in $\mathcal{Z}$ which is the unique smallest one nor an edge in $\mathcal{Z}$ which is the unique greatest one**. In this case $\mathcal{Z}$ is an equilateral triangle. The robots can apply the algorithm in [7] allowing the robots to form an equilateral triangle starting from an arbitrary initial configuration.

**There does not exist an edge $e$ in $\mathcal{T}$ which is the unique smallest one but there exists an edge $e$ in $\mathcal{T}$ which is the unique greatest one.** By symmetry with the above paragraph.

## 5 Conclusion

We studied the relationship between the arbitrary formation problem and the leader election problem among robots having the chirality in $SSM$. 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 [12], we deduce that arbitrary pattern formation problem and Leader election are equivalent, *i.e.,* it is possible to solve the pattern formation problem for $n \neq 2$ if and only if the leader election is solvable too.

In a future work, we would like to investigate the same equivalence (1) in the case without chirality and (2) in a weakest model, assuming more asynchrony, such that Corda.

## References


[1] D Canepa and M Gradinariu Potop-Butucaru. Flocking via leader election in robot networks. In *9th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2007)*, volume 4838 of *Lecture Notes in Computer Science, Springer*, pages 52–66, Paris, France, 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] R Cohen and D 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, O Labbani-Igbida, and F Petit. Circle formation of weak mobile robots. *ACM Transactions on Autonomous and Adaptive Systems*, 3(4), 2008.

[6] Y Dieudonné and F Petit. Deterministic leader election in anonymous sensor networks without common coodinated system. In *11th International Conference On Principles of Distributed Systems (OPODIS 2007)*, volume 4878 of *Lecture Notes in Computer Science, Springer*, pages 132–142, 2007.

[7] Yoann Dieudonné and Franck Petit. Squaring the circle with weak mobile robots. In *19th International Symposium On Algorithms and Computation (ISAAC 2008)*, volume 5369 of *Lecture Notes in Computer Science, Springer*, pages 354–365, Gold Coast, Australia, 2008.

[8] P Flocchini, G Prencipe, N Santoro, and P Widmayer. Distributed coordination of a set of autonomous mobile robots. In *IEEE Intelligent Vehicule 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 $r^3$ 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.