Pattern Formation

Giuseppe Prencipe(✉)

University of Pisa, Pisa, Italy
giuseppe.prencipe@unipi.it

Abstract. The PATTERN FORMATION problem is one of the most important coordination problem for robotic systems. Initially the entities are in arbitrary positions; within finite time they must arrange themselves in the space so to form a pattern given in input. In this chapter, we will mainly deal with the problem in the $\mathcal{OBLOT}$ model.

Keywords: Pattern formation · Agreement · Multiplicity detection

1 Introduction

In this chapter, we will describe the PATTERN FORMATION problem, where the robots are required to form, in a not predetermined area of the plane where they operate, a pattern they receive in input. The pattern can be given as a set of points in the plane (expressed in their Cartesian coordinates), or as a geometric predicate (e.g., "form a *circle*").

The standard requirements are that, initially, no two entities are in the same position (i.e., there are no dense points), and that the number of points prescribed in the pattern and the number of robots are the same. The robots are said to *form the pattern* if, at the end of the computation, the positions of the robots coincide, in everybody's local view, with the points of the pattern (or satisfy the predicate). Depending on the application, the formed pattern may be *translated*, and/or *rotated*, and/or *scaled*, and/or *flipped* into its mirror position with respect to the initial pattern. If dense points are allowed in the robots configurations and in the pattern, the problem is called *pattern formation with multiplicity*.

The PATTERN FORMATION problem is practically relevant because, if the robots can form a given pattern, they can agree on their respective roles in a subsequent, coordinated action.

The more general and difficult version of this problem is the ARBITRARY PATTERN FORMATION problem, where the robots must be able to form *any* arbitrary pattern $\mathbb{P}$ they are given in input, starting from *any* arbitrary initial configuration where the robots occupy distinct location. The pattern formation problem, in its general as well as in the more specific versions, has been extensively investigated (e.g., see [1–11]).

P. Flocchini et al. (Eds.): Distributed Computing by Mobile Entities, LNCS 11340, pp. 37–62, 2019.
https://doi.org/10.1007/978-3-030-11072-7_3

2 Views and Symmetricity

Useful tools to study what patterns are formable by oblivious robots are based on the notion of *view* [9,12]: this notion is strictly related to that of a *symmetricity* of a set of entities in the plane (either point or robots). In this chapter we will just give a quick overview of these concepts, that will be detailed in Chap. 6.

Let Z_i denote the local coordinate system of robot r_i. The *global view* $GV_i(t)$ from robot r_i at time t is the infinite rooted tree defined as follows (refer also to Fig. 1):

1. The root v_i of $GV_i(t)$ corresponds to r_i.
2. Node v_i has $n-1$ children, one for each robot r_j, with $j \neq i$. The edge from node v_i to node v_j corresponding to r_j is labeled $((a,b),(c,d))$, where (a,b) is the position of r_j with respect to Z_i, and (c,d) is the position of r_i with respect to Z_j.
3. Node v_j, with $j \neq i$ has $n-1$ children, one for each robot r_l, with $j \neq l$; the edge from v_j to v_l is labeled $((a',b'),(c',d'))$, where (a',b') is the position of r_j with respect to Z_i, and (c',d') is the position of r_i with respect to Z_j.

Since in general a robot does not know the coordinate systems of the other robots, which are integral part of the definition of global view, the global view of a configuration is in general not available to the robots and, in most cases, impossible to derive.

Something that the robots can locally compute in absence of any other information is the *local view*. The *local view* $LV_i(t)$ *of* robot r_i at time t is the set of vectors $vec(r_i, r_j)$ for all $j \neq i$ with respect to Z_i. In other words, the local view $LV_i(t)$ corresponds to the information that r_i obtains when performing Look at time t. Two local views $LV_i(t)$ and $LV_j(t)$ are said to be *equivalent* $(LV_i(t) \equiv LV_j(t))$ if they are equal up to rotations, mirroring, and scaling.

An important property of the equivalence classes defined by the views (both in the case of global views, and of local views) is that they all have the same size.

Lemma 1 ([9]). *Given a configuration $\mathbb{E}$ at time t, all the equivalence classes of robots with the same global (resp., local) view, have the same cardinality m.*

Moreover, the robots can be partitioned into $\frac{n}{m}$ groups of m robots each, such that two robots have an equivalent view if and only if they belong to the same group. Note that, in the case of global view, the equivalence relationship is equality.

Lemma 2 ([9]). *If the system has* Chirality*, given a configuration $\mathbb{E}$ at time t, the robots in the same equivalence class form a regular m-gon, and the regular m-gons formed by all the groups have a common center.*

The size m of the equivalence classes, called *symmetricity*, is denoted by $\sigma(\mathbb{E})$ in the case of global views, and by $\rho(\mathbb{E})$ in the case of local views.

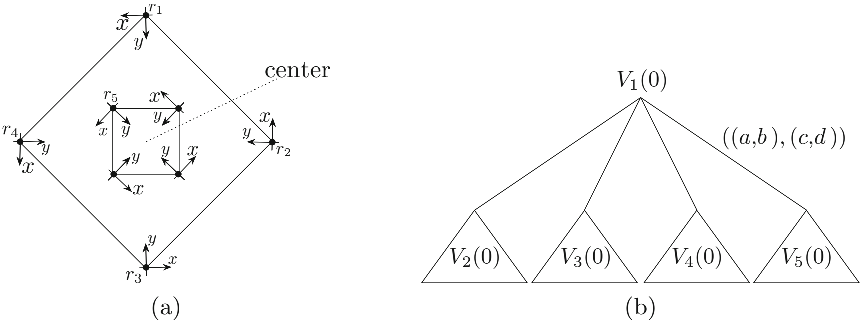

Fig. 1. (a) A configuration of robots. (b) The global view of r_1.

For example, in the configuration of robots depicted in Fig. 1(a) there are two classes of symmetry, each containing 4 robots, both when considering global and local views: In this case $\sigma(\mathbb{E}) = \rho(\mathbb{E}) = 4$. Moreover, since in this example there is chirality, Lemma 2 holds and the robots can be partitioned into 2 groups of 4 robots each group forming a 4-gon with a common centre.

Note that Lemma 2 does not hold when there is no chirality, i.e., when the axis of the coordinate systems of the robots are not rotationally symmetric. Consider, for example, the configuration depicted in Fig. 1(b). It is clear that all robots have the same global and local views, thus belong to the same equivalence class; they however do not form a single n-gon, but rather 2 distinct $\frac{n}{2}$-gons.

More examples are shown in Fig. 2 where, in all cases, the robots have the same local views; thus $\rho(\mathbb{E}_1) = \rho(\mathbb{E}_2) = \rho(\mathbb{E}_3) = n$. On the other hand, the global views are not always the same. More precisely, in Fig. 2(a), we have that $\sigma(\mathbb{E}_1) = 1$ because all global views are different; in Fig. 2(b), $\sigma(\mathbb{E}_2) = n$ because the global views are all identical; finally, in Fig. 2(c), $\sigma(\mathbb{E}_3) = \frac{n}{2}$.

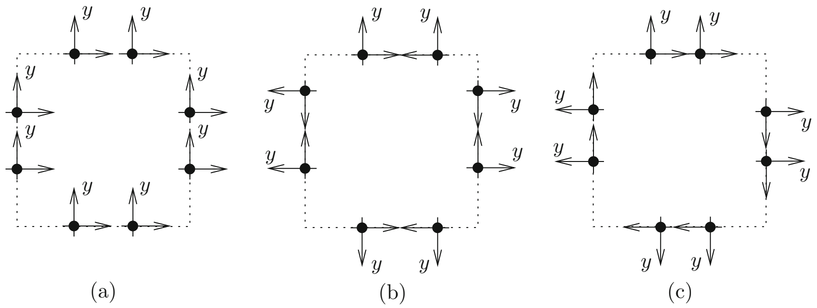

Fig. 2. Three configurations of robots: (a) $\mathbb{E}_1$, with $\rho(\mathbb{E}_1) = n$ and $\sigma(\mathbb{E}_1) = 1$; (b) $\mathbb{E}_2$, with $\rho(\mathbb{E}_2) = \sigma(\mathbb{E}_2) = n$; and (c) $\mathbb{E}_3$, with $\rho(\mathbb{E}_3) = n$ and $\sigma(\mathbb{E}_3) = \frac{n}{2}$.

So far we have defined the symmetricity in terms of configurations; this concept can be extended also to patterns. The symmetricity of a pattern $\mathbb{P}$ can be defined analogously to the one of a configuration with respect to local views from the points of the pattern. It shall be indicated with $\rho(\mathbb{P})$.

An equivalent alternative definition of symmetricity ρ and σ is based on rotations and groups, and is detailed in Chap. 6.

3 Arbitrary Pattern Formation

In the most general version of the problem, the robots are required to form any *arbitrary* pattern $\mathbb{P}$ they are given in input, starting from any *arbitrary* plain initial configuration; that is, they are required to solve the Arbitrary Pattern Formation problem. We note that, since rotation is allowed, two robots always form the desired pattern. Therefore we will assume to have at least three robots in the system. Also, we will assume that the robots operate under the $\mathcal{OBLOT}$ model.

3.1 Arbitrary Pattern Formation and Leader Election

A problem related to the Arbitrary Pattern Formation problem is the Leader Election problem: the robots in the system are said to *elect a leader* if, after a finite number of cycles, all the robots deterministically agree on (i.e., choose) the same robot l, called the leader. A deterministic algorithm that lets the robots in the system elect a leader in a finite number of cycles, given any initial configuration, is called a *leader election algorithm.*

The relationship between the arbitrary pattern formation problem and the leader election problem, is as follows:

Theorem 1 ([5]). *If it is possible to solve the* Arbitrary Pattern Formation *problem for $n \geq 3$ robots, then the* Leader Election *problem is solvable too.*

Proof. Let $\mathcal{A}$ be a pattern formation algorithm. Let $\mathbb{P}$ be a pattern defined in the following way:

1. All the robots but one are evenly placed on the same line $\mathcal{L}$; the distance between two adjacent robots is d; and
2. the last robot is on $\mathcal{L}$, but the distance from its unique adjacent robot is $2d$.

After all the robots execute $\mathcal{A}$ to form $\mathbb{P}$, the unique robot that has only one neighbor, and whose distance from it is $2d$, is identified as the leader.

We will now show that in general, the leader election problem is deterministically unsolvable.

Theorem 2 ([5]). *There exists no deterministic algorithm that solves the* Leader Election *problem, even in* $\mathcal{F}$sync *with* `Chirality`.

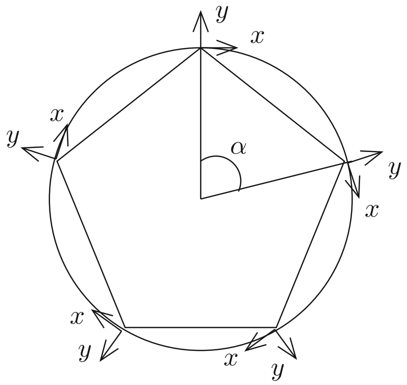

Fig. 3. Theorem 2: The unbreakable symmetry of a 5-gon.

Proof. By contradiction, let $\mathcal{A}$ be a deterministic algorithm for solving the Leader Election problem, and let us assume that the robots have no agreement on the local compasses (i.e., `Disorientation` holds). Consider any pattern different from a regular n-gon or a single point, and let the initial positions be such that the robots form a regular n-gon. Let $\alpha = 360°/n$ be the characteristic angle of the n-gon, and let the local coordinate system of each robot be rotated by α with respect to its neighbor on the polygon (see Fig. 3). In this situation, all the robots have the same (local) view of the world. Now, for any move that any one robot can make in its local coordinate system by executing algorithm $\mathcal{A}$, we know that each robot can make the same move in its local coordinate system. If all of them move in the exact same way at the same time (i.e., they move according to a synchronous schedule), they again end up in a regular n-gon or a single point. Therefore, by letting all the robots move at the same time in the same way, we always proceed from one regular n-gon or single point to the next. Hence, no leader can be elected. The same argument applies even if `Chirality` holds.

Thus, by Theorem 1, we can state the following:

Corollary 1. *In a system with $n > 2$ robots, the* Arbitrary Pattern Formation *problem is unsolvable.*

Furthermore, even if the robots agree on the direction and direction of one axis (agreement `k-Axes`, with $k = 1$), the Leader Election problem is still unsolvable when n is even: in the following, we will refer to this kind of agreement as `OneAxis`.

Theorem 3 **([5]).** *Let the robots agree only on the direction and orientation of one axis; there exists no deterministic algorithm that solves the* Leader Election *problem, hence the* Arbitrary Pattern Formation *problem, when n is even.*

Proof. By contradiction, let $\mathcal{A}$ be a deterministic leader election algorithm. Without loss of generality, let us assume that the robots agree on the direction and orientation of the Y axis, and consider an initial placement of the robots symmetric with respect to a vertical axis; i.e., each robot r has a *specular partner* $\widehat{r}$. In addition, let the local coordinate systems be specular with respect to the symmetry axis: the directions of the X axis of r and of the X axis of $\widehat{r}$ are opposite; thus the (local) view of the world is the same for r and $\widehat{r}$. In this setting, at time $t = 0$, both r and $\widehat{r}$ are in the same state; i.e., $\tau(r, 0) = \tau(\widehat{r}, 0)$. Consider now a semi-synchronous scheduler: robots are activated at discrete time instants; each robot is activated infinitely often; an active robot performs its operations instantaneously. Additionally, if a robot r is activated at time $t \geq 0$, the scheduler will activate at that time also $\widehat{r}$. As a consequence, if $\tau(r, t) = \tau(\widehat{r}, t)$, since the two robots execute the same protocol $\mathcal{A}$, their next state will still be the same: if r moves to d, $\widehat{r}$ moves to the point $\widehat{d}$ specular to d with respect to the symmetry axis. In other words, in this execution of protocol $\mathcal{A}$, $\tau(r, t) = \tau(\widehat{r}, t)$ for all $t \geq 0$. On the other hand, since $\mathcal{A}$ is an election protocol, it must exist a time $t' > 0$ such that a robot, say r' becomes leader. Since the leader is unique, $\tau(r', t') \neq \tau(r, t')$ for all $r \neq r'$, contradicting the fact that $\tau(r', t') = \tau(\widehat{r'}, t')$.

Let us consider now the converse relationship between the Arbitrary Pattern Formation problem and the Leader Election problem. Assume that all robots share a common protocol Leader($\mathbb{E}$) that, given any configuration $\mathbb{E}$, deterministically returns a unique leader in $\mathbb{E}$. We can now employ such a protocol to form an arbitrary target pattern $\mathbb{P}$, i.e., to solve the Arbitrary Pattern Formation problem, assuming that the robots agree on a common chirality. The overall idea of the algorithm consists of three main steps [13]: (1) the robots move to some appropriate positions, and build a kind of global coordinate system; (2) next, they compute the final positions to occupy in order to form the input pattern; (3) finally, the robots move towards these final positions, paying attention to maintain unchanged the global coordinate system.

In particular, given a set of points P and its $SEC(P)$, we call the *concentric enclosing circles* of $SEC(P)$ all the circles having the same center of $SEC(P)$ and passing through at least one point in P. Starting from a *leader configuration* (i.e., a configuration where a leader can be located), the robots first move to an *agreement configuration*:

Definition 1 (Agreement Configuration). *A configuration* $\mathbb{T}$ *is an agreement configuration if and only if both following conditions hold:*

1. *There exists a robot* r_l *in* $\mathbb{T}$ *such that* r_l *is the unique robot located on the smallest concentric enclosing circle of* $SEC(\mathbb{T})$*;*
2. *There is no robot at the center of* $SEC(\mathbb{T})$*.*

In order to achieve an agreement configuration from a leader configuration $\mathbb{E}$, the robots act as follows. If there is a robot r that is located at the center c of $SEC(\mathbb{E})$, let s be the closest robot to c among the robots in $\mathbb{E} \setminus \{r\}$, and p the median point on the segment $\overline{rs}$. Then, by moving r towards p, an agreement configuration is achieved. Otherwise (no robot is at c), we consider the smallest concentric enclosing circle of $SEC(\mathbb{E})$, call it $\mathcal{C}$; if there is only one robot on this circle, then the robots are already in an agreement configuration. Thus, let us assume there is more than one robot on $\mathcal{C}$. Now, the availability of protocol Leader() is exploited: let $r^* = \text{Leader}(\mathbb{E})$, and let r be the first robot on $\mathcal{C}$, according to the clockwise orientation, with respect to the half-line $\overrightarrow{cr^*}$ (recall that Assumption `Chirality` holds). By moving r towards the median point of the segment $\overline{rc}$, an agreement configuration is obtained. Note that the previous strategy works also when all robots are on $SEC(\mathbb{E})$ (i.e., when $\mathcal{C} \equiv SEC(\mathbb{E})$): the only difference is in the way robot r is chosen. In fact, in this case, r is the first *non-critical* robot on $\mathcal{C}$, i.e. the first robot on $\mathcal{C}$ whose movement would not change $SEC(\mathbb{E})$ (in this case r might coincide with r^*).

Once the robots are in an agreement configuration $\mathbb{T}$, they can also agree on their final positions: in particular, the center c of $SEC(\mathbb{P})$ is mapped onto the center o of $SEC(\mathbb{T})$; the pattern is rotated so that $\overrightarrow{or_l}$ is mapped onto $\overrightarrow{cs}$, with s the first non-critical point located on the smallest concentric enclosing circle of $\mathbb{P}$; and $\mathbb{P}$ is scaled with respect to the radius of $SEC(\mathbb{T})$ so that all the distances are expressed according to the radius of $SEC(\mathbb{T})$ (in particular $SEC(\mathbb{T}) = SEC(\mathbb{P})$).

Then, the robots occupy these positions, starting from those situated on SEC, and then on all the circles concentric to SEC from the largest to the smallest. During this phase, the final positions are maintained unchanged, by making sure that the robots remain in an agreement configuration until the pattern is formed. In particular, the protocol makes sure that no angle above 180° is created on SEC (otherwise the smallest enclosing circle changes), and that the leader of the agreement configuration remains the unique closest robot from the center of SEC and does not leave the radius where it is located. In other words,

Theorem 4 ([13]). *In $\mathcal{A}$Sync, assuming* `Chirality`*, for any $n \geq 4$ if the* Leader Election *problem is solvable, then the* Arbitrary Pattern Formation *problem is solvable.*

3.2 Arbitrary Pattern Formation and Compasses

The solvability of the Arbitrary Pattern Formation problem, and in general which patters can be formed regardless of the starting configuration, strictly depend on the level of agreement that the robots have about their local coordinate systems.

Following the ideas of the proof of previous Theorem 2, it is possible to show a *necessary* condition for the solvability of the Arbitrary Pattern Formation problem: the absence of common agreement on the coordinate system, leads to the inability to form arbitrary patterns.

Theorem 5 ([5]). *Without any agreement on the local compasses,* Arbitrary Pattern Formation *is* impossible*, even in $\mathcal{F}$Sync with chirality.*

As a consequence, some agreement is necessary.

Sense of Direction. Total agreement on the coordinate system (Assumption `ConsistentCompass`) is indeed *sufficient* to solve the Arbitrary Pattern Formation problem even in $\mathcal{A}$sync. To see how, consider the following protocol [5]:

1. Each robot establishes the (lexicographic) total order of the points of the local pattern (Fig. 4(a)).
2. Each robot establishes the (lexicographic) total order of the robots' positions retrieved in the last `Look` (Fig. 4(b)). As we will see, this order will be the same for all robots.
3. The first and second robots move to the positions matching the first and second pattern points. This movement can be performed in such a way that the order of the robots does not change (Fig. 4(c) and (d)). Once this is done, the first two robots' positions will determine the translation and scaling of the pattern (Fig. 4(e)).
4. All other robots go to the other points of the pattern. This can be done by moving the robots sequentially to the pattern's points. The sequence is chosen in such a way to guarantee that, after one robot has made even only a small move towards its destination, no other robot will move before that one has reached its destination (Fig. 4(f)).

We note that the final positions of the robots are not rotated w.r.t. the input positions; in other words the algorithm keeps the "orientation" given by the input pattern. Moreover, in this case Theorem 1 holds also for $n = 2$, since the rightmost and topmost robot in the system can always be identified as the leader.

Theorem 6 ([5]). *With* `ConsistentCompass`*,* Arbitrary Pattern Formation *is solvable in* $\mathcal{A}$sync*.*

Partial Agreement: Odd Number of Robots. Let us now consider the case when the robots have partial agreement: they agree only on the orientation of one axis, say Y; that is, there is common agreement also on the direction of the X axis, but not on its orientation (assumption `OneAxis`). Note that this case, if there is also chirality, would trivially coincide with the total agreement one.

As stated by Corollary 1, the Arbitrary Pattern Formation problem is unsolvable in general; furthermore, by Theorem 3, it is also unsolvable by an even number of robots when the Assumption `OneAxis` considered in this section holds, since symmetric initial configuration can impede the formation of arbitrary patterns. However, for breaking the symmetry, it is sufficient to know that the number n of robots is odd: in this case, in fact, either the robots are in a symmetric initial situation, in which there is a unique middle robot that will move in order to break the symmetry; or the initial situation is not symmetric, and this asymmetry can be used to identify an orientation of the X axis.

In more detail, let us define some *references* related to a set of points $\mathbb{E}$ that will be used in the following:

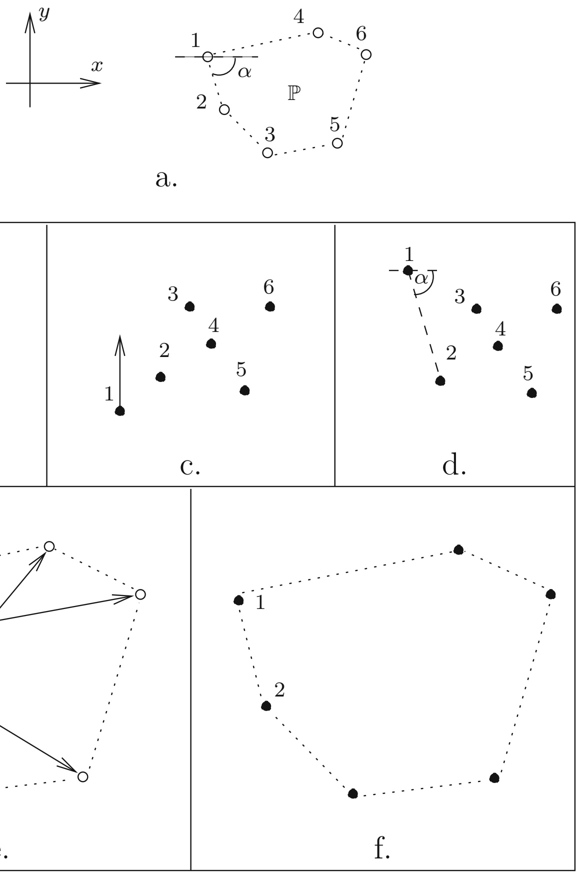

Fig. 4. An example of the arbitrary pattern formation protocol in presence of `ConsistentCompass`. (a) The input pattern $\mathbb{P}$. The robots have complete knowledge on the local coordinate systems. The numbers represent the lexicographical ordering the robots give to the points of $\mathbb{P}$, and $\alpha = \texttt{Angle(}p_1, p_2\texttt{)}$. (b) The robots sort the robots' positions retrieved in the last `Look` state, and compute $\beta = \texttt{Angle(}r_1, r_2\texttt{)}$. (c) r_1 moves in such a way that $\texttt{Angle(}r_1, r_2\texttt{)} = \alpha$. (d) The relative positions of r_1 and r_2 are such that $\texttt{Angle(}r_1, r_2\texttt{)} = \alpha$. (e) At this point, all the robots can translate and scale the input pattern according to $\overline{r_1 r_2}$. Then, all the robots, one at a time, reach the final positions of the pattern to form. (f) The final configuration.

- The two vertical lines that are tangent to the convex hull of $\mathbb{E}$, and the vertical axis $\Phi_m^{\mathbb{E}}$ that is in the middle between them.
- These three vertical lines delimit two regions (or *sides*): one to the left of $\Phi_m^{\mathbb{E}}$ and one to its right. Let $\mathcal{M}_{\mathbb{E}}$ and $\mathcal{L}_{\mathbb{E}}$ denote the side in $\mathbb{E}$ with more and less points, respectively. If the two sides have the same number of points, then $\mathcal{M}_{\mathbb{E}}$ is the rightmost side. If $|\mathcal{M}_{\mathbb{E}}| \neq |\mathcal{L}_{\mathbb{E}}|$, then $\mathbb{E}$ is said to be *unbalanced*; otherwise, we will call it *balanced*.

- Finally, $\varPhi^{\mathbb{E}}_{\mathcal{M}}$ denotes the one of the two axes tangent to the convex hull of $\mathbb{E}$ that lies in $\mathcal{M}_{\mathbb{E}}$, and $\varPhi^{\mathbb{E}}_{\mathcal{L}}$ the other.

We will describe now the protocol to form any pattern with an odd number of points, where the points are not all on the same vertical line. The case, where the robots have to form a vertical line is easier.

First, the robots check that the robots are not on the same vertical line $\varXi$; otherwise, the second topmost robot on this line, say r, moves towards its (local) right, up to a distance equal to the distance between the topmost and the bottommost robot on $\varXi$ (no other robot move until r reaches this distance). At this point, the references on both the input pattern $\mathbb{P}$ and on the observed configuration $\mathbb{D}$ can be computed: in particular, let $\varUpsilon_m = \varPhi^{\mathbb{P}}_m$, $\varUpsilon^+ = \varPhi^{\mathbb{P}}_{\mathcal{M}}$, $\varUpsilon^- = \varPhi^{\mathbb{E}}_{\mathcal{L}}$ the references in $\mathbb{P}$, and $K_m = \varPhi^{\mathbb{D}}_m$, and $K^+ = \varPhi^{\mathbb{D}}_{\mathcal{M}}$, $K^- = \varPhi^{\mathbb{E}}_{\mathcal{L}}$ the references in $\mathbb{D}$. The final goal of the robots is to find a way of mapping these two sets of references onto each other so that the final destinations the robots have to reach to form $\mathbb{P}$ can be uniquely computed.

To this aim, the robots need to *unbalance* $\mathbb{D}$, so that also an agreement on the orientation of the x axis cen be reached. If $\mathbb{D}$ is balanced, the symmetry that derives from having the two sides with the same number of robots is broken as follows. First all the robots[1] in $\mathcal{M}_{\mathbb{D}}$ are moved on K^+ and all the robots in $\mathcal{L}_{\mathbb{D}}$ on K^-. After all the robots have performed these movements, since $\mathbb{D}$ is still balanced and the total number of robots is odd, there is an odd number of robots on K_m: the topmost robot on K_m, say *top**, is selected to move towards its (local) right, so that an unbalanced configuration can be achieved. This movement is performed carefully since, as soon as *top** leaves K_m and enters the side to its right, the configuration will become unbalanced.

The fact that the configuration is unbalanced allows the robots to implicitly reach an agreement on the direction of the x axis; hence, on a *global coordinate system* (*GCS*): the common orientation of the x axis is given by mapping $\mathcal{M}_{\mathbb{P}}$ onto $\mathcal{M}_{\mathbb{D}}$.

Once the *GCS* has been established, the topmost robots on K^+ and on K^- (*top*$^+$ and *top*$^-$, respectively) move vertically on K^+ and on K^-, respectively, until they reach positions corresponding to the two topmost points on $\varUpsilon^+$ and $\varUpsilon^-$ in $\mathbb{P}$. Once *top*$^+$ and *top*$^-$ place themselves in the correct positions, they will never move again. At this point, the set of final positions of the robots can be easily computed, by scaling the pattern according to these mappings. Note that here the pattern does not need to be rotated.

Now, all robots are ready to reach their final destinations. Note that at this point it might be possible that the unbalancing process is not completed yet; i.e., *top** is still moving towards its destination. Should this be the case, the other robots can however detect it, and will not start their move until *top** stops (again, details can be found in [5]). The robots reach their final destinations sequentially:

[1] Note that, since at this time the robots still do not have a common agreement on the direction of the X axis, for some robots $\mathcal{M}_{\mathbb{D}}$ and $\mathcal{L}_{\mathbb{D}}$ might be different. All of them, however, agree on K_m.

- First, the robots in $\mathcal{S}^-$ (side of $\mathbb{D}$ where K^- lies) sequentially fill the final positions that are in $\mathcal{S}^-$. If there are more robots than available final positions, the "extra" robots are sequentially moved towards K_m, starting from the topmost robots that is closest to K_m.
- Second, the robots in $\mathcal{S}^+$ (side of $\mathbb{D}$ where K^- lies), except for the bottommost on K^+, sequentially fill the final positions in $\mathcal{S}^+$. If there are more robots than available final positions, the "extra" robots are sequentially moved towards K_m, starting from the topmost robots that is closest to K_m.
- Third, if there are still unfilled final positions in $\mathcal{S}^+$ (that is, there were not enough robots in $\mathcal{S}^+$ in the second step), the robots on K_m are sequentially moved in $\mathcal{S}^+$, starting from the topmost, to fill the final positions occupied by no robots.
- Fourth, if there are still unfilled final positions in $\mathcal{S}^-$ (that is, there were not enough robots in $\mathcal{S}^-$ in the first step), the robots on K_m are sequentially moved in $\mathcal{S}^-$, starting from the topmost, to fill the final positions still available.

At this point, all the robots not on K_m occupy the correct positions except one: the bottommost robot on K^+, say r.

- If there is an available destination in $\mathcal{S}^+$, then r goes there. At this point, all the robots but those on K_m are in correct positions. Note that now all available destinations are also on K_m: thus, the robots on K_m move sequentially (and only vertically on K_m) towards the available final destinations.
- If there are no available final positions inside $\mathcal{S}^+$ and $\mathcal{S}^-$, r moves towards K_m. Once it reaches the median axis, all the robots but those on K_m are in correct positions, and again the algorithm proceeds as in the previous case.
- If there is an available destination in $\mathcal{S}^-$, r first moves towards K_m. Then, the topmost robot on K_m moves in $\mathcal{S}^-$ on the last unfilled final position. Once also this position becomes occupied, only the robots on K_m must be adjusted, as in the two previous cases.

Thus, the above plus Theorem 3 imply the following:

Theorem 7 ([5]). *With* `OneAxis`*,* ARBITRARY PATTERN FORMATION *is solvable only if n is odd, and this can be done in* $\mathcal{A}$SYNC.

Partial Agreement: Even Number of Robots. By Theorem 3, an arbitrary pattern can not be formed by an even number of robots with `OneAxis`. In this section, we are interested in determining which class of patterns, if any, can be formed in this case starting from any initial position. Again, we will assume that the robots in the system have common agreement on the direction and orientation of only the Y axis, and that the number n of robots in the system is even.

We say that $\mathbb{P}$ is a *symmetric pattern* if it has at least one axis of symmetry Λ; that is, for each $p \in \mathbb{P}$ there exists exactly another point $p' \in \mathbb{P}$ such that p and p' are symmetric with respect to Λ (see Figs. 5(b), (c) and (d)).

The proof of the unsolvability result of Theorem 3 is useful to better understand what kind of patterns can not be formed, hence what kind of pattern

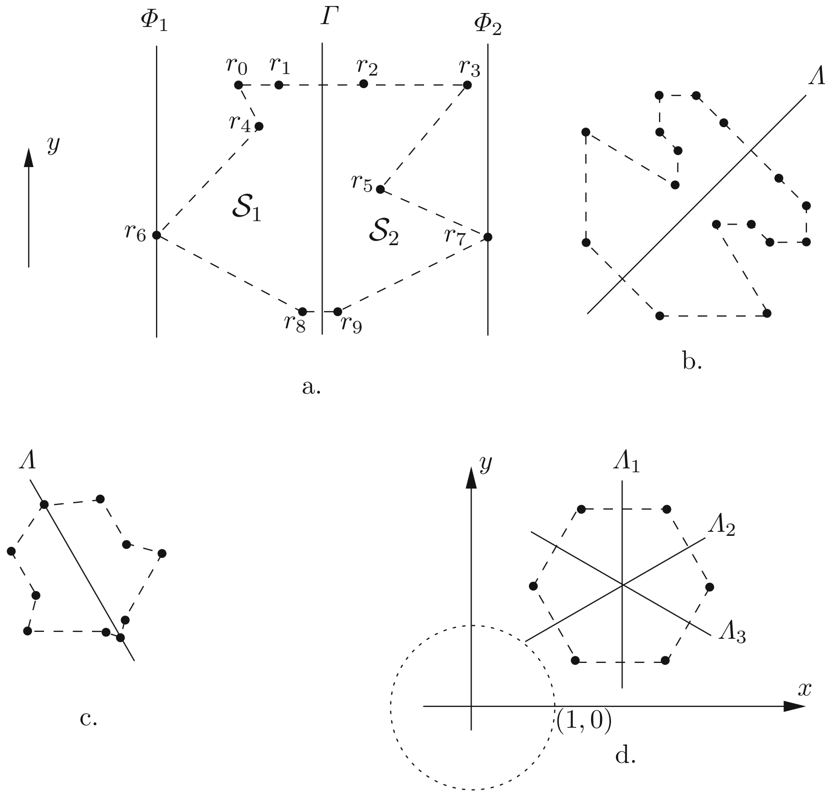

Fig. 5. (a) An unachievable asymmetric pattern. (b) An achievable pattern with one axis of symmetry not passing through any vertex. (c) An unachievable pattern. (d) An achievable pattern that has three axes of symmetry not passing through any vertex. Note that this pattern has also axes of symmetry passing through vertexes.

formation algorithms can not be designed. In fact, the ability to form a particular type of patterns would imply the ability to elect a robot in the system as the leader. Formally,

Theorem 8 ([5]). *If an algorithm $\mathcal{A}$ lets the robots form (a) an asymmetric pattern, or (b) a symmetric pattern that has all its axes of symmetry passing through some vertex, then $\mathcal{A}$ is a leader election algorithm.*

From Theorems 3 and 8, it follows that:

Corollary 2. *There exists no pattern formation algorithm that lets the robots in the system form (a) an asymmetric pattern, or (b) a symmetric pattern that has all its axes of symmetry passing through some vertex.*

Let us call $\mathfrak{T}$ the class containing all the arbitrary patterns, and $\mathfrak{P} \subset \mathfrak{T}$ the class containing only patterns with at least one axis of symmetry not passing through any vertex (e.g., see Figs. 5(b) and (d)); let us call *empty* such an axis. Corollary 2 states that if $\mathbb{P} \in \mathfrak{T} \setminus \mathfrak{P}$, then $\mathbb{P}$ can not be in general formed; hence, according to Part (b), the only patterns that might be formed are symmetric ones with at least one *empty axis*.

The idea behind the algorithm that solves the Arbitrary Pattern Formation problem with partial agreement and an even number of robots is as follows. First, the robots compute locally an *empty axis* of the input pattern $\mathbb{P}$, say Λ, and then rotate $\mathbb{P}$ so that Λ is parallel to the common understanding of the orientation of y; let us denote by $\mathbb{P}_R$ the rotated pattern.

If the robots lie all on the same line, the algorithm forces them to place on at least two distinct vertical lines, Γ and Γ' (this is achieved as for the odd case). Then, the topmost robot on Γ, say *Out*, and the topmost robot on Γ', say Out', move so that they place themselves in the correct position: in particular, since $\mathbb{P}_R$ is symmetric with respect to Λ, *Out* and Out' must place themselves to the same height. This is because, by Corollary 2, the input pattern can not be a vertical line.

At this point, the set of final positions can be computed, by scaling the input pattern with respect to $\overline{\Gamma\Gamma'}$, and by translating it so that the topmost point on the rightmost vertical axis tangent to $\mathbb{P}$ is mapped onto *Out*, and the topmost point on the leftmost vertical axis tangent to $\mathbb{P}$ is mapped onto[2] Out'.

At this point, the robots move to reach a balanced configuration, with each side containing half of the robots. The balancing is obtained as follows. Let $\mathcal{S}$ and $\mathcal{S}'$ be the two sides determined by Γ_m, the vertical median axis between Γ and Γ'.

- In the side that has more than $n/2$ robot (if any), the robots are moved sequentially (starting from the topmost with the smallest horizontal distance from Γ_m) towards Γ_m, using a path that avoids collisions, until there are exactly $n/2$ robots in that side.
- In a side that has $\leq n/2$ robots, the robots are moved towards the final positions in that side.
- The robots that are on Γ_m wait until $|\mathcal{S}| \leq n/2$ and $|\mathcal{S}'| \leq n/2$, and all the robots in the two sides are on a final position. At this point, sequentially (from the topmost) they move towards the final positions still available in the two sides. In fact, by the way the input pattern has been rotated, no final positions can be on Γ_m.

Thus, we can state the following:

Theorem 9 ([5]). *With* `OneAxis`*, when n is even only patterns in $\mathfrak{P}$ can be formed, and this can be done in $\mathcal{A}$*Async.

[2] Note that, since $\mathbb{P}_R$ is symmetric, nothing changes if the topmost point on the leftmost vertical axis tangent to $\mathbb{P}$ is mapped onto *Out*, and the topmost point on the rightmost vertical axis tangent to $\mathbb{P}$ is mapped onto Out'.

No Agreement. In absence of any additional assumption, and in particular in absence of any agreement on the compasses, Theorem 2 implies that no asymmetric pattern can be formed from all arbitrary initial configurations. Furthermore, as discussed later in Sect. 4, a symmetric pattern $\mathbb{A}$ with symmetricity $\sigma(\mathbb{A})$, can be formed only with the same or lower symmetricity. This means that the only patterns that might (possibly) be formed from all initial configurations are either an *n-gon*, or the *uniform circle* (i.e. a circle along which the robots are placed at equal distance), or the *point* (i.e., all robots are gathered at the same location). Note that this fact holds regardless of the synchronicity (i.e., even in $\mathcal{F}$SYNC).

The problem of forming a point (a.k.a. *Gathering*) and forming a uniform circle are important in their own right, and are analyzed respectively in Chap. 4 and Chap. 5, respectively. Interestingly, to date it is not known whether these problems can be solved in $\mathcal{A}$SYNC without additional assumptions; in the case of $\mathcal{S}$SYNC, they are both solvable.

3.3 Landmarks Covering: Formation of Visible Patterns

An interesting problem related to ARBITRARY PATTERN FORMATION (APF) is the LANDMARKS COVERING problem: in the space there are n points, the *landmarks*, visible to all robots[3]; the problem is for the robots to reach a configurations where at each landmark there is precisely one robot. A solution protocol must enable the robots to cover the landmarks, regardless of the location of the landmarks and of the initial location of the robots.

In other words, the LANDMARKS COVERING problem is precisely the ARBITRARY PATTERN FORMATION problem when the points of the input pattern are globally visible. Clearly, any solution to APF under some conditions, will solve also LANDMARKS COVERING under those conditions. The research interest is whether LANDMARKS COVERING can be solved more efficiently than APF, or with fewer conditions than APF, or in situations where APF is not (known to be) solvable. In terms of efficiency, the main goal of any LANDMARKS COVERING solution protocol is that of minimizing the robots movements, i.e., the total amount traveled by the robots to reach the final configurations in which all landmarks are covered.

Interestingly, unlike the ARBITRARY PATTERN FORMATION problem, the LANDMARKS COVERING problem can always be solved in $\mathcal{A}$SYNC, provided there is `Chirality`. Furthermore this can be done always with minimal travel costs and without collisions [14].

The solution strategy consists in the robots computing a unique perfect *matching* between robots and landmarks which minimizes the total travel costs from each robot to the landmark assigned by the matching; each robot then moves until it reaches the assigned landmark, avoiding collisions. The clear difficulty is to perform this process obliviously; to do so, the determined matching must be *invariant* to the movements of the robots towards their destination, so

[3] Equivalently, the position of the landmarks is known *a priori* to all robots.

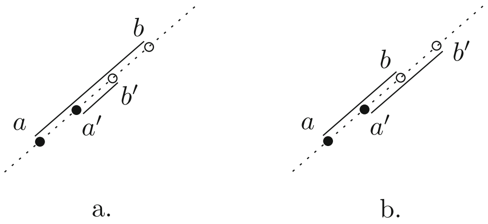

Fig. 6. Examples of matchings.

that each robot, every time it becomes active, can determine which landmark was initially assigned it, regardless of the progress made by the other robots towards their assigned landmarks.

Consider the initial configuration of the robots $\mathbb{A}$ and let $\mathbb{B}$ denote the pattern of the landmarks. We can view a perfect matching M from $\mathbb{A}$ to $\mathbb{B}$ as a set of pairs $\{(a, b)\}$ where a is a robot location in $\mathbb{A}$ and b is a landmark in $\mathbb{B}$, and its *cost* is the sum $\sum_{(a,b)\in M} |\boldsymbol{ab}|$ of the Euclidean distances between the matched points. Let $\mathcal{M}(\mathbb{A}, \mathbb{B})$ denote the set of all perfect matchings M of minimum cost between $\mathbb{A}$ to $\mathbb{B}$ such that for all distinct pairs $(a, b), (a', b') \in M$, the points a, a', b', b do not reside on the same line in that specific order. For example, M may not include the match shown in Fig. 6(a), but may include the pairs shown in Fig. 6(b). We call the matchings in this set *optimal.* It is easy to verify that $\mathcal{M}(\mathbb{A}, \mathbb{B}) \neq \emptyset$; note that there might be more than one optimal matching between $\mathbb{A}$ and $\mathbb{B}$.

We can compute a unique optimal matching, called *clockwise matching*, between two set of n distinct points, A and B, as follows:

(1) First consider the bipartite graph $G[A, B] = (V, E)$ whose vertex set $V = A \cup B$ comprises of the points of A and B, and where the edge set $E = \cup_{M\in\mathcal{M}(A,B)} M$ contains all pairs matched in at least one optimal matching.
(2) Consider now the connected components $G_1, G_2, \ldots, G_k$ of $G[A, B]$, and the periphery[4] C_i of component G_i; let A_i and B_i be the points of A in $G_i \setminus C_i$ and the points of B in $G_i \setminus C_i$, respectively.
(3) Consider next the subgraph $\hat{G}[A, B]$ of $G[A, B]$ recursively defined as follows: $\hat{G}[A, B] = \emptyset$ if $A = B = \emptyset$; otherwise $\hat{G}[A, B] = \cup_{1\leq j\leq k}(C_i \cup \hat{G}[\mathcal{A}_i, B_i])$. Note that each connected component Q of $\hat{G}[A, B]$ is either a cycle or a single edge.
(4) Finally, for each connected component Q_i of $\hat{G}[A, B]$, construct the matching W_i where $W_i = Q_i$ if Q_i is a single edge, otherwise, W_i is a clockwise tour $(a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)$ of Q_i.

[4] For a plane graph, the periphery is the boundary of the exterior face.

The *clockwise matching* $W[A, B]$ between A and B is just the union of the matchings W_i of all the connected components Q_i of $\hat{G}[A, B]$.

Important properties are that the *clockwise matching* W so determined is unique and indeed optimal; i.e., $W[A, B] \in \mathcal{M}(A, B)$. But the crucial fact is that these properties are invariant with respect to robots moving towards the matched landmarks. In fact,

Lemma 3 ([14]). *Let* $A = \{a_1, \ldots, a_n\}$*,* $B = \{b_1, \ldots, b_n\}$*, and* $= \{c_1, \ldots, c_n\}$ *be set of points which satisfy following:*

1. $\{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\} \in W[A, B]$
2. $c_i \in \overline{a_i b_i}$
3. *if there exists* $j \neq i$ *such that* $a_j \in \overline{a_i b_i}$ *then* $c_i = a_i$

Then $\{(c_1, b_1), (c_2, b_2), \ldots, (c_n, b_n)\} \in W[C, B]$.

Thus, the (collision avoiding) solution protocol is simply [14]:

```
Algorithm LandmarkCover
Assumptions: Visible Landmarks; Chirality.

1. Let $A = \{a_1, \ldots, a_n\}$ be the position of the robots (as returned by Look) and let $B = \{b_1, \ldots, b_n\}$ be the positions of the landmarks in my coordinate system.
2. Compute the clockwise matching $W[A, B]$. Let $a \in A$ be my position and $b \in B$ the landmark assigned to me in $W[A, B]$.
3. If $\forall a' \in A \setminus \{a\}, a' \notin \overline{ab}$, then move towards $b$.
```

Theorem 10 ([14]). *The* Landmarks Covering *problem can be solved in* $\mathcal{A}$SYNC *with minimal travel costs, provided there is* `Chirality`*.*

In other words, with `Chirality`, every *visible pattern* can be formed in $\mathcal{A}$SYNC.

4 Pattern Formation and Initial Configuration

The proof of Theorem 2 shows that, with no agreement on the local coordinate systems, the Arbitrary Pattern Formation problem cannot be solved. Thus, an interesting question is what patterns could be formed, in absence of common coordinate system, starting from a *specific* configuration $\mathbb{E}$. Once again, we will assume the $\mathcal{OBLOT}$ scenario.

4.1 Impossibility

The patterns that the robots can or cannot form starting from configuration $\mathbb{E}$ at time $t = 0$ are strictly related to the classes of equivalence derived from the definition of views (seen in Sect. 2).

If the views of two or more robots are identical, in some executions (e.g., under a scheduler that activates them always at the same time) those robots will always perform the same actions, without being able to break their symmetry; so, the patterns that can be possibly formed must have the same or higher symmetricity, but always a multiple of the original one.

Theorem 11 ([9]). *Starting from a configuration $\mathbb{E}$ with symmetricity $\sigma(\mathbb{E})$, it is* impossible *to form any pattern $\mathbb{P}$ with $\sigma(\mathbb{P}) < \sigma(\mathbb{E})$, or $\sigma(\mathbb{P}) \neq k \cdot \sigma(\mathbb{E})$ for some integer $k > 1$.*

In other words, if $\mathbb{E}$ is totally asymmetric (i.e., $\sigma(\mathbb{E}) = 1$), all patterns are potentially formable; on the other hand, if $\sigma(\mathbb{E}) = m > 1$, only patterns with the same symmetricity or with a symmetricity that is a multiple of m are candidate to be formable. Notice that this impossibility holds even if the robots are not oblivious. In case of systems with chirality, by Lemma 2, we obtain the inability to form a pattern that cannot be partitioned, as the initial configuration, in $\frac{n}{m}$ regular m-gons.

Theorem 12 ([9]). *In systems with* `Chirality`*, starting from a configuration $\mathbb{E}$ with symmetricity $\sigma(\mathbb{E}) = m$, it is* impossible *to form any pattern unless it is the union of $\frac{n}{m}$ regular m-gons all having the same center.*

4.2 Possibility

Once we know which are the only patterns that could be formed starting from a configuration $\mathbb{E}$, the questions become whether those patterns can be formed, and how. In $\mathcal{A}$SYNC, no answers are known. In the case of $\mathcal{S}$SYNC there are some conditional answers.

If the robots are *not oblivious* (recall that the impossibility holds even in this case), they can record all the snapshots in which they are active; the change of coordinates in two successive snapshots allows to detect movement and to measure it; hence information can be communicated by moving appropriate distances [9]. In particular, they can communicate their own coordinate systems and unit of measures, so that the complete views can be locally constructed and examined; once this is done, forming the pattern is straightforward.

We are however interested in oblivious robots, for which there is *no* memory, and hence no tool to record information, to detect and measure movement, and thus to communicate. Interestingly, it is possible for oblivious robots to form all the formable patterns [15], if the robots have `Chirality`, move with *fixed mobility* (possibly different for each robot) and know the *maximum movement* $\hat{\delta}$.

Theorem 13 ([15]). *A team of oblivious robots in $\mathcal{S}$SYNC with* `Chirality`*,* fixed mobility*, and* known maximum movement*, starting from configuration $\mathbb{E}$ with $\sigma(\mathbb{E}) = m$ can form any pattern $\mathbb{P}$ decomposable into $\frac{n}{m}$ regular m-gons all having the same center.*

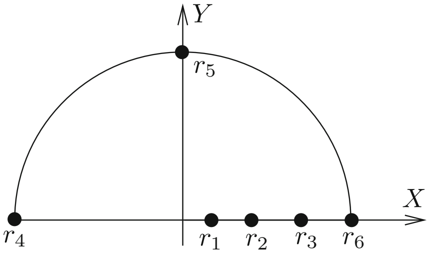

Fig. 7. A T-stable configuration with six robots.

Notice that, since the robots do not agree on a common coordinate system, the level of symmetry perceived by the robots (given by their local views) might not correspond to the actual level of symmetry of the global view which, as defined earlier, take into account also the coordinate systems.

Let $\mathbb{P} = p_1, \ldots, p_n$ be the pattern to be formed and let us assume that the robots start in n distinct positions. For simplicity we describe only the case when the pattern does not contain dense points. Moreover, we assume (again for simplicity) that each robot knows the origin of its own coordinate system, which does not change throughout the algorithm. The result still holds with some modifications also when these assumptions are removed. Also for simplicity we assume the unit distance of a robot coincides with $\hat{\delta}$.

The algorithm distinguishes the case when $\rho(\mathbb{E}) = 1$, and thus the initial configuration is totally asymmetric, even without considering the coordinate systems, from the case when $\rho(\mathbb{E}) > 1$.

Case $\rho(\mathbb{E}) = 1$. In this case the initial configuration $\mathbb{E}$ is perceived as asymmetric. This is the simplest case and also a building block possibly used in the other cases.

Since the symmetricity is 1 and there is chirality, a total order can be imposed on the robots, even in absence of a common coordinate system. The robots are in fact ordered in non decreasing order of their radii with respect to the centre c of the smallest enclosing circle $SEC(\mathbb{E})$ (for points with the same distance, ties are broken by using chirality). Let this order correspond to $r_1, \ldots, r_n$, where the robots are aware of their own index. The algorithm is designed in such a way that *SEC* will never change until the pattern is "almost" formed.

Intuitively, the robots move from $\mathbb{E}$ to a special configuration, called a *T-stable configuration*, where *SEC* contains exactly three robots on the circumference: two opposite on a diameter and the third at 90° from both, and no robots occupy the center (see Fig. 7). The robots can then agree on a common coordinate system by selecting as X the line passing through the two robots positioned opposite on the diameter of *SEC*, and as Y the line passing through c and through the third robot placed at 90° on the circumference.

The unit distance of this common coordinate system is chosen in a very specific way as $\frac{Rad}{2^l}$ where Rad is the radius of SEC, and l is the smallest positive integer such that $|r_j| < |p_j|$ for each $1 \leq j \leq n$, where $|r_j|$ (resp., $|p_j|$) indicates the distance from point r_j (resp., vertex p_j) to its own origin. This choice is made for the unit distance to be sufficiently small so that robots never move away from c while going towards their position to form the pattern. The robots now move one by one to their final destination following their order (which implies that robots closer to c move to their destination first). This order, combined with the fact that no robot has to move away from c in the process, guarantees that the magnitude of the unit distance does not change in the formation process, and that a robot that has reached its final position does not have to move anymore. The movements are performed without destroying the T-stable configuration, paying particular attention to the movements of the last three robots.

Case $\rho(\mathbb{E}) > 1$. When the robots perceive $\rho(\mathbb{E}) > 1$, it does not necessarily mean that $\sigma(\mathbb{E}) > 1$, because the different coordinate systems might induce more asymmetry. In this general case, the robots perform two procedures. First they try to move from $\mathbb{E}$ to a configuration that reflects a symmetry m that divides $\rho(\mathbb{P})$. Once/if such a situation is reached, they proceed to form the pattern. If, while changing symmetricity, they happen to form a configuration $\mathbb{E}'$ with $\rho(\mathbb{E}') = 1$, they instead form the pattern using the algorithm described in the previous case.

Let us describe the first procedure that allows the robots to appropriately reduce the perceived symmetricity ρ until it divides the symmetricity of the pattern to be formed.

The idea is the following. First the centre of the smallest enclosing circle c is identified. Point c is also the centre of symmetry of $\mathbb{E}$; that is, the unique point such that the robots can be divided in $\frac{n}{\rho(\mathbb{E})}$ groups each forming a regular $\rho(\mathbb{E})$-gon with centre c. Then each robot moves away from c in a straight line according to its coordinate system of a small amount. The amount is very carefully computed so to guarantee that: (1) it is smaller than the robot's unit distance and thus can be reached instantaneously in one step, (2) if two robots are located symmetrically with respect to c and have non symmetrical local coordinate systems, they will move of a *different* amount.

Depending on the activation schedule of the robots, the above procedure is shown to either break completely the symmetry in one step reaching a configuration $\mathbb{E}'$ where $\rho(\mathbb{E}') = 1$, or to reduce the symmetry eventually reaching, after repeated applications of the procedure, a configuration $\mathbb{A}$ such that $m = \rho(\mathbb{A})$ divides $\rho(\mathbb{P})$.

Now both the pattern and the configuration can be partitioned into $k = \frac{n}{m}$ regular m-gons all having the same center so to have a correspondence between each m-gon with a group of m robots. Let $\mathcal{R}_1, \ldots, \mathcal{R}_k$ be the k sets of robots and let $\mathcal{R}_k = \{r_1, \ldots, r_m\}$. Set $\mathcal{R}_k$ is special and it is used to create consistent coordinate systems. In fact, in this case it is not possible for the robots to agree on a common coordinate system based on a T-stable configuration (like for the asymmetric case). Because of the rotational symmetry induced by the $\frac{n}{m}$ regular

m-gons around c, each robot in one class of symmetry decides its destination individually. Each robot s_j in $\mathcal{R}_i$ chooses as X-axis the line passing through the common centre c and the closest among the robots in $\mathcal{R}_k$; while the unit distance is chosen as described earlier. As for the destination point: each robot chooses the closest location among the m possible locations and ties are broken, for example, by chirality. In this way, the coordinate systems of robots belonging to the same class are rotational symmetric with respect to c and intervals $\frac{2\pi}{m}$, and the destinations form a regular m-gon with the same centre that matches the one to be formed.

Notice that the algorithm described above works also for configurations and patterns with dense points, provided the robots have strong multiplicity detection. Indeed it allows to form in $\mathcal{S}$SYNC all patterns formable according to the strong *global* symmetricity $\sigma(\mathbb{E})$ of the initial configuration $\mathbb{E}$.

Possibility in $\mathcal{A}$SYNC. If we restrict ourselves to just plain patterns and initial configurations, and consider the weaker *local* symmetricity $\rho(\mathbb{E})$ of the initial configuration $\mathbb{E}$, than it is possible to form the patterns with symmetricity divisible by $\rho(\mathbb{E})$, even in $\mathcal{A}$SYNC:

Theorem 14 **([16]).** *A team of oblivious robots in* $\mathcal{A}$SYNC *with* `Chirality`*, starting from a plain configuration* $\mathbb{E}$ *with* $\rho(\mathbb{E})$ *can form any pattern* $\mathbb{P}$ *such that* $\rho(\mathbb{E})$ *divides* $\rho(\mathbb{P})$.

It is unknown whether this can be done without chirality.

5 Forming a Sequence of Patterns in $\mathcal{S}$SYNC

In this chapter we have discussed, under a variety of assumptions on the robots' capabilities and features, how to form a (possibly arbitrary) pattern given in input. A natural question is whether the robots can form not just a single pattern but a *series of distinct patterns*, given in a particular order, or, more generally of characterizing the series that can be formed. To enable a series of pattern to be formed, a protocol must guarantee that a robot that wakes up in an arbitrary configuration can, in spite of its obliviousness, figure out what pattern in the sequence is being formed so to join the others in performing the required tasks. In other words, a solution must provide, through the robots' movement some form of memory in an otherwise memoryless system.

In this section we consider $\mathcal{OBLOT}$ robots with `Chirality`, in $\mathcal{S}$SYNC under unlimited mobility (i.e., all robots always reach their destinations when performing their move). The focus is on *infinite* series: *periodic* (or cyclic) series $\mathbb{S}^\infty = \langle \mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_m \rangle^\infty$, i.e. the periodic repetition of a finite series $\mathbb{S}$ of distinct patterns. The results are then generalizable to infinite *aperiodic* series. Three different scenarios are analyzed, depending on the level of anonymity of the robots: completely anonymous robots, visibly indistinguishable but ordered set of robots and distinctly labeled robots.

Before describing the three scenarios, we introduce some special patterns needed in the rest of the section: (1) `POINT` is the pattern consisting of a single point; (2) TWO-POINTS is the pattern consisting of exactly two points; (3) POLYGON(k), for any $k \geq 3$, is the pattern consisting of points $p_1, p_2, \ldots, p_k$ that are vertexes of a regular convex polygon of k sides.

5.1 Anonymous Robots

Consider n identical robots starting from distinct locations. Central to the anonymous case is the notion of symmetry in a configuration, which is quantified using the concept of *centered view*, and *centered symmetricity*, $\hat{\rho}$, a slight modification of the notion of local view and of symmetricity ρ discussed in Sect. 4.

If r_i is not located at the centre of the smallest enclosing circle, its centered view $CV_i(t)$ contains the coordinates of all the other robots considering as origin $(0, 0)$ its own position and as $(1,\ 0)$ the position of the center. On the other hand, if r_i is in the centre of the smallest enclosing circle, the origin is still the location of r, but any robot r_j whose view $CV_j(t)$ is minimum among all the other robots is thought to be at coordinate $(1, 0)$. Finally, no information about the coordinate system of the robots is available in these views because they are assumed unknown and not necessarily consistent.

Notice that, given any arbitrary configuration $\mathbb{E}$, there is a total order of the distinct centered views of the robots in $\mathbb{E}$, in spite of their anonymity. The elements of CV_i can be ordered lexicographically to obtain an ordered sequence $Q(CV_i)$, for each robot $r_i \in \mathbb{E}$. For any two robots r_i and r_j, the ordered sequences $Q(CV_i)$ and $Q(CV_j)$ contain the same number of elements and these sequences can be ordered lexicographically. So, $CV_i < CV_j$ if and only if $Q(CV_i)$ is lexicographically smaller than $Q(CV_j)$.

An obvious consequence of anonymity is that from a configuration $\mathbb{E}$ consisting of anonymous robots at w distinct locations, a configuration $\mathbb{E}'$ where the robots occupy more than w distinct locations might not be reachable, which restricts the size of patterns in any formable series of patterns. To form repetitively any series $\mathbb{S}$ of patterns, all the patterns in $\mathbb{S}$ should be of the same size. Thus, only patterns of size n are considered, where n is the number of robots. Each robot starts from a distinct location and during the pattern formation algorithm, no two robots should occupy the same location (i.e. no dense points are allowed). Moreover, those patterns are indeed formable.

The formation algorithm is based on the identification of special configurations: the *bi-circular* and the *q-symmetric-circular* configurations. Before giving an intuition of the technique employed, we define these special configurations (see Fig. 8 for an example of a bi-circular configuration).

Definition 2 (BCC). *A configuration is called* bi-circular *(denoted by BCC) if: (i) there is a unique location (called the* pivot*), such that the smallest enclosing circle SEC containing all the robots, has diameter more than three times the diameter of the circle $\mathcal{C}$ containing all robots except those at the pivot; (ii) SEC and $\mathcal{C}$ intersect at exactly one point: the point directly opposite the pivot (called the* base-point*).*

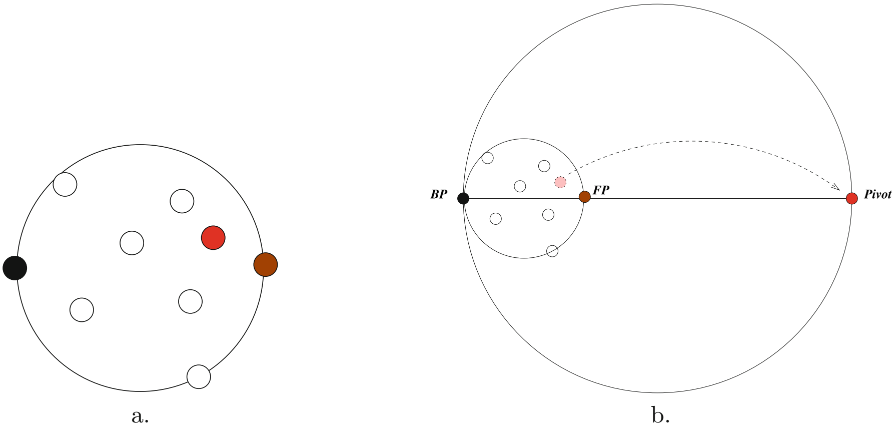

Fig. 8. (a) An arbitrary configuration of robots and the smallest enclosing circle. (b) A bi-circular configuration.

Definition 3 **(SCC).** *A configuration containing n robots is called* q-symmetric-circular *or, SCC(q), $1 < q < n$, if: (i) the smallest enclosing circle SEC has exactly q points on its circumference that are occupied by robots; (ii) all the other robots lie on or in the interior of a smaller circle $\mathcal{C}$ that is concentric to SEC such that $Diameter(SEC) \geq (5 + \sin^{-1}(\pi/q)) \cdot Diameter(\mathcal{C})$; (iii) there are no robots in the center of SEC.*

In both configurations, the former circle (SEC) is called the *primary enclosure* while the latter ($\mathcal{C}$) is called the *secondary enclosure*. The point on the secondary enclosure directly opposite the base-point is called the *frontier-point*. The ratio of the diameter of the primary enclosure over the diameter of the secondary enclosure is called the *stretch* of the configuration.

An interesting property of the bi-circular configuration is that in such a configuration the robots can agree on a coordinate system and define a unique way to order the robots. It can also be shown that from an arbitrary initial configuration either a particular type of BCC configuration or a particular type of SCC(q) configuration can always be formed. More precisely:

Lemma 4 **([17]).** *Starting from any configuration $\mathbb{E}$ with symmetricity $\hat{\rho}(\mathbb{E}) = q$, and for any $k \geq (5 + \sin^{-1}(\pi/q))$ we can reach a configuration $\mathbb{E}'$ such that either (i) $\mathbb{E}'$ is SCC(q') having stretch k, where $q' > 1$ is a factor of q, or, (ii) $\mathbb{E}'$ is BCC having stretch $k' = (k+1)/2$.*

It can also be shown that, once a bi-circular configuration containing n robots is formed, any pattern $\mathbb{P}$ of size n can be formed.

Lemma 5 **([17]).** *(i) In any bi-circular configuration, the robots can agree on a unique coordinate system. (ii) Starting from a bi-circular configuration with $n \geq 4$ robots in distinct locations, any pattern $\mathbb{P}$ of size n can be formed.*

Similarly, it can be shown that:

Lemma 6 **([17]).** *Starting from a configuration of type* SCC(*q*)*,* $q > 1$*, with* n *robots occupying distinct locations any pattern* $\mathbb{P}$ *such that the symmetricity* $\hat{\rho}(\mathbb{P}) = q \cdot a$*,* $a \geq 1$ *and size(*$\mathbb{P}$*)* $= n$ *can be formed.*

Based on the above properties, the idea of the algorithm for forming a cyclic series of distinct patterns $\langle \mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_m \rangle^\infty$ by n anonymous robots is the following.

Let F be a function that maps each pattern $\mathbb{P}_i$ to a real number $t_i = F(\mathbb{P}_i)$ that satisfies the condition of Lemma 6. To signal the formation of pattern $\mathbb{P}_i$, one of the following configurations is unambiguously used: either $\mathtt{SCC}(x)$ with stretch k_i, where x is any factor of q or, configuration BCC with stretch $k'_i = (k_i + 1)/2$. Due to Lemma 4 it is possible to form one of these configurations starting from an arbitrary configuration of symmetricity q. By computing the stretch of the configuration, the robot can then identify which pattern $\mathbb{P}_i$ is being formed. The robots can then form, by Lemmas 5 and 6, pattern $\mathbb{P}_i$. During the formation of pattern $\mathbb{P}_i$, at each intermediate configuration, each robot can uniquely identify which pattern is being formed. Once the pattern has been completed the resulting configuration has symmetricity q. Hence, by Lemma 4, it is again possible to form a SCC or BCC configuration having the appropriate stretch for the next pattern $\mathbb{P}_{i+1}$ in the sequence. Using this technique, the robots can move from one pattern to the next, and thus they can form the required sequence of patterns.

Theorem 15 **([17]).** *In* $\mathcal{S}$Sync *with* unlimited mobility *and* chirality*,* n *anonymous robots starting from distinct locations in an arbitrary configuration* $\mathbb{E}$*, can form a cyclic series of distinct patterns* $\langle \mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_m \rangle$*, each of size* n*, if and only if* $\hat{\rho}(\mathbb{P}_i) = \hat{\rho}(\mathbb{P}_j) \geq \hat{\rho}(\mathbb{E})$ $\forall i, j \in \{1, 2, \ldots m\}$*.*

The condition imposed by the previous theorem on the kind of patterns in the sequence can be relaxed if the robots are equipped with lights: this scenario will be analyzed in Chap. 11.

5.2 Robots with Distinct Visible Identities

Let us consider now the case when each robot r_i has a unique identity ID_i (w.l.g, $ID_i = i$) and any other robot can see this identity. During the Look operation, a robot r_i obtains a snapshot containing (j, x_j, y_j) tuples where $j \neq i$ and (x_j, y_j) is the location of the j-th robot, with respect to the local coordinate system of robot r_i. In this case, even in absence of agreement on directions, the symmetry among the robots can be broken by the use of distinct labels. The view of each robot is unique as it contains information about both the identities and locations of the other robots. Thus, there are no symmetric configurations. Moreover, as opposed to the anonymous case, robots can be allowed to form dense points, since the robots can be separated later, if required.

When there is only one robot, the only pattern that can be formed is obviously POINT. With $n = 2$ robots, only two patterns can be formed: POINT and TWO-POINTS and it is easy to form the sequence $(\mathtt{POINT}, \mathtt{TWO\text{-}POINTS})^\infty$, by movement of a single robot (say r_2). The more interesting cases occur when there are at least three robots (i.e., $n \geq 3$), in this case any sequence of distinct patterns $\mathbb{S} = \langle \mathbb{P}_1, \mathbb{P}_2, \ldots \mathbb{P}_m \rangle$ can be formed, with the only restriction that each pattern $\mathbb{P}_i$ has at most n points. A description of the algorithm is given below.

Robots r_1, r_2 and r_n have special roles. In particular, r_1 and r_2 remain fixed in distinct locations for the entire algorithms serving as fixed points of reference for the other robots. The idea is to apply a known function F to each pattern $\mathbb{P}_j$ so to obtain a real number $w_j = F(\mathbb{P}_j)$, $w_j \in (1, \infty)$ (distinct for every pattern). Before forming pattern $\mathbb{P}_j$, robot r_k moves to a location between r_1 and r_2 such that the ratio of distances $dist(r_1, r_2)/dist(r_1, r_n)$ is equal to w_j. This is the signal for the other robots to indicate which pattern is being formed. Each robot r_i, $2 < i < n$ can compute the location where it should move to in order to form pattern $\mathbb{P}_j$. Once each of these robots has moved into the correct positions, robot r_n moves to complete the pattern. During the execution of the algorithm every configuration of the robots (excluding at most the first two configurations) either corresponds to some pattern $\mathbb{P}_l \in \mathbb{S}$, or is an intermediate configuration which signals the formation of $\mathbb{P}_l$ (i.e. where r_1, r_2, and r_n maintain a ratio of $w_j = F(\mathbb{P}_l)$). The function F must be chosen in such a way that the ratio $dist(r_1, r_2)/dist(r_1, r_n)$ in an actual pattern never matches any values in the range of F. Thus, each robot can unambiguously determine the location that it needs to move to, by looking at the current configuration.

This algorithm works for any sequence of patterns not containing the `POINT` pattern. In order to include the `POINT` pattern in the sequence of patterns formed, small modifications must be done to the algorithm in the behaviour of robots r_2 and r_n. Based on the algorithm above, the author conclude that:

Theorem 16 ([17]). *In* $\mathcal{S}$SYNC *with* unlimited mobility *and* chirality*,* $n \geq 2$ *robots having distinct visible identities, can form any cyclic sequence of distinct patterns* $\langle \mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_m \rangle$ *provided that* $\forall i$*, size(*$\mathbb{P}_i$*)* $= n_i \leq n$*.*

5.3 Robots with Invisible Distinct Identities

In this case the identities of the robots are not visible to other robots. The robots are assumed to be ordered with labels $1, 2, 3, \ldots, n$ and each robot r_i knows its own label i, but it can not visibly identify the label of other robots. In this case, the information contained in the views of the robots is similar to the anonymous case. Thus, two robots may have identical views (in particular, robots at the same location have identical views). However, since the robots have distinct identities, they can execute different algorithms depending on their own labels.

Consider first the case when there are at least four robots. The `BCC` configuration, defined for the anonymous case, is used here as well to signal the formation of specific patterns in a series. As already mentioned, dense points are allowed and the algorithm must ensure that there is at least one robot at the pivot and one at the base-point of the bi-circular configuration.

From any arbitrary configuration $\mathbb{E}$ with more than 3 robots, a bi-circular configuration of any given stretch $k > 3$, can be formed by the movement of a single robot (this single robot will place itself in a pivot position).

The technique for forming any given pattern $\mathbb{P}$ starting from a bi-circular configuration of stretch k_i is as follows: As mentioned before, the bi-circular configuration can be formed by robot r_n jumping to the pivot location. Once the robots are in bi-circular configuration BCC with stretch k_i, robot r_1 and robot r_{n-1} occupy the base-point and the frontier-point. These three robots remain in their location while the other robots move to the required positions for forming pattern $\mathbb{P}$. The positions are assigned in the following manner. The points in the pattern $\mathbb{P}$ are mapped to locations in the bi-circular configuration such that the smallest enclosing circle of pattern $\mathbb{P}$ coincides with the secondary enclosure of the configuration and the base-point coincides with the lexicographically smallest point p_i on the smallest enclosing circle of $\mathbb{P}$, i.e., $p_i \in SEC(\mathbb{P})$ and $p_i \leq p_j$, for any $p_j \in SEC(\mathbb{P})$. Notice that this mapping is unique. Let $\Gamma(\mathbb{P})$ be the unique mapping obtain by each robot (i.e., the locations that correspond to points in the pattern $\mathbb{P}$). The elements of $\Gamma(\mathbb{P})$ are sorted in such a way that the first point is the base-point of the current BCC configuration of the robots, and all points which lie on the secondary enclosure $\mathcal{C}$ precede those that are located in the interior of $\mathcal{C}$. For $1 \leq i \leq size(\mathbb{P}_i)$ robot r_i is assigned the ith location in $\Gamma(\mathbb{P})$ and for $size(\mathbb{P}_i) < j \leq n$ robot r_j is assigned the n-th location in $\Gamma(\mathbb{P})$.

During the formation of a pattern $\mathbb{P}_i$ of size $size(\mathbb{P}_i)$, the algorithm ensures that the BCC configuration is maintained by keeping robots r_1, r_{n-1} and r_n stationary at the base-point, at the frontier-point and at the pivot positions respectively. Only when all the other robots have moved to their assigned location, robot r_{n-1} moves to its own assigned location, and also this is done ensuring that BCC is preserved with the appropriate stretch so that robot r_n can unambiguously move to the required position to complete the pattern.

The remaining cases are when there are exactly 2 or 3 robots. For $n = 2$, the case of invisible identities is same as that of visible identities. The case of $n = 3$ has been studied in [18] and an algorithm for forming any sequence of patterns of at most three points has been given. As mentioned before, the transformations between any two patterns of size 3 is straightforward and requires the movement of a single robot (say r_3). The only challenging scenario involves the formation of POINT and TWO-POINTS, where the intermediate configurations before and after forming POINT must be distinguished from the configuration forming TWO-POINTS. In conclusion:

Theorem 17 ([17]). *In* $\mathcal{S}$SYNC *with* unlimited mobility *and* chirality*, n robots having distinct invisible identities can form any cyclic sequence of distinct patterns $\langle \mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_m \rangle$ where $\forall i$, size($\mathbb{P}_i$) $\leq n$.*

References

1. Ando, H., Suzuki, I., Yamashita, M.: Formation and agreement problems for synchronous mobile robots with limited visibility. In: Proceedings of the 1995 IEEE Symposium on Intelligent Control, pp. 453–460 (1995)
2. Chatzigiannakis, I., Markou, M., Nikoletseas, S.: Distributed circle formation for anonymous oblivious robots. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 159–174. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24838-5_12

3. Défago, X., Konagaya, A.: Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In: Proceedings of 2nd Workshop on Principles of Mobile Computing, pp. 97–104 (2002)
4. Défago, X., Souissi, S.: Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. Theor. Comput. Sci. **396**(1–3), 97–112 (2008)
5. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Arbitrary pattern formation by asynchronous oblivious robots. Theor. Comput. Sci. **407**(1–3), 412–447 (2008)
6. Kasuya, M., Ito, N., Inuzuka, N., Wada, K.: A pattern formation algorithm for a set of autonomous distributed robots with agreement on orientation along one axis. Syst. Comput. Jpn. **37**(10), 89–100 (2006)
7. Katreniak, B.: Biangular circle formation by asynchronous mobile robots. In: Pelc, A., Raynal, M. (eds.) SIROCCO 2005. LNCS, vol. 3499, pp. 185–199. Springer, Heidelberg (2005). https://doi.org/10.1007/11429647_16
8. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. J. Robot. Syst. **13**, 127–139 (1996)
9. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: formation of geometric patterns. SIAM J. Comput. **28**(4), 1347–1363 (1999)
10. Tanaka, O.: Forming a circle by distributed anonymous mobile robots. Master's thesis, Department of Electrical Engineering (1992)
11. Wang, P.K.C.: Navigation strategies for multiple autonomous mobile robots moving in formation. J. Robot. Syst. **8**(2), 177–195 (1991)
12. Yamauchi, Y., Uehara, T., Kijima, S., Yamashita, M.: Plane formation by synchronous mobile robots in the three-dimensional euclidean space. J. ACM **64**(3), 16:1–16:43 (2017)
13. Dieudonné, Y., Petit, F., Villain, V.: Leader election problem versus pattern formation problem. In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC 2010. LNCS, vol. 6343, pp. 267–281. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15763-9_26
14. Fujinaga, N., Ono, H., Kijima, S., Yamashita, M.: Pattern formation through optimum matching by oblivious CORDA robots. In: Lu, C., Masuzawa, T., Mosbah, M. (eds.) OPODIS 2010. LNCS, vol. 6490, pp. 1–15. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17653-1_1
15. Yamashita, M., Suzuki, I.: Characterizing geometric patterns formable by oblivious anonymous mobile robots. Theor. Comput. Sci. **411**(26–28), 2433–2453 (2010)
16. Fujinaga, N.: Oblivious pattern formation algorithms for asynchronous mobile robots based on bipartite matching approach. Master's thesis, Department of Informatics, Kyushu University (2012)
17. Das, S., Flocchini, P., Santoro, N., Yamashita, M.: On the computational power of oblivious robots: forming a series of geometric patterns. In: Proceedings of 29th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 267–276 (2010)
18. Bouzid, Z., Lamani, A.: Robot networks with homonyms: the case of patterns formation. In: Défago, X., Petit, F., Villain, V. (eds.) SSS 2011. LNCS, vol. 6976, pp. 92–107. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24550-3_9