



HAL
open science

Probabilistic Asynchronous Arbitrary Pattern Formation

Quentin Bramas, Sébastien Tixeuil

► **To cite this version:**

Quentin Bramas, Sébastien Tixeuil. Probabilistic Asynchronous Arbitrary Pattern Formation. [Research Report] Université Pierre et Marie Curie. 2015. hal-01184532v2

HAL Id: hal-01184532

<https://hal.science/hal-01184532v2>

Submitted on 11 Mar 2016 (v2), last revised 20 Sep 2017 (v4)

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.

Probabilistic Asynchronous Arbitrary Pattern Formation*

Quentin Bramas¹

Sébastien Tixeuil¹

¹Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris.
{quentin.bramas, sebastien.tixeuil}@lip6.fr

Abstract

We propose a new probabilistic pattern formation algorithm for oblivious mobile robots that operates in the ASYNC model. Unlike previous work, our algorithm makes no assumptions about the local coordinate systems of robots (the robots do *not* share a common “North” nor a common “Right”), yet it preserves the ability from any initial configuration that contains at least 5 robots to form any general pattern (and not just patterns that satisfy symmetricity predicates). Our proposal also gets rid of the previous assumption (in the same model) that robots do not pause while moving (so, our robots really are fully asynchronous), and the amount of randomness is kept low – a single random bit per robot per Look-Compute-Move cycle is used. Our protocol consists in the combination of two phases, a probabilistic leader election phase, and a deterministic pattern formation one. As the deterministic phase does not use chirality, it may be of independent interest in the deterministic context. A noteworthy feature of our algorithm is the ability to form patterns with multiplicity points (except the gathering case due to impossibility results), a new feature in the context of pattern formation that we believe is an important asset of our approach.

1 Introduction

We consider a set of mobile robots that move freely in a continuous 2-dimensional Euclidian space. Each robot repeats a Look-Compute-Move (LCM) cycle [10]. First, it *Looks* at its surroundings to obtain a snapshot containing the locations of all robots as points in the plane, with respect to its ego-centered coordinate system. Based on this visual information, the robot *Computes* a destination and then *Moves* towards the destination. The robots are identical, anonymous and oblivious *i.e.*, the computed destination in each cycle depends only on the snapshot obtained in the current cycle (and not on the past history of execution). The snapshots obtained by the robots are not consistently oriented in any manner.

The literature defines three different models of execution: in the fully synchronous (FSYNC) model, robots execute LCM cycles in a lock-step manner, in the semi-synchronous (SSYNC) model, each LCM cycle is supposed atomic, and in the most general asynchronous (ASYNC) model, each phase of each LCM cycle may take an arbitrary amount of time. This last model enables the possibility that a robot observes another robot while the latter is moving (and moving robots appear in the snapshot *exactly the same way static robots do*), and that move actions are based on obsolete observations.

In this particularly weak model it is interesting to characterize which additional assumptions are needed for the robots to cooperatively perform a given task. In this paper, we consider the pattern formation problem in the most general ASYNC model. The robots start in an arbitrary initial configuration where no two robots occupy the same position, and are given the pattern to be formed as a set of coordinates in their own local

*This work was performed within the Labex SMART supported by French state funds managed by the ANR within the Investissements d’Avenir programme under reference ANR-11-IDEX-0004-02.

coordinate system. An algorithm solves the pattern formation problem if within finite time the robots form the input pattern and remain stationary thereafter.

Related Works. The pattern formation problem has been extensively studied in the deterministic setting [2, 1, 7, 8, 4, 11, 12, 5, 6, 10]. The seminal paper on mobile robots [10] presents a deterministic solution to construct general patterns in the SSYNC model, with the added assumption that robots have access to an infinite non-volatile memory (that is, robots are *not* oblivious). The construction was later refined for the ASYNC model by Bouzid *et al.* [2], still using a finite number of infinite precision variables.

The search for an oblivious solution to the general pattern formation proved difficult [6]. For oblivious deterministic robots to be able to construct any general pattern, it is required that they agree on a common “North” (that is, a common direction and orientation) but also on a common “Right” (that is, a common chirality), so that robots get to all agree on a common coordinate system. If only a “North” (and implicitly if only a “Right”) is available, then some patterns involving an even number of robots cannot be formed. Relaxing the common coordinate system condition led to a characterization of the patterns that can be formed by deterministic oblivious robots [7, 8, 12]. The best deterministic algorithm so far in the ASYNC model without a common coordinate system [12] proves the following: If ρ denotes the geometric symmetry of a robot configuration (i.e., the maximum integer ρ such that the rotation by $2\pi/\rho$ is invariant for the configuration), and I and P denote the initial and target configurations, respectively, then P can be formed if and only if $\rho(I)$ divides $\rho(P)$. All aforementioned deterministic solutions assume that both the input configurations and the target configuration do not have multiplicity points (that is, locations hosting more than one robot), and that robots share a common chirality. Overall, oblivious deterministic algorithms either need a common coordinate system or cannot form any general pattern.

To circumvent those impossibility results, the probabilistic path was taken by Yamauchi and Yamashita [13]. The robots are oblivious, operate in the most general ASYNC model, and can form any general pattern from any general initial configuration (with at least $n \geq 5$ robots), without assuming a common coordinate system. However, their approach [13] makes use of three hypotheses that are not proved to be necessary: (i) all robots share a common chirality, (ii) a robot may not make an arbitrary long pause while moving (more precisely, it cannot be observed twice at the same position by the same robot in two different Look-Compute-Move cycles while it is moving), and (iii) infinitely many random bits are required (a robot requests a point chosen uniformly at random in a continuous segment) anytime access to a random source is performed. While the latter two are of more theoretical interest, the first one is intriguing, as a common chirality was also used extensively in the deterministic case. The following natural open question arises: is a common chirality a necessary requirement for mobile robot *general* pattern formation? As the answer is yes in the deterministic [6] case, we concentrate on the probabilistic case.

Our contribution. In this paper, we propose a new probabilistic pattern formation algorithm for oblivious mobile robots that operate in the ASYNC model. Unlike previous work, our algorithm makes no assumptions about the local coordinate systems of robots (they do *not* share a common “North” nor a common “Right”), yet it preserves the ability from any initial configuration that contains at least 5 robots to form any general pattern (and not just patterns such that $\rho(I)$ divides $\rho(P)$). Besides relieving the chirality assumption, our proposal also gets rid of the previous assumption [13] that robots do not pause while moving (so, they really are fully asynchronous), and the amount of randomness is kept low – a single random bit per robot is used per use of the random source – (vs. infinitely many previously [13]). Our protocol consists in the combination of several phases, including a deterministic pattern formation one. As the deterministic phase does not use chirality, it may be of independent interest in the deterministic context.

A noteworthy property of our algorithm is that it permits to form patterns with multiplicity points (*without* assuming robots are endowed with multiplicity detection), a new feature in the context of pattern formation that we believe is an important asset of our approach. Of course, the case of gathering (a special pattern defined by a unique point of multiplicity n) remains impossible to solve in our settings [9].

2 Model

Robots operate in a 2-dimensional Euclidian space. Each robot has its own local coordinate system. For simplicity, we assume the existence (unknown from the robots) of a global coordinate system. Whenever it is clear from the context, we manipulate points in this global coordinate system, but each robot only sees the points in its local system. Two set of points A and B are *similar*, denoted $A \approx B$, if B can be obtained from A by translation, scaling, rotation, or symmetry. A *configuration* P is a set of positions of robots at a given time. Each robot that looks at this configuration may see different (but similar) set of points.

Each time a robot is activated it starts a Look/Compute/Move cycle. After the *look* phase, a robot obtains a configuration P representing the positions of the robots in its local coordinate system. After an arbitrary delay, the robot *computes* a path to a destination. Then, it *moves* toward the destination following the previously computed path. The duration of the move phase, and the delay between two phases, are chosen by an adversary and can be arbitrary long. The adversary decides when robots are activated assuming a *fair* scheduling *i.e.*, in any configuration, all robots are activated within finite time. The adversary also controls the robots movement along their target path and can stop a robot before reaching its destination, but not before traveling at least a distance $\delta > 0$ (δ being unknown to the robots).

An execution of an algorithm is an infinite sequence $P(0), P(1), \dots$ of configurations. An algorithm ψ forms a pattern F if, for any execution $P(0), P(1), \dots$, there exists a time t such that $P(t) \approx F$ and $P(t') = P(t)$ for all $t' \geq t$. In the sequel, the set of points F denotes the pattern to form. The coordinates of the points in F are given to the robots in an arbitrary coordinate system so that each robot may receive different, but equivalent, pattern F . If the pattern contains points of multiplicity, the robots receives a multiset, which is a set where each element is associated with its multiplicity. Even if the robots are not endowed with multiplicity detection, they know from the pattern what are the points of multiplicity to form. In particular, then can deduce from the pattern, the number n of robots, even if they do not see n robots.

3 Algorithm Overview

Our algorithm is divided into four phases. Since robots are oblivious and the scheduling is asynchronous, we cannot explicitly concatenate several phases to be executed in a specific order. However, one can simulate the effect of concatenation of two (or more) phases by inferring from the current configuration which phase to execute. Implementing this technique is feasible if phases are associated with disjoint sets of configurations where they are executed. Also, in order to simplify the proof of correctness, robots should not switch phases when placed in a configuration containing moving robots *i.e.*, a phase has to ensure that if the configuration resulting from a movement is associated with another phase, then all the robots are static (that is, none of them is moving). When this property holds, the first time a phase is executed, we can suppose that the configuration is static.

Our algorithm can form an arbitrary pattern. In particular, the pattern F can contain points of multiplicity. If this is the case, the robots create a new pattern \tilde{F} from F where they remove the multiplicity, and add around each point p of multiplicity m , $m - 1$ points really close to p , and located at the same distance to the center of the smallest circle enclosing F . The algorithm then proceed as usual with \tilde{F} instead of F . The initial pattern F is formed by the termination phase, when \tilde{F} is almost formed. So from now, we suppose that the pattern does not contains points of multiplicity and we refer to the details of the termination phase to see how an arbitrary pattern is formed.

In the following we define the phases of our algorithm and the set of associated configurations, starting from the more precise one (the phase we intuitively execute at the end to complete the pattern formation). Unless otherwise stated, the *center* of a configuration refers to the center of the smallest enclosing circle of this configuration.

Termination. The termination phase occurs when all robots, except the closest to the center, forms the target

pattern (from which we remove one of the point closest to the center). The phase consists in moving the last robot towards its destination. While moving, the robot remains the closest to the center, so that the resulting configuration is associated to the same phase.

Almost Pattern Formation. Among the remaining configurations, we associate the guided ones to this phase. A configuration is *guided* when a unique robot is sufficiently close to the center and induces by its position a global sense of direction and orientation to every robot. In particular, when executing this phase, robots are totally ordered and have a unique destination assigned. The phase consists first in moving all the robots (except the one that is closest to the center), one by one, so that they are at the same distance to the center as their destination in the pattern. Secondly, the robots moves toward their destination, keeping their distance to the center and the ordering unchanged. The configuration has to remain guided until each robot, except the closest to the center, reaches its destination. A configuration obtained after executing this phase is either associated to the same phase, or to the termination phase.

Formation of a Guided Configuration 1 (FBC1). Among the remaining configurations, we associate the ones that contain a *centered equiangular or biangular* (CEB) set to this phase. A CEB-set is a subset of robots that exists when the configuration is symmetric or has a non-trivial symmetricity. Moreover it is constructed independently from the coordinate system (so it is unique when it exists), and is invariant when the robots in this set move toward (or away from) the center of the configuration. When the configuration contains a CEB-set, our algorithm consists in moving the robots in this set to obtain a guided configuration. The invariance property of the CEB-set is important to ensure that resulting configurations are still associated with this phase.

In more details, when this phase is executed, the robots in the CEB-set Q moves either toward or away from the center with probability $1/2$. We show that, with probability 1, a unique robot is elected after a finite number of activations. Then, the elected robot performs a special move to force the other robots in Q to terminate their movement. Once each robot is static, the elected robot moves toward the center to create a guided configuration. During the execution of this phase, it is possible that the configuration is associated with the termination phase. If this happens, our algorithm makes sure that all robots are static.

Formation of a Guided Configuration 2 (FBC2). We associate all remaining configurations to this phase. When executing this phase, the configuration does not have a CEB-set. This implies that the configuration is not symmetric, so the robots are totally ordered. Therefore, the smallest robot moves toward the center to create a guided configuration (it remains the smallest robot while doing so). Before the movement, the robot checks if there exists a point in its path that creates a configuration containing a CEB-set. If it is the case, the robot chooses this point as its destination so that, when the configuration contains a CEB-set (and the robots switch to the FBC1 phase), all the robots are static.

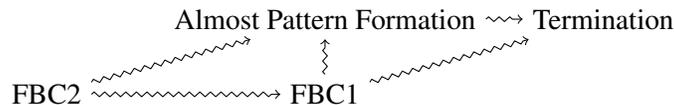


Figure 1: Relations between the phases of the algorithm.

We can define the relation \rightsquigarrow between phases, where $A \rightsquigarrow B$ if executing the phase A can lead to a configuration associated with phase B . Since the transitive closure of this relation on the set of phases that form our algorithm is a partial order (see Figure 1), then our algorithm terminates if and only if each phase terminates.

4 Algorithm Details

In this section we describe in more detail the phases of our algorithm. We start by listing the necessary notations used in the remaining of the section.

4.1 Notations

Let P be a set of points, then $C(P)$ denotes the smallest enclosing circle of P . Otherwise mentioned, $c(P)$ denotes the center of $C(P)$. The circle of a robot $r \in P$ is the circle centered at $c(P)$ containing r . We say a robot moves on its circle if its trajectory is contained in its circle. A radial movement is a linear movement whose origin and destination are on the same half-line of origin $c(P)$. We say a robot moves radially if it performs a radial movement.

The pattern to form, F , is given to each robot as a set of points in their local coordinate system. However, at each activation, robots can translate and scale their local coordinate system so that $C(P) = C(F)$. Hence, we suppose in the remainder of the paper that $C(P) = C(F)$, and that the radius of $C(P)$ is the common unit distance (unless otherwise mentioned). This is possible because in our case, the configuration where all robots share the same location (that is, are gathered) is not reachable. For two points a and b , $|a|_b = |a - b|$ denotes the distance between a and b . In a n -robot configuration P , as we are often interested in the distance between a point and the center $c(P)$, we simply write $|a|$ instead of $|a|_{c(P)}$.

The interior, resp. the exterior, of a disc or a circle C , denoted $interior(C)$, resp. $exterior(C)$, does not include the circumference. A set of points A (or simply a point) *holds* $C(P)$ if $C(P \setminus B) \neq C(P)$, for a subset $B \subset A$. The angle formed by three points u, v and w is denoted by $ang(u, v, w) \in [0, 2\pi)$, and the orientation depends on the context. If the orientation is not given, it is either clockwise or counterclockwise, but it does not vary for a given robot during a cycle.

Partial Ordering and Symmetricity. Given a set of points (typically a configuration) P , we order the points based on their coordinates coordinate systems defined by the points that are the closest to, but not at, the center $c(P)$. Formally, let $M = \{r \in P \text{ s.t. } |r| = \min_{r' \in P \setminus \{c(P)\}} |r'|\}$. For each robot r_m in M and each orientation o in $\{\odot, \ominus\}$ we define the polar coordinate system $Z_{r_m}^o : P \rightarrow \mathbb{R}^+ \times [0, 2\pi)$, $r \mapsto (|r|, ang(r_m, c(P), r))$ oriented by o . We denote by $Z_{r_m}^o(P)$ the increasing sequence of coordinates in $Z_{r_m}^o$ of the points in P . In particular, if r_1 and r_2 are two points such that $|r_1| < |r_2|$, then the coordinates in $Z_{r_m}^o(P)$ of r_1 are smaller than the coordinates of r_2 .

We define Z_{\min} as the set of coordinate systems that minimize the sequence of coordinates, using the lexicographical order. We use this set of coordinate systems to define the relation $<$, where $r < r'$ if and only if the coordinates of r are smaller than the coordinates of r' in every coordinate systems $Z \in Z_{\min}$. From this relation we deduce the *partial ordering* of P .

We define the *symmetricity* of a configuration P as the number of minimal points in its partial ordering. In particular, if the robots are endowed with chirality (or if the configuration does not have an axis of symmetry), this definition matches the definition of symmetricity of previous work [1, 8]. In particular, if a configuration P is such that $\rho(P) = k > 1$, then P can be partitioned in n/k regular k -gons centered at $c(P)$ (where a 2-gon is a line with center its middle). However, this is not true in the general case when robots may not have a common sense of chirality. Also, it is important to notice that if $c(P) \in P$ then $\rho(P) = 1$, even if the configuration is symmetric or is invariant by rotation.

Ordered and Guided Configuration. An *ordered* configuration is a configuration where the partial order of robots is a *total* order. In particular, this implies that there is a unique coordinate system Z in Z_{\min} and all the robots agree on Z as a global coordinate system.

When the configuration is (totally) ordered, let f_1, f_2, \dots, f_n be any total ordering of points in F satisfying the partial ordering of F (an arbitrary ordering can be chosen if more than one satisfies the condition). Even if the pattern is given to robots using an arbitrary coordinate system, each robot can scale it so that $C(F) = C(P)$, mirrors it so that the orientation chosen for the ordering of the points in F coincides with the orientation of the ordering of the robots, and rotate it so that the points f_2 and r_2 are on the same half-line of origin $c(P)$. Without loss of generality, we can suppose that F is given to the robots with those property, in the global coordinate system Z .

One can observe that the choice of the ordering of points in F is not important, since the resulting coordinates of points in F in the global coordinate system Z are identical for two different orderings (indeed, the resulting sets are equivalent, and are equals after applying the aforementioned transformations). So, from now

on, when the configuration is ordered, robots see the points in F in the same way in the global coordinate system, and have a common ordering of points in F .

A *guided configuration* is an ordered configuration that satisfies: (i) $|r_1| \leq |r_2|/2$; (ii) $|r_2| \leq |f_2|$; (iii) $2\text{ang}(r_1, c(P), r_2) < \min_{|f|=|f_2|} \text{ang}(f_2, c(F), f)$ (see Figure 2a).

4.2 Termination

We first explain how patterns with points of multiplicity are handled. Let F be the initial pattern that can contain points of multiplicity. Let \tilde{F} be the set of points constructed from F by removing the multiplicity and adding, for each point $p \in F$ of multiplicity $m > 1$, $m - 1$ points p_1, \dots, p_{m-1} such that $|p_i| = |p|$, $\text{ang}(p, c(F), p_i) < \pi$, and $|p_i - p| = \frac{d}{4i}$, with $d = \min_{f, f' \in F} |f - f'|$. The orientation of the angle is either deduced from the pattern, or arbitrary if F has an axis of symmetry. For all algorithm phases except Termination, \tilde{F} is used instead of F whenever F contains points of multiplicity.

To execute the Termination phase, the configuration must be totally ordered (using the set of points, excluding multiplicity information), and at least one robot must be located at each point of F (except maybe the smallest one). Also, if there are no points of multiplicity, every robot must be at distance at most $d/4$ from its destination in \tilde{F} . Since the positions of the robots are totally ordered, even when there exists a point of multiplicity, the robots can see the pattern F to form in a common global coordinate system (as well as \tilde{F}). However, when there are points of multiplicity, the robots cannot deduce their destination in F (or \tilde{F}) from the ordering of their position.

If there exists a robot $r \neq r_1$ not located at a point in F , then r chooses the closest point in F as its destination and moves toward it while remaining in its circle. The global coordinate system remains unchanged because r_1 does not move. Eventually, r_1 becomes the only robot not located at its destination. When r_1 remains the only robot that is not located on its destination in F , then it moves toward its destination and the other robots do not move. The global coordinate system is modified but r_1 remains the closest robots to the center (except maybe when it reaches its destination). When r_1 reaches its destination (which can be a points already occupied by another robot), the pattern F is formed.

4.3 Almost Pattern Formation

In this phase we assume that the configuration is guided. We have a total order over robots r_1, \dots, r_n , and each robot sees the pattern in the same way in a global coordinate system Z . In Z , the points of F are also totally ordered f_1, \dots, f_n so that each robot r_i knows its final destination f_i . The goal of this phase is that each robot, except r_1 , reaches its destination. In the sequel $P' = P \setminus \{r_1\}$ and $F' = F \setminus \{f_1\}$. One can see that $C(P') = C(P)$ because r_1 the configuration is guided, and we can assume that $C(F') = C(F)$ (if this is not the case, we can modify the ordering of points in F so that f_1 does not hold $C(F)$).

Reaching its Circle. This sub-phase consists in moving robots so that there is the right number of robots on each circle centered at $c(P)$. Let C_1, C_2, \dots, C_m be the m circles centered at $c(P)$ with decreasing radius, each containing at least one point in F' . For each $1 \leq i \leq m$, let $m_i = |C_i \cap F'| > 0$. We have $\sum_{i=1}^m m_i = |F'| = n - 1$.

Before beginning this sub-phase, the robots that have a null angle (except r_2) move on their circle following the direct orientation while preserving the order (*i.e.*, without reaching another robot), so that no robot has a null angle (except r_2). This is required for proper operation of action *ii*), defined below. Also, if $m_1 = 2$, since two robots cannot move on $C(P)$ synchronously to keep $C(P)$ unchanged, we need to execute a special procedure to ensure that the two robots are located at the two points of $C(P) \cap F'$, keeping $C(P)$ unchanged. Informally, this procedure moves another robot on $C(P)$ if there are only two robots on it, then the two greatest robots reach their destination point in $C(P) \cap F'$. Then, the other robots can leave safely $C(P)$. From now on, we suppose that if $m_1 = 2$, then C_1 already contains two robots located at their corresponding point in F .

Recursively, we move robots such that each circle C_i contains exactly m_i robots. We define the following procedure for a given i , $1 \leq i \leq m$. The procedure executes three actions sequentially and assumes, if $i > 1$, that

$$|\text{interior}(C_{i-1}) \cap P'| = \sum_{j=i}^m m_j.$$

- i) **cleanExterior(i):** If $i > 1$ and $|\text{interior}(C_{i-1}) \cap \text{exterior}(C_i) \cap P'| > 0$, then the smallest robot in $\text{exterior}(C_i)$ moves to C_i while it remains greater than robots already in C_i . To do so, it can move a little toward $c(P)$, so that there is no other robot in its circle, then moves on its circle so that its angle is greater than the angles of robots in C_i , and finally moves radially toward $c(P)$ to reach C_i (e.g., movement of robot r_5 in Figure 2b). If $i = m$, we also ensure that its angle is less than $2\pi - \text{ang}(r_1, c(P), r_2)$, in order not to break the guided configuration. We repeat this procedure until there are no more robots between C_{i-1} and C_i (e.g., movement of robot r_6 in Figure 2b).
- ii) **locateEnoughRobots(i):** If $|C_i \cap P'| < m_i$, then we have $|\text{interior}(C_i) \cap P'| \geq 1$. Indeed, if $i = 1$, $|\text{interior}(C_1) \cap P'| = m_1 - |C_1 \cap P'| \geq 1$, otherwise, there are by hypothesis at least m_i robots inside C_{i-1} , and after performing action i), these robots are not between C_i and C_{i-1} . The greatest robot in $\text{interior}(C_i)$ now moves to C_i while remaining smaller than robots already in C_i . To do so, it can move a little away from $c(P)$ so that no other robot remains in its circle, then move on its circle so that its angle is smaller than the angles of robots in C_i (but not null), and finally move radially away from $c(P)$ to reach C_i (e.g., movement of robot r_7 in Figure 2b). We can repeat this action until there are exactly m_i robots on C_i .
- iii) **removeRobotsInExcess(i)** If $i > 1$ and $|C_i \cap P'| > m_i$, then the smallest robot in C_i moves a little toward $c(P)$ (here, “a little” means a small distance such that the order is preserved, i.e., the robot does not reach the circle of another robot nor C_{i+1}). We repeat this process until there are exactly m_i robots on C_i . If $i = 1$ and $|C_1 \cap P'| > m_1$, then we cannot do the exact same thing because we have to ensure that $C(P)$ does not change. However, we know that $m_1 \geq 3$. The m_1 greatest robots r_n, \dots, r_{n-m_1} remain on C_1 , and have to be the only robots to hold $C(P)$. To do so, the angles formed by two consecutive robots in $\{r_n, \dots, r_{n-m_1}\}$ have to be smaller than, or equal to π . This is obtained by moving the robots on C_1 , while preserving the ordering and $C(P)$, such that r_n, \dots, r_{n-m_1} form the regular m_1 -gon that has the line $c(P)r_2$ as an axis of symmetry (see Figure 2d). At the same time, if the m_1 -gon is not formed yet, other robots in C_1 move on C_1 to be evenly distributed in the arc between angle 0 and π/m_1 (see the blue arc in Figure 2d), again while preserving the ordering and $C(P)$. Overall, each robot on C_1 has a deterministic (and non-blocking) destination. Once the m_1 -gon is formed (even if some other robots are still moving), the smallest robot in $C(P) \cap P$ moves a little toward $c(P)$ (see Figure 2e). This is repeated until only r_n, \dots, r_{n-m_1} remain on C_1 .

After executing the above procedure for a given i , we have m_i robots on C_i and $|\text{interior}(C_i) \cap P'| = |\text{interior}(C_{i-1}) \cap P'| - m_i = \sum_{j=i+1}^m m_j$, so that we can execute the same procedure with $i + 1$. If $i < m$ it is important to observe that some robots (those ordered to move in the last two cases) may still be in movement, but since they are now strictly between C_i and C_{i+1} , they receive a new order with deterministic destination when executing the procedure with $i + 1$. Hence, at the end of the procedure with $i = m$, all robots are static.

After executing those actions for $i = 1, 2, \dots, m$, each circle contains the proper number of robots.

Reaching its Destination. Let $i \in [1, m]$, C_i now contains m_i robots and the m_i destinations for those robots. The robots and the destinations are ordered so that each robot is aware of its corresponding destination (see Figure 2c). They can all move toward their destination, while remaining on C_i and preserving the robots ordering (i.e., without reaching another robot position). When a robot r is active and another robot is on the way, r chooses on the circle half the distance to this robot. There cannot be a deadlock since there is no cycle in the waiting relation. Indeed, robots on C_i are ordered by angle so that they behave like they are on a finite segment of length 2π . If $i = 1$, during their movement, robots also ensure that $C(P)$ remains unchanged. To do so, if a robot $r \in C_1$ is active and detects that its movement can modify $C(P)$, then it moves as much as possible without changing $C(P)$.

Lemma 1. *During Phase 4.3, each robot in P' reaches its destination in F' , and $C(P)$ remains unchanged.*

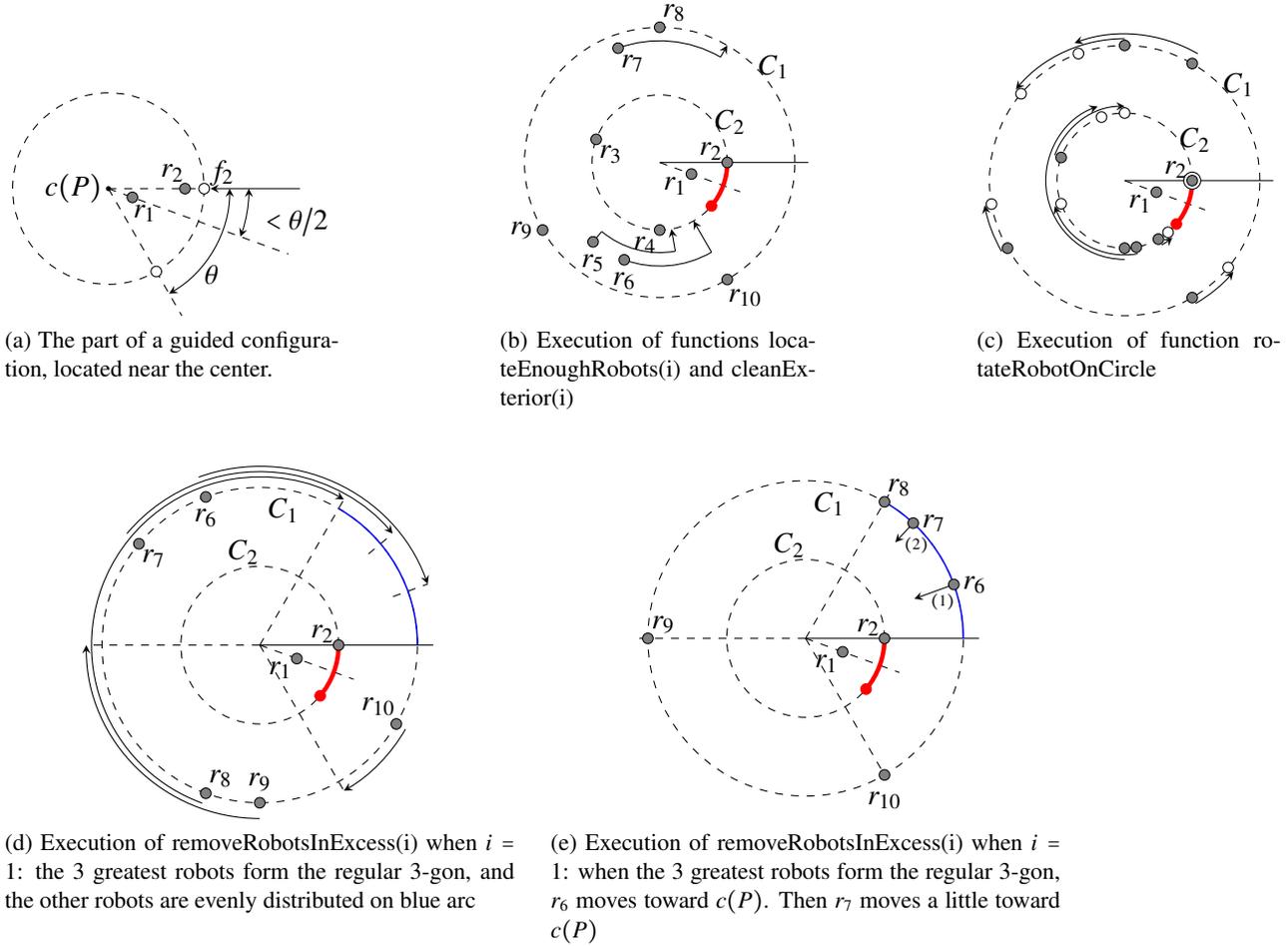


Figure 2: Illustration of the pattern formation algorithm

Proof. First we show that there is no deadlock. Suppose we have on a circle C , m robots $r_1 < r_2 < \dots < r_m$ and m destinations $d_1 < d_2 < \dots < d_m$. r_i has destination d_i and moves toward it (staying on C) in the direct orientation if $r_i < d_i$ and in the indirect orientation otherwise. For the sake of contradiction, suppose that $r_i < d_i$ and r_i cannot reach d_i , even after an infinite number of activation. we observe that, to block r_i , r_{i+1} must satisfy $r_i < r_{i+1} \leq d_1$ and r_{i+1} is not able to free the way for r_{i+1} . This implies that $r_{i+1} \leq r_{i+3} \leq d_1$. Recursively, this means that $r_m \leq d_1$. But nothing blocks r_m to reach d_m when $r_m \leq d_m$, a contradiction.

Now suppose for the purpose of contradiction that $C(P)$ is modified. This means that there exist two robots r and r' on C_1 that form an angle greater than π . Before $C(P)$ is modified, they form an angle of at most π , so that one robot's movement on C_1 in the direct orientation, and the other's movement on C_1 in the indirect orientation. This is possible only if there is no point in F on C_1 between r and r' , which is a contradiction with the fact that $C(P) = C(F)$. \square

At the end of Phase 4.3, robots in P' form F' and the configuration becomes associated with Phase 4.2.

4.4 Formation of a Guided Configuration 1

If the current configuration P is not guided, and contains a CEB-set Q , we execute this phase to obtain a guided configuration. To do so, one robot has to be elected to guide the configuration. Once a unique robot is elected, other robots may still be moving. One way to be sure that the other robots are static is to give them new

destinations. So, the elected robot moves on its circle with a small angle. After this move, the other robots can still detect the elected robot. Moreover, the angle of the elected robot can be computed by all robots, in the same way. Then, this angle can be used as a persistent memory. In more details, the robots can agree to perform some action depending on that angle. Particularly, just after the robot election, when the angle is smaller than a given ε , all the robots are ordered to move at the same distance from the center as the elected robot. After that, we are sure that all the robots are static, and the elected robot moves again on its circle to have an angle of 2ε from its original position, in the CEB-set. While this is done, all the other robots remain still, and the elected robot can then move toward the center of the configuration to create a guided configuration. We now give the formal definition of a CEB-set, how to compute it, and we detail the main parts of this phase, which are the robot election and how we use this robot to form a guided configuration.

CEB-set Definition. For a given point c , the *string of angles* $SA_{c,r,o}(P)$ starting from a robot r with orientation o is the sequence of angles formed by the robots in P with r , around Point c , and with Orientation o . If for two robots $r \neq r'$, the strings of angles $SA_{c,r,o}(P)$ and $SA_{c,r',o}(P)$ are equal, we say the configuration is *regular* (this is tantamount to say the string of angles is periodic). Regular sets have been introduced by Bouzid et al. [1] for solving the gathering problem. Its main properties is that there exists at most one point c such that $SA_{c,r,o}(P) = SA_{c,r',o}(P)$, and this point is invariant by robots movement toward to, or away from c (as it is also the Weber point). Particular regular configurations are equiangular configurations (the period of the string of angles is 1) and biangular configurations (the period of the string of angles is 2). The point c is called the center of regularity of the configuration.

We say that a configuration is *sym-regular*, if two string of angles, centered at the center $c(P)$ of $C(P)$, with opposite orientations, are equal, *i.e.*, if there exists $r, r' \in P$ (possibly $r = r'$) such that $SA_{c(P),r,\odot}(P) = SA_{c(P),r',\ominus}(P)$. We define the *centered equiangular of biangular set* (CEB) set $CEB(P)$ of a configuration P as follow:

- If P is not regular nor sym-regular: then P does not have a CEB-set.
- If the whole configuration P is equiangular or biangular: then $CEB(P) = P$ and, in this case, the center of the configuration $c(P)$ is the center of regularity.
- Otherwise: the CEB-set is the maximum subset of robots that is equiangular or biangular with center $c(P)$. The robots in the CEB-set must be closer to the center than the other robots. Also, in this case, the CEB-set must not include robots that can change the smallest enclosing circle of P . The construction of the CEB-set Q of a regular or sym-regular configuration P is presented as Algorithm **constructCEBSet**.

Algorithm constructCEBSet:

```

 $Q = \emptyset$   Ignore  $\leftarrow \emptyset$ 
while  $P \setminus (Ignore \cup Q) \neq \emptyset$  do
    Let  $S$  be the set of smallest robots in  $P \setminus (Ignore \cup Q)$  according to the partial ordering of robot
    if  $Q \cup S$  holds  $P$  then Ignore  $\leftarrow Ignore \cup S$ 
    else if  $Q \cup S$  is equiangular or biangular then  $Q \leftarrow Q \cup S$ 
    else stop

```

Theorem 1. For a configuration P , if $\rho(P) > 1$, then P has a unique, non-empty, CEB-set.

If by moving one robot, a configuration has a CEB-set and the robot is among the closest robots to the center of the CEB-set, we say that the configuration has a CEB-set with a *shifted robot*. The shifted robot is the robot that we need to move to create the CEB-set. The configuration is seen as if the shifted robot is at its right position (*i.e.*, the position where it has to be for the configuration to contain a CEB-set). The difference between the angles of this robot before and after the move (with origin $c(P)$) is called the *shift angle*. We have the following theorem:

Theorem 2. *Let $n \geq 5$, and P be a n -robot configuration that contains a CEB-set with a shifted robot. Let $\theta(P)$ be the smallest angle, centered at $c(P)$, between two robots in the configuration. If the shift angle is at most $\theta(P)/2$, then the shifted robot is unique.*

The proof of this result is not trivial, and can be found in the appendix, due to lack of space. When computing the CEB-set of a configuration, all the robots can see if the CEB-set has a shifted robot (*i.e.*, is not exactly equiangular or biangular, due to the misposition of one robot). The existence of a shifted robot is crucial for our algorithm because the shift angle can be used as a persistent memory. In more details, if all the other robots move radially, they do not change the shift angle, so that the shifted robot can modify the shift angle to remember some information for the next activation. Moreover, all the other robots can see the shift angle and deduce some information from it. In our algorithm the shift angle is used to order the robots to make a specific move, and then, another shift angle is used to stop them.

Robot Election. Let Q be the CEB-set. We say a robot r_e is *elected* if it is the shifted robot of Q or if $|r_e| < \frac{7}{8} \min_{r \in Q \setminus \{r_e\}} |r|$. A robot is aware that it is elected if it is elected during its look phase. When a robot is aware it is elected, and is not yet the shifted robot, it moves on its circle to create a shift angle of $\theta/8$. To elect a robot, each robot r in Q proceeds in the following way. If there is another robot in Q that is strictly closer to the center, then r does not move. If r is not elected and is one of the closest robot (unique or not), then r chooses randomly (each choice with probability $1/2$) to go toward or away from the center $c(P)$. If r chooses to move toward the center, it moves a distance $|r|/8$. If r chooses to move away to the center, it moves a (possible null) distance $\min(\frac{1}{2}(d - |r|), \frac{1}{7}|r|)$, where d is the minimum distance to the center among robots in $P \setminus Q$ (and $d = \infty$ if $P \setminus Q = \emptyset$). This ensures that robots in Q remain in the CEB-set in the resulting configuration.

Also, our algorithm ensure that $n - 1$ robots cannot form part of the pattern. To do so, a procedure is called that makes sure that, if a point in F is in the path of a robot, then this point is either chosen as its destination, or avoided (the procedure is described in the appendix). So that, if the configuration is associated with another phase, all the robots are static.

Lemma 2. *The following properties hold: (i) eventually one robot is aware it is elected with probability one, and (ii) once a robot is aware it is elected, another robot cannot be elected.*

Proof. *i)* Suppose first that robots always reach their destination. Initially the configuration is in a state where there are at least two robots whose location (or destination for those that are moving) are the closest to the center. Let r be the first (or one of the first) activated robot among them. With probability $1/2$, it chooses to move toward the center. Let r' be another robot among them. If r' is activated after r begins its movement, r' does not move, otherwise it moves away from the center with probability $1/2$. So with probability greater than $1/2^n$, r is the only robot to move toward the center. After that, if another robot is activated before the next activation of r , it does not move. At the next activation of r , with probability $1/2$, r chooses to move toward the center and becomes elected (and r is aware it is elected when it is next activated since the other robots are static). If this does not happen, *i.e.*, if the first or the second choice of r is to move away the center or if another robot chose to move toward the center, then the configuration gets back to its initial state. So, we have infinitely often a probability greater than $1/2^{n+1}$ to have an elected robot. So, eventually there is a single elected robot with probability one. Now, If robots do not always reach their destinations, the probability that r moves by a distance d toward the center (to become elected), is $1/2^{\lceil \frac{d}{\delta} \rceil}$, instead of $1/4$ in the first case. Indeed, in the first case r needs to chose two times to move toward the center, and here r needs to choose $\lceil \frac{d}{\delta} \rceil$ times to move toward the center. The probability that one of the other robot chooses to move away (and stay still while it is not one of the closest robot to the center) is still $1/2$. So that there is again a non null probability that a robot is elected, which implies that a robot is eventually elected with probability one.

ii) Once a robot r_e is elected, the other robots are currently either moving away, not moving, or moving toward the center by a distance at most the eighth of their distance to the center. In each case, when another robot looks again (after it finishes its movement) it sees that the robot r_e is the only closest robot to the center, and then it chooses not to move. Indeed, if another robot r is moving toward the center, it is by a distance at

most $|r|/8$. Since we have $|r_e| < \frac{7}{8}|r|$ when r started its movement, we have $|r_e| < |r|$ after r finishes it. After r 's movement, r_e is maybe no longer elected, but since r_e was aware it was elected, it already chooses to move on its circle to create a $1/8$ -shifted-regular set (and it moves by a non null distance, so that in the next look phase, it is shifted). \square

Using the Shifted Robot to Form a Guided Configuration. If the elected robot is not shifted or if the shift angle is in $[0, \theta/8)$, it moves on its circle to create a shift $\theta/8$. Also, if Q is shifted with a shift in $(\theta/8, \theta/4)$ such that the other robots in Q are not on the same circle as the shifted robot, then the shifted robot moves on its circle to create a shift $\theta/8$. If another robot is activated during the movement of the shifted robot (*i.e.*, when the shift is not exactly $\theta/8$), it chooses not to move. When the shift angle is exactly $\theta/8$, the shift angle waits for the other robots in Q to reach its circle. When this is the case, the shifted robot moves on its circle to create a shift angle of $\theta/4$. Then it moves toward the center. This creates an ordered configuration. The shifted robots $r = r_1$ move radially such that $|r| = \min(|r_2|/2, |f_2|/2)$. Then we have to move r_2 such that $|r_2| \leq |f_2|$. Since the configuration is not symmetric (because there is a shifted robot), we can choose θ small enough to ensure that the third condition to have a guided configuration is satisfied.

4.5 Formation of a Guided Configuration 2

If P does not have a CEB-set, then, by Theorem 1, $\rho(P) = 1$ and P has a unique smallest robot r_1 . Robots check if there is a position in the segment $[r_1, c(P))$ such that the whole configuration is equiangular or biangular (possibly with a shifted robot). If it is the case, the first such point in the path becomes the destination of r_1 . Otherwise, r_1 is ordered to move toward the center until it the configuration becomes guided. The resulting configurations cannot have a CEB-set (except if it is the whole configuration) since P does not contain one and r_1 performs a radial movement. Starting from a static initial configuration, once the configuration is guided, or once $CEB(P) = P$, all robots are static.

5 Concluding Remarks

Similarly to previous work [13], the initial configuration should not contain multiplicity positions. In the case where the initial configuration contains points of multiplicity, a convenient solution would be to reuse known pattern formation algorithms (such as ours) and run a preliminary phase where multiplicity points are eliminated. This task is known as the scattering task in the literature [3]. However, even the most recent developments [3] only considers the SSYNC model. Of course, as our protocol also performs correctly in SSYNC, it is possible to combine the two to obtain a protocol in SSYNC that manages multiplicities both in I and in F . Indeed, combining protocols in SSYNC is facilitated because moves are always aware of the latest configuration, so for all configurations that have multiplicities and do not belong to a legitimate path toward the target pattern, the scattering phase is run, until robots either reach a configuration where there is no point of multiplicity or a configuration that makes progress toward the target pattern. Extending this scheme to the ASYNC model requires to solve the open problem of ASYNC scattering, and making sure the combinations of protocols is feasible.

References

- [1] Zohir Bouzid, Shlomi Dolev, Maria Potop-Butucaru, and Sébastien Tixeuil. Robocast: Asynchronous communication in robot networks. In Chenyang Lu, Toshimitsu Masuzawa, and Mohamed Mosbah, editors, *Principles of Distributed Systems - 14th International Conference, OPODIS 2010, Tozeur, Tunisia, December 14-17, 2010. Proceedings*, volume 6490 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2010.

- [2] Zohir Bouzid and Anissa Lamani. Robot networks with homonyms: The case of patterns formation. In Xavier Défago, Franck Petit, and Vincent Villain, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 6976 of *Lecture Notes in Computer Science*, pages 92–107. Springer Berlin Heidelberg, 2011.
- [3] Quentin Bramas and Sébastien Tixeuil. The random bit complexity of mobile robots cattering. In *AD HOC NOW*, Lecture Notes in Computer Science, page to appear, Athens, Greece, July 2015. Springer Berlin Heidelberg.
- [4] Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. On the computational power of oblivious robots: Forming a series of geometric patterns. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '10, pages 267–276, New York, NY, USA, 2010. ACM.
- [5] Yoann Dieudonné, Franck Petit, and Vincent Villain. Leader election problem versus pattern formation problem. In Nancy A. Lynch and Alexander A. Shvartsman, editors, *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings*, volume 6343 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2010.
- [6] 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.
- [7] Nao Fujinaga, Hirotaka Ono, Shuji Kijima, and Masafumi Yamashita. Pattern formation through optimum matching by oblivious corda robots. In Chenyang Lu, Toshimitsu Masuzawa, and Mohamed Mosbah, editors, *Principles of Distributed Systems*, volume 6490 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2010.
- [8] Nao Fujinaga, Yukiko Yamauchi, Shuji Kijima, and Masafumi Yamashita. Asynchronous pattern formation by anonymous oblivious mobile robots. In MarcosK. Aguilera, editor, *Distributed Computing*, volume 7611 of *Lecture Notes in Computer Science*, pages 312–325. Springer Berlin Heidelberg, 2012.
- [9] Giuseppe Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *Theoretical Computer Science*, 384(2):222–231, 2007.
- [10] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. Comput.*, 28(4):1347–1363, 1999.
- [11] Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26-28):2433–2453, 2010.
- [12] Yukiko Yamauchi and Masafumi Yamashita. Pattern formation by mobile robots with limited visibility. In Thomas Moscibroda and Adele A. Rescigno, editors, *Structural Information and Communication Complexity - 20th International Colloquium, SIROCCO 2013, Ischia, Italy, July 1-3, 2013, Revised Selected Papers*, volume 8179 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2013.
- [13] Yukiko Yamauchi and Masafumi Yamashita. Randomized pattern formation algorithm for asynchronous oblivious mobile robots. In Fabian Kuhn, editor, *Distributed Computing*, volume 8784 of *Lecture Notes in Computer Science*, pages 137–151. Springer Berlin Heidelberg, 2014.

A Pseudo-code

The algorithm consists of several phases (that are not exactly divided in the same way as in the paper), described in pseudo-code in algorithm **formPattern**. Each procedure call has a phase condition. A phase is executed if and only if its phase condition is not verified. If the condition is verified, the next phase is considered. Each time a robot is activated, it must find the first phase with a condition that is not verified and follow the corresponding instructions. Each phase is done not to break the previous phase conditions. The condition line 3 is checked before because the movement line 4 breaks the condition of the other phases. Line 6 corresponds to the robot election. The goal is to select a robot by performing random radial movement in the CEB-set. Lines 7 to 16 correspond to the deterministic pattern formation algorithm. For simplicity, our pseudo-code does not handle the case of a pattern that contains points of multiplicity.

Algorithm **formPattern**: main algorithm that forms a pattern F

```

1  $ClosestF \leftarrow \{\text{smallest elements in } F \text{ that does not hold } C(F)\}$ 
2  $ClosestP \leftarrow \{\text{smallest elements in } P\}$ 
3 if  $ClosestP = \{r\}$  and  $\exists f \in ClosestF$  s.t.  $P \setminus \{r\} \approx F - \{f\}$  then
4    $r$  moves toward  $f$ 
5 else
6    $r_1 \leftarrow selectARobot()$ 
7    $P' \leftarrow P \setminus \{r_1\}$ 
8    $F' \leftarrow F \setminus \{f_1\}$  (with  $f_1 \in ClosestF$ )
9   Let  $C_1, C_2, \dots, C_m$  be the  $m$  circles centered at  $c(P)$  with decreasing radius, each containing at least one point in  $F'$ . For each  $1 \leq i \leq m$ ,
   let  $m_i = |C_i \cap F'| > 0$ 
10  if  $|C_i \cap F'| = 2$  then
11     $fixEnclosingCircle()$ 
12  for  $i = 1, 2, \dots, m$  do
13     $cleanExterior(i)$ 
14     $locateEnoughRobots(i)$ 
15     $removeRobotsInExcess(i)$ 
16   $rotateRobotOnCircle()$ 

```

Algorithm **handlePartiallyFormedPattern**: executed before the robot election to handle configuration that can create a configuration that verifies in line 3 of the main algorithm

```

if  $\{F_r \cap F_r^c\}$  is a partition of  $F$  such that  $F_r^c \approx P \setminus Q$  and
 $|Q| = 1$  robots in  $Q$  are located on an halfline  $[c(P), f)$ , with  $f \in F_r$  then
   $d_1 \leftarrow \text{radius of } C(F_r)$ 
   $d_2 \leftarrow \min\{d \mid \mathcal{D}(d_1) \cap \text{exterior}(\mathcal{D}(d)) \cap F_r = \emptyset\}$ 
   $d \leftarrow (d_1 + d_2)/2$ 
  if  $\exists r \in Q$  s.t.  $|r| > d$  then
    if  $\exists r \in Q$  s.t.  $|r| > d_1$  then
      for  $r \in Q$  s.t.  $|r| > d_1$  do
         $r$  moves radially at distance  $|d_1|$  from  $c(P)$ 
    else
      for  $r \in Q$  s.t.  $|r| > d$  do
         $r$  moves radially at distance  $|d|$  from  $c(P)$ 
  exit

```

B Robot Election Pre-phase

This pre-phase is executed before the robot election, *i.e.*, when the current configuration P contains a CEB-set Q of cardinal m . Before executing the robot election algorithm, a robot checks if the current configuration satisfies the following conditions:

Algorithm selectARobot: select a robot

Phase Condition: There exists a selected robot r_s
Returned Value: r_s

```
if  $P$  contains a CEB-set  $Q$  with shifted robot then
   $r_e \leftarrow$  the shifted robot
   $\varepsilon \leftarrow$  the shift angle
   $S \leftarrow \{r \in P \mid |r| > |r_e|\}$ 
  if  $S \neq \emptyset$  and  $\varepsilon \neq \theta/8$  then
     $r_e$  moves on its circle to create a  $\theta/8$ -shifted angle
  else if  $S \neq \emptyset$  and  $\varepsilon = \theta/8$  then
    for  $r \in S$  do
       $r$  moves radially at distance  $|r|$  from  $c(P)$ 
  else if  $\varepsilon < \theta/4$  then
     $r_e$  moves on its circle to create a  $\theta/4$ -shifted angle
  else
     $r_e$  moves radially toward  $c(P)$  to become selected
else if  $P$  contains a CEB-set  $Q$  then
  if  $P \setminus Q \neq \emptyset$  then
     $d \leftarrow \min_{r' \in P \setminus Q} |r'|$ 
  else
     $d \leftarrow \infty$ 
  handlePartiallyFormedPattern()
  for  $r \in P$  do
    if  $|r| < \frac{7}{8} \min_{r' \in Q \setminus \{r\}} |r'|$  then
       $r$  moves on its circle to create a  $\theta/8$ -shifted angle
    else if  $\{r' \neq r \text{ s.t. } |r'| < |r|\} = \emptyset$  then
       $c \leftarrow 1$  with probability  $1/2$ ,  $0$  otherwise
      if  $c$  then
         $r$  moves a distance  $|r|/8$  toward  $c(P)$ 
      else
         $r$  moves a distance  $\min(\frac{1}{2}(d - |r|), \frac{1}{7}|r|)$  away from  $c(P)$ 
else
   $r_1 \leftarrow$  unique robot with maximum view that does not hold  $C(P)$ 
  if  $\exists r \in [r_1, c(P)], P \cup \{r\} \setminus \{r_1\}$  is biangular then
     $r_1$  moves toward  $r$ 
  else
     $r_1$  moves toward  $c(P)$ 
```

Algorithm cleanExterior(i): remove robots outside C_i

Phase Condition: $i = 1$ or $|\text{interior}(C_{i-1} \cap \text{exterior}(C_i) \cap P')| = 0$
 $r \leftarrow$ smallest robot in $\text{exterior}(C_i)$
 $C \leftarrow$ circle centered at $c(P)$ that contains r

```
if  $|C \cap P| > 1$  then
   $r$  moves toward  $c(P)$  without reaching the circle of another robot nor  $C_i$ 
else
   $a \leftarrow \max_{r' \in C_i} \text{ang}(r_2, c(P), r')$  if  $\text{ang}(r_2, c(P), r) > a$  then
     $r$  moves toward  $c(P)$  to reach  $C_i$ 
  else
     $r$  moves on  $C_i$  in the direct orientation to have an angle  $(2\pi + a)/2$ 
```

Algorithm locateEnoughRobots(i): locate enough robots on C_i

Phase Condition: $|C_i \cap P'| \geq m_i$
 $r \leftarrow$ greatest robot in $\text{interior}(C_i)$
 $C \leftarrow$ circle centered at $c(P)$ that contains r

```
if  $|C \cap P| > 1$  then
   $r$  moves away from toward  $c(P)$  without reaching the circle of another robot nor  $C_i$ 
else
   $a \leftarrow \min_{r' \in C_i} \text{ang}(r_2, c(P), r')$ 
  if  $\text{ang}(r_2, c(P), r) < a$  then
     $r$  moves away from  $c(P)$  to reach  $C_i$ 
  else
     $r$  moves on  $C_i$  in the indirect orientation to have an angle  $a/2$ 
```

Algorithm removeRobotsInExcess(i): remove robot in excess on C_i

Phase Condition: $|C_i \cap P'| = m_i$

// Where $Poly(a, b)$ denotes the set of vertices of the regular a -gon centered at $c(P)$ that have the line $c(P)r_2$ as axis of symmetry union b points evenly distributed in the arc between angle 0 and π/a

if $i > 1$ **then**

$r \leftarrow$ smallest robot on C_i
 r moves toward $c(P)$ without reaching the circle of another robot

else

if robots the m_1 greatest robots on C_1 forms $Poly(m_1, 0)$ **then**

$r \leftarrow$ smallest robot on C_1
 r moves toward $c(P)$ without reaching the circle of another robot

else

 robots on C_1 form $Poly(m_1, |P \cap C_1| - m_1)$

Algorithm rotateRobotOnCircle: move the robots on their circle to reach their final destination

Phase Condition: $F' = P'$

Let r_1, \dots, r_{n-1} be the robots in P' in the lexicographic order of their polar coordinates in the global coordinate system.

Let d_1, \dots, d_{n-1} be the points of F' in the lexicographical order of their polar coordinates in the global coordinate system.

for $i = 1, \dots, n - 1$ **do**

$A \leftarrow$ the arc of the circle of r_i delimited by r_i and d_i that does not contain the point of angle 0

if $A \cap P' \neq \emptyset$ **then**

$c \leftarrow$ closest robot in $A \cap P'$
 $d \leftarrow$ point of A in the middle r_i and c
 $A \leftarrow$ the arc of the circle of r_i delimited by r_i and d

if $r_i \in C(P)$ **and then**

$d \leftarrow$ the farthest point on A so that $C(P)$ does not change

r_i moves on A toward d

- i) the pattern can be rotated so that robots in $P \setminus Q$ are located at points in F ,
- ii) among the m remaining points of F , denoted F_r , at least $m - 1$ are on $m - 1$ half lines, each containing exactly one robot in Q .

If those conditions are not both satisfied, the robot election is performed as previously described. Otherwise three cases can happen. Let d_1 be the radius of the smallest circle enclosing F_r . If $\mathcal{D}(d_1) \cap F_r \neq \emptyset$, let d_2 be the smallest radius such that $\mathcal{D}(d_1) \cap \text{exterior}(\mathcal{D}(d_2)) \cap F_r = \emptyset$, otherwise let $d_2 = d_1$. Also, let $d = (d_1 + d_2)/2$ ($\mathcal{D}(a)$ denotes the open disc centered at $c(P)$ of radius a)

In the first case, at least one robot r satisfies $|r| > d_1$. Then, all such robots move toward $c(P)$ to reach the circle of radius d_1 . After each robot reaches its destination, either the whole configuration forms F , or P from which we remove the robot with maximum view from F from which we remove a point with maximal view (the configuration is associated with the termination phase), or the configuration is still with the same phase and satisfies the second or the third case. If the configuration is no more associated with the same phase, then the configuration is static.

In the second case, at least one robot r satisfies $|d_1| \geq |r|$ and $|r| > |d|$. Then, all such robots move toward $c(P)$ to reach the circle of radius d . After each robot reaches its destination, the configuration satisfies the third case. During this phase the configuration is associated with the same phase since no robot reaches a point in F_r .

In the third case, the robots in Q are at most at distance d from $c(P)$. Then the robot election proceeds as previously described, except that a robot with destination p such that $|p| \geq d$ does not move. During this phase the configuration is associated with the same phase. Indeed, if $d_1 \neq d$, then there is at least one point in F_r that does not contain a robot (and it is not a point with maximum view since some points are closer to $c(P)$), otherwise (all points in F_r are at distance d to $c(P)$), there are at least two robots inside $\mathcal{D}(d)$ (because the whole configuration associated with this phase) and they cannot reach the circle of radius d .

C Pattern Formation when $|C(F) \cap F'| = 2$

Algorithm fixEnclosingCircle: locate the robot of $C(P) \cap P'$ on $C(P) \cap F'$ when $|C(F) \cap F'| = 2$

Phase Condition: $|C(F) \cap F'| \neq 2$ or there are only two robot in $C(P)$ located on the two point of $C(F) \cap F'$

```

if  $|C(P) \cap P'| = 2$  then
   $r \leftarrow$  greatest robot in interior  $C(P)$ 
   $C \leftarrow$  circle centered at  $c(P)$  that contains  $r$ 
  if  $|C \cap P| > 1$  then
     $r$  moves away from toward  $c(P)$  without reaching the circle of another robot nor  $C(P)$ 
  else
     $a \leftarrow \min_{r' \in C(P)} \text{ang}(r_2, c(P), r')$ 
    if  $\text{ang}(r_2, c(P), r) < a$  then
       $r$  moves away from  $c(P)$  to reach  $C(P)$ 
    else
       $r$  moves on  $C(P)$  in the indirect orientation to have an angle  $a/2$ 
  else
     $r \leftarrow$  greatest robot in  $C(P)$ 
     $r' \leftarrow$  smallest robot in  $C(P)$ 
    if  $r$  and  $r'$  are located the points of  $C(P) \cap F'$  then
       $r'' \leftarrow$  second smallest robot in  $C(P) \cap P'$ 
       $r''$  moves toward  $c(P)$  without reaching the circle of another robot
    else
      Let  $r_1, \dots, r_k$  be the other robots in  $C(P) \cap P'$  in the lexicographical order of their polar coordinates
      // perform the following movements while preserving  $C(P)$  and the ordering of robots
       $r$  moves on  $C(P)$  toward the greatest point in  $C(P) \cap F'$ 
       $r'$  moves on  $C(P)$  toward the smallest point in  $C(P) \cap F'$ 
      for  $i = 1, \dots, j$  do
         $a \leftarrow \text{ang}(r_2, c(P), r') + i \times (\text{ang}(r_2, c(P), r') + \text{ang}(r_2, c(P), r)) / (j + 1)$ 
         $r_i$  moves on  $C(P)$  toward the point in  $C(P) \cap F'$  with angle  $a$ 

```

We execute this special phase before executing the first sub-phase of phase 4.3, if $|C(F) \cap F'| = 2$. If $|C(F) \cap F| = 2$ and there are not exactly two robots in $C(P)$ located at the two points in $C(F) \cap F$, then the following is executed.

If there are only two robots on $C(P)$, then the greatest robot in $\text{interior}(C(P))$ reaches $C(P)$, while remaining smaller than robots in $C(P)$ (see Action *locateEnoughRobots(i)*). Now, there are at least three robots on $C(P)$. The greatest robot r in $C(P)$ moves toward the greatest robot in $C(F)$, the smallest r' moves toward the other point in $C(P) \cap F'$, and the other robots choose evenly distributed destinations between r and r' . Those movements are done while keeping $C(P)$ and the ordering unchanged. The smallest robot is chosen for r' instead of the second greatest so that no robot can prevent r' to reach the smallest point in $C(P) \cap F'$, especially if it has a null angle. Once r and r' reach their destination, the other robots can leave $C(P)$, starting from the smallest. Those last movements change the ordering of r , so that it becomes the second greatest robot.

D Omitted Proofs

D.1 Proof of Theorem 1

Theorem 1 (restated). *For a configuration P , if $\rho(P) > 1$, then P has a unique, non-empty, CEB-set.*

Proof. If $\rho(P) > 1$, then the configuration is regular or sym-regular. If P is biangular or equiangular, then $\text{CEB}(P) = P$. Otherwise, if P is regular, then the first set S in the construction of Q (algorithm **constructCEB-Set**) does not hold P , so S is added to Q , which becomes non-empty. If P is sym-regular and not regular, then the set S of smallest robots always contains one or two robots. Since $n \geq 5$, the while loop is executed at least three times. As we cannot have three disjoint subset of P that hold P , Q is not empty. \square

D.2 Uniqueness of the shifted robot

Let P be a n -robot configuration with $n \geq 5$. $\theta(P)$ is the minimum angle between two robots in P , centered at the center $c(P)$. We show that, for a shifted robot to be unique, it is sufficient that the shift angle is at most $\theta(P)/2$, and the orientation of the shift (the inverse of the orientation in which we have to rotate the robot to create a CEB-set configuration) must reduce the minimum angle it forms with the other robots of the configuration. In the sequel, let $W(P)$ denote the position of the Weber point of set P .

Lemma 3. *Let P be a n -robot configuration and $P' = P \setminus \{r\} \cup \{r'\}$. If $\vec{u} = \overrightarrow{W(P')W(P)}$, then for any point $p \in P \setminus \{r\}$, we have:*

$$\cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)r}\right)\right) - \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P')r'}\right)\right) \geq \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P')p}\right)\right) - \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)p}\right)\right)$$

Proof. It is known that $W(P)$ is such that

$$\sum_{p \in P} \vec{v}_p = 0$$

where \vec{v}_p is the normalized vector $\vec{v}_p = \frac{\overrightarrow{W(P)p}}{|\overrightarrow{W(P)p}|}$, or any vector such that $\|v_p\| \leq 1$ if $p = W(P)$. The equality is true by taking only the \vec{u} component of each vector, which implies

$$\sum_{p \in P} \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)p}\right)\right) = 0$$

Where we assume that $\cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)p}\right)\right)$ can be any number in $[-1, 1]$ if $p = W(P)$. The same equality holds in P' . By subtraction, we get:

$$\cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)r}\right)\right) - \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P')r'}\right)\right) = \sum_{p \in P \setminus \{r\}} \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P')p}\right)\right) - \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)p}\right)\right) \quad (1)$$

One can observe that, for every point $p \in P \setminus \{r\}$, we have:

$$\cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P')p}\right)\right) \geq \cos\left(\text{ang}\left(\vec{u}, \overrightarrow{W(P)p}\right)\right)$$

so that each term in the sum of the right hand side of equation (1) is positive, which implies that the left hand side is not smaller than each term in the right hand side, Q.E.D. \square

Lemma 4. *Let P be a n -robot configuration and $P' = P \setminus \{r\} \cup \{r'\}$, with $\theta = \text{ang}_{\min}(r, W(P), r') \leq \theta(P)$, and $|r'|_{W(P)} = |r|_{W(P)} = \min_{p \in P} |p|_{W(P)}$. We have $\forall p \in P$, $\text{ang}_{\min}(W(P), p, W(P')) < \theta$.*

Proof. For this lemma, we assume that all angles are computed using a global orientation such that $\text{ang}_{\min}(r, W(P), r') = \text{ang}(r, W(P), r')$. Also we define $\vec{u} = \overrightarrow{W(P')W(P)}$ and $\overrightarrow{\text{arg}}_p(r) = \arg(\vec{u}, \overrightarrow{p}r)$

From the previous lemma, we have:

$$\cos(\overrightarrow{\text{ang}}_{W(P)}(r)) - \cos(\overrightarrow{\text{ang}}_{W(P')} (r')) \geq \cos(\overrightarrow{\text{ang}}_{W(P')} (p)) - \cos(\overrightarrow{\text{ang}}_{W(P)} (p)) \quad (2)$$

With the notations of Figure 3:

$$\cos(\tau) - \cos(\tau + \theta - \gamma) \geq \cos(\beta') - \cos(\beta' + \gamma') \quad (3)$$

We observe that $\pi \leq \overrightarrow{\text{arg}}_{W(P)}(r) < \overrightarrow{\text{arg}}_{W(P')} (r')$ is not possible, because this would mean that the half-lines $HL(W(P), r)$ and $HL(W(P'), r')$ intersect, which is not possible.

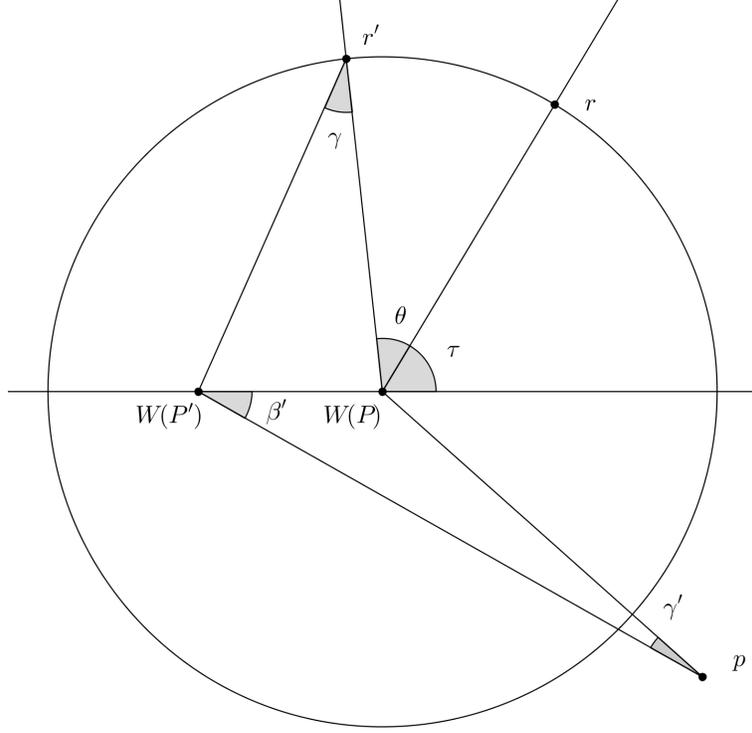


Figure 3: Part of the configuration P , for the proof that γ' cannot be greater than θ .

We can have three cases:

The case $\overrightarrow{arg}_{W(P)}(r) \leq \pi < \overrightarrow{arg}_{W(P')} (r')$: then

$$\cos(\overrightarrow{arg}_{W(P)}(r)) - \cos(\overrightarrow{arg}_{W(P')} (r')) \leq \cos(\overrightarrow{arg}_{W(P)}(r)) + 1 < 1 - \cos(\theta)$$

On the other hand, if $arg_{\min}(W(P), p, W(P')) \geq \theta$, we have:

$$\cos(\overrightarrow{arg}_{W(P')} (p)) - \cos(\overrightarrow{arg}_{W(P)}(p)) \geq 1 - \cos(arg_{\min}(W(P), p, W(P'))) \geq 1 - \cos(\theta)$$

which contradicts Equation 2, then we have $arg_{\min}(W(P), p, W(P')) < \theta$.

The case $\overrightarrow{arg}_{W(P')} (r') < \pi < \overrightarrow{arg}_{W(P)}(r)$: again, we have

$$\cos(\overrightarrow{arg}_{W(P)}(r)) - \cos(\overrightarrow{arg}_{W(P')} (r')) \leq 1 - \cos(\overrightarrow{arg}_{W(P')} (r')) < 1 - \cos(\theta)$$

with the same argument, this implies that $arg_{\min}(W(P), p, W(P')) < \theta$.

The case $\overrightarrow{arg}_{W(P)}(r) < \overrightarrow{arg}_{W(P')} (r') \leq \pi$: This case corresponds to the configuration shown in Figure 3. Using the notations of the figure, we want to prove that $\gamma' < \theta$ for any position of $r, r', W(P), W(P')$, and p (for an arbitrary $p \in P \setminus \{r\}$) verifying the conditions of the lemma. Like in previous cases, we suppose that $\gamma' \geq \theta$ and we reach a contradiction.

For simplicity, we suppose that $|W(P) - r| = 1$ and we characterize the configuration using the angles $\tau, \gamma, \theta, \tau'$ and the distance $l = |W(P) - p|$ rather than the positions of the points. To this purpose, we start by giving the distance $\omega = |W(P) - W(P')|$ with respect to the angles τ, γ , and θ using the law of sinus:

$$\omega(\tau, \gamma, \theta) = \frac{\sin(\gamma)}{\sin(\tau + \theta - \gamma)}$$

Then, the angle β' is computed using the law of sinus:

$$\beta'(\tau, \gamma, \theta, l, \gamma') = \beta'_1(\tau, \gamma, \theta, l, \gamma') = \arcsin\left(\frac{l \sin(\gamma')}{\omega(\tau, \gamma, \theta)}\right) \text{ or } \beta'(\tau, \gamma, \theta, l, \gamma') = \beta'_2(\tau, \gamma, \theta, l, \gamma') = \pi - \beta'_1(\tau, \gamma, \theta, \gamma', l)$$

which exists only if $\omega(\tau, \gamma, \theta) \geq l \sin(\gamma')$. Then, we define f_6 :

$$f_6 : (\tau, \gamma, \theta, l, \gamma', i) \mapsto \cos(\beta'_i(\tau, \gamma, \theta, l, \gamma')) - \cos(\beta'_i(\tau, \gamma, \theta, l, \gamma') + \gamma') - (\cos(\tau) - \cos(\tau + \theta - \gamma))$$

The domain of definition of f_6 is such that $\theta \in [0, \pi/4]$, $\gamma \in [0, \theta]$, $\tau \in [0, \pi - \theta]$, $l \geq 1$, $\gamma' \in [0, \pi/2]$, $i \in \{1, 2\}$, and $\omega(\tau, \gamma, \theta) \geq l \sin(\gamma')$. In the sequel, except stated otherwise, $f_6(\tau, \gamma, \theta, \gamma', l, i)$ denotes the image by f_6 of an arbitrary tuple $(\tau, \gamma, \theta, \gamma', l, i)$ in its definition domain. We want to show that if $\gamma' \geq \theta$, then $f_6(\tau, \gamma, \theta, \gamma', l, i) > 0$, which contradicts Equation 3. So now, for the sake of contradiction we suppose that $\gamma' \geq \theta$. The following claims 1 and 3 prove the lemma.

Claim 1: if $\omega(\tau, \gamma, \theta) > 1$, then $f_6(\tau, \gamma, \theta, l, \gamma', i) > 0$. First, if $\omega(\tau, \gamma, \theta) > 1$, we can use a simple lower bound:

$$\cos(\beta'_1(\tau, \gamma, \theta, l, \gamma') - \gamma') - \cos(\beta'_1(\tau, \gamma, \theta, l, \gamma')) \geq 1 - \cos(\gamma') > 1 - \cos(\theta)$$

Also, $\omega(\tau, \gamma, \theta) > 1$ implies that $\sin(\gamma) > \sin(\tau + \theta - \gamma)$, which in turn implies that $\gamma > \tau + \theta - \gamma$. Then we have:

$$\theta > \gamma > \tau + \theta - \gamma > \tau$$

so that

$$1 - \cos(\theta) \geq \cos(\tau) - \cos(\tau + \theta - \gamma) \Rightarrow f_6(\tau, \gamma, \theta, l, \gamma', i) > 0$$

Claim 2: if $\omega(\tau, \gamma, \theta) \leq 1$, The minimum of f_6 is the minimum of $f_3 : (\tau, \gamma, \theta) \mapsto f_6(\tau, \gamma, \theta, 1, \theta, 2)$. We want to show that the minimum of f_6 is reached when γ' is minimum (i.e., when $\gamma' = \theta$), when $l = 1$, and with $i = 2$.

Since $l > \omega(\tau, \gamma, \theta)$, then $\beta'_1 = \beta'_1(\tau, \gamma, \theta, l, \gamma') \geq \gamma'$. First, we have:

$$\cos(\beta'_2) - \cos(\beta'_2 + \gamma') = \cos(\pi - \beta'_1) - \cos(\pi - \beta'_1 + \gamma') = \cos(\beta'_1 - \gamma') - \cos(\beta'_1)$$

Since $\beta'_1 \leq \pi/2$, from the concavity of the cosine in $[0, \pi/2]$ and, if $\beta'_1 + \gamma' > \pi/2$, from the symmetry of the cosine with respect to $\pi/2$, we have:

$$\cos(\beta'_1 - \gamma') - \cos(\beta'_1) \leq \cos(\beta'_1) - \cos(\beta'_1 + \gamma')$$

So that the minimum of f_6 is obtained with $i = 2$. Moreover, since

$$\pi/2 \geq \beta'_1 \geq \beta'(\tau, \gamma, \theta, l, \theta)$$

Then, again, by concavity of the cosine we have:

$$\begin{aligned} \cos(\beta'_2) - \cos(\beta'_2 + \gamma') &= \cos(\beta'_1 - \gamma') - \cos(\beta'_1) \geq \cos(\beta'(\tau, \gamma, \theta, l, \theta) - \gamma') - \cos(\beta'(\tau, \gamma, \theta, l, \theta)) \\ &\geq \cos(\beta'(\tau, \gamma, \theta, l, \theta) - \theta) - \cos(\beta'(\tau, \gamma, \theta, l, \theta)) \end{aligned}$$

Finally, again, by concavity of the cosine, since $\beta'(\tau, \gamma, \theta, l, \theta) \leq \beta'(\tau, \gamma, \theta, 1, \theta)$, we have:

$$\cos(\beta'(\tau, \gamma, \theta, l, \theta) - \theta) - \cos(\beta'(\tau, \gamma, \theta, l, \theta)) \geq \cos(\beta'(\tau, \gamma, \theta, 1, \theta) - \theta) - \cos(\beta'(\tau, \gamma, \theta, 1, \theta))$$

So that $f_6(\tau, \gamma, \theta, l, \gamma', i) \geq f_6(\tau, \gamma, \theta, 1, \theta, 2) = f_3(\tau, \gamma, \theta)$.

Claim 3: if $\omega(\tau, \gamma, \theta) \leq 1$, then $f_3(\tau, \gamma, \theta) > 0$. Let $f_\tau : \tau \mapsto f_3(\tau, \gamma, \theta)$ for fixed γ and θ . By analyzing Function f_τ , we observe that f_τ is increasing and then decreasing, so that the minimum of f_τ is obtained with the greatest or smallest possible value of τ . For fixed $\theta \leq \pi/4$, and $\gamma \leq \theta$, we must have $\tau \leq \pi - \theta$ and $\tau \geq 2\gamma - \theta$ (because $\omega(\tau, \gamma, \theta) \leq 1$).

On the one hand, we have (where γ and θ are chosen in the definition domain of f_3):

$$\forall \gamma, \forall \theta, \quad f_3(\pi - \theta, \gamma, \theta) = 1 - \cos(\theta) + \cos(\theta) - \cos(\gamma) > 0$$

and on the other hand, we have:

$$\forall \gamma, \forall \theta, \quad f_3(2\gamma - \theta, \gamma, \theta) = 1 - \cos(\theta) - \cos(2\gamma - \theta) + \cos(\gamma) > 0$$

Where the last inequality is true since $\cos(\gamma) - \cos(\theta) > 0$.

□

Lemma 5. Let P be a n -robot configuration. Suppose the robots are indexed in the clockwise order r_1, r_2, \dots, r_n around the Weber point $W(P)$. Let $P' = P \setminus \{r\} \cup \{r'\}$, with $\text{ang}_{\min}(r, W(P), r') \leq \theta(P)/2$ and $|r'|_{W(P)} = |r|_{W(P)} = \min_{i \in [1..n]} |r_i|_{W(P)}$. Then, the robots in P' are ordered in the same way as in P around $W(P')$ (with r' instead of r).

Proof. Using the same notation as the previous lemma, we define $\vec{u} = \overrightarrow{W(P')W(P)}$ and $\overrightarrow{\text{arg}}_p(r) = \arg(\vec{u}, \vec{pr})$.

We suppose that the ordering of the robots in P is such that $\overrightarrow{\text{arg}}_{W(P)}(r_i) < \overrightarrow{\text{arg}}_{W(P)}(r_{i+1})$, for all $i \in [1..n-1]$ (a circular permutation of the ordering can make this true).

For all $i \in [1..n]$, let $r'_i = r_i$ if $r \neq r_i$, otherwise let $r'_i = r'$. We now show that r'_1, \dots, r'_n is an ordering of the robots in P' around $W(P')$. If $r \neq r_i$ and $r \neq r_{i+1}$, we have:

$$\begin{aligned} \overrightarrow{\text{arg}}_{W(P')} (r'_i) &< \overrightarrow{\text{arg}}_{W(P)} (r_i) + \theta(P)/2 && \text{from Lemma 4} \\ &\leq \overrightarrow{\text{arg}}_{W(P)} (r_{i+1}) - \theta(P)/2 && \text{by definition of } \theta(P) \\ &\leq \overrightarrow{\text{arg}}_{W(P')} (r'_{i+1}) && \text{from Lemma 4} \end{aligned}$$

If $r = r_{i+1}$, resp. $r = r_i$, then we have $\overrightarrow{\text{arg}}_{W(P)}(r) < \overrightarrow{\text{arg}}_{W(P')} (r')$, resp. $\overrightarrow{\text{arg}}_{W(P')} (r') < \overrightarrow{\text{arg}}_{W(P)}(r) + \theta(P)$, so that the previous inequality holds.

Overall, $\overrightarrow{\text{arg}}_{W(P')} (r'_i) < \overrightarrow{\text{arg}}_{W(P')} (r'_{i+1})$ for all $i \in [1..n-1]$, and the ordering is preserved. \square

Theorem 4. Let P be a n -robot ($n \geq 5$) biangular configuration that contains a shifted robot. The shifted robot is unique.

Proof. Let P be a biangular configuration with a shifted robot r (resp., r'), associated to the biangular set $P_{bi} = P \setminus \{r\} \cup \{r_{bi}\}$ (resp., associated to the biangular set $P'_{bi} = P \setminus \{r'\} \cup \{r'_{bi}\}$). For the sake of contradiction, we suppose that $r \neq r'$. From Lemma 5, the ordering is unchanged between P , P_{bi} and P'_{bi} . So, let r_1, \dots, r_n be any ordering of robots in P around $W(P)$. An ordering of robots in P_{bi} (resp., P'_{bi}) is obtained by replacing $r = r_{i_0}$ by r_{bi} (resp., $r' = r'_{i'_0}$ by r'_{bi}).

If $n \geq 5$ and P is equiangular, there are 3 robots in $P_{bi} \cap P'_{bi}$ ordered in the same way around the center of equiangularity. Their angles are uniquely determined by the difference between their indexes, indeed we have:

$$\text{ang}(r_i, W(P), r_j) = \text{ang}(r_i, W(P'), r_j) = \frac{|i-j|2\pi}{n}$$

Let $\text{arc}_\alpha(a, b)$ be the set of points p such that $\text{ang}(a, p, c) = \alpha$. $\text{arc}_\alpha(a, b)$ is a circular arc from a to b . Two circular arcs intersect in at most two points, so that there is at most one point $p \notin \{a, b, c\}$ in $\text{arc}_\alpha(a, b) \cap \text{arc}_\beta(b, c)$. This implies that that three robots are enough to deduce the position of the unique possible center of equiangularity of P_{bi} and P'_{bi} . With a given center, there cannot be two shifted robots. Indeed, if $c(P_{bi}) = c(P'_{bi})$, then the angles formed with r' and the other robots are the same as the robot r'_{bi} and $|r'_{bi}|_{c(P'_{bi})} = |r|_{c(P'_{bi})}$ i.e., $r' = r_{bi}$.

If $n \geq 8$ and P is biangular, there are 3 robots in $P_{bi} \cap P'_{bi}$ whose difference between indexes are even, and the previous argument holds.

The last case to consider is when $n = 6$ and P is biangular. In this case, if there are 3 robots in $P_{bi} \cap P'_{bi}$ whose difference between indexes are even, then the previous argument holds. Otherwise, without loss of generality, either $\{r_1, r_2, r_3, r_4\} \in P_{bi} \cap P'_{bi}$ or $\{r_1, r_3, r_4, r_6\} \in P_{bi} \cap P'_{bi}$.

In the first case, the center of biangularity is the unique point of intersection of $\text{arc}_{2\pi/3}(r_1, r_3)$ with $\text{arc}_{2\pi/3}(r_2, r_4)$. Indeed, the intersection is unique due to the ordering of robots.

In the second case, there can be two intersection points between $\text{arc}_{2\pi/3}(r_1, r_3)$ and $\text{arc}_{2\pi/3}(r_4, r_6)$. We suppose $r = r_2$ and $r' = r_5$. Let $\alpha \leq \beta$, resp. $\alpha' \leq \beta'$, the two angles in the string of angles $SA_{c(P_{bi})}(P_{bi})$, resp. $SA_{c(P'_{bi})}(P'_{bi})$. For simplicity, we can suppose that $\alpha = \arg(r_1, c(P_{bi}), r_{bi})$.

On the one hand, the robot r must be shifted in the orientation that decreases its angle with respect to the other robots *i.e.*, decreasing its angle with r_1 . On the other hand, r is in P'_{bi} so that r must verify $\arg(r_6, c(P'_{bi}), r) = 2\pi/3$. In particular, this implies that $\arg(r_1, c(P'_{bi}), r) < \arg(r_1, c(P_{bi}), r_{bi})$ so that $\alpha' < \alpha$. Applying the same argument with r' leads to the contradiction $\alpha < \alpha' < \alpha$. □

Theorem 2 (restated). *Let $n \geq 5$, and P be a n -robot configuration that contains a CEB-set with a shifted robot. Then, the shifted robot is unique.*

Proof. Let P be a n -robot configuration that contains a CEB-set Q with a shifted robot. If the whole configuration is biangular, Theorem 4 implies the result.

Otherwise, the configuration is regular or sym-regular, the center is the center of $SEC(P)$, and, even if different robots can be considered as shifted (when $|Q| < 5$), there can be only one robot that minimizes the angle with the other robots. □