Forming Sequences of Geometric Patterns with Oblivious Mobile Robots

Shantanu Das · Paola Flocchini · Nicola Santoro · Masafumi Yamashita

Abstract We study the computational power of distributed systems consisting of simple autonomous robots moving on the plane. The robots are endowed with visual perception, allowing them to see each other, but they do not have any means of explicit communication with each other. Further the robots are oblivious, meaning that they always act based on their current perception of the environment, and they have no memory of the past. Such system of simple robots have been studied extensively with the objective of achieving coordinated tasks e.g. arranging the robots in a line or a circle. In fact it has been shown that obliviousness is not a limiting factor to form a *single* geometric pattern, however arbitrary. This paper aims to understand the computational limits imposed by the obliviousness of the robots by studying the formation of a *series* of geometric patterns instead of a single pattern. If such a series of patterns could be formed this would create some form of memory in an otherwise memory-less system. We show that, under particular conditions, oblivious robot systems can indeed form a given series of geometric patterns starting from any arbitrary configuration. More precisely, we characterize the series of patterns that can be formed by oblivious robot systems under various additional restrictions such as anonymity, asynchrony and lack of common orientation. These result! s provide strong indications that oblivious solutions may be obtained also for tasks that intuitively seem to require memory.

1 Introduction

1.1 Problem and Background

We are interested in understanding the computational power of distributed systems consisting of simple autonomous robots. The robots move on the plane, can see each other but cannot explicitly communicate with one another. This lack of direct communication capabilities means that all synchronization, interaction, and communication of information among the sensors take place solely by observing the position of the robots in the plane. Each robot has a local coordinate system (a set of Cartesian axes, an origin, and a unit of distance); however there might be no relationship between the coordinate items of different robots. There is no central coordinator; the robots are

---

S. Das
LIF, Aix-Marseille University & CNRS,
163 Avenue de Luminy, Marseille 13009, France.
Tel.: +33-491-829528 Fax: +33-491-829275 E-mail: shantanu.das@lif.univ-mrs.fr

P. Flocchini
School of Electrical Engineering and Computer Science, University of Ottawa, Canada. E-mail: flocchin@eecs.uottawa.ca

N. Santoro
School of Computer Science, Carleton University, Canada. E-mail: santoro@scs.carleton.ca

Masafumi Yamashita
Dept. of Computer Science and Communication Engineering, Kyushu University, Japan. E-mail: mak@csce.kyushu-u.ac.jp

indistinguishable, they have the same capabilities, and execute the same (deterministic) algorithm. Each robot computes its next location based on the visual information and moves to this location.

These systems have been extensively investigated by researchers from robotics, AI, control and, more recently, distributed computing (e.g., [1, 2, 4, 6–13, 16, 19, 22, 25–27]).

A particular class of these systems are those in which the robots are *oblivious*: they have no memory of the past, and do not rely on it for their computations. In other words, the current behavior of an oblivious robot depends only on the presently observed configuration of the robots but not on past history of observations and computations by the robot. Thus, a system of oblivious robots is inherently *self-stabilizing*; each robot may start from any arbitrary initial state and there is no need to initialize the system. For this reason there has been a strong interest in the study of such systems (e.g., [1, 2, 7, 8, 10, 12, 16, 19, 21, 26, 27]); for a recent account of the current research on oblivious robots see also [14]. Designing algorithms for oblivious robots is specially challenging and some simple tasks such as the gathering of two robots, are known to be impossible for oblivious robots [26].

The studies on the computational power of systems of oblivious robots have focused on determining what minimal capabilities are necessary so that the robots can perform simple basic tasks e.g. gathering at a point [2, 4, 8], or scattering uniformly in a given region [9]. Many such problems can be generalized to the abstract problem of (geometric) *pattern formation*. A pattern is represented by a set of points in the Euclidean plane that form some geometric figure such as a circle, a line, or some other arbitrary shape. Given a particular pattern as input, the robots must position themselves with respect to each other such that the location of the robots correspond to points in the pattern. Notice that *point formation* (i.e., formation of a single point) corresponds to the well known *gathering* or *rendezvous* problem, extensively studied in the literature (e.g., [1, 2, 8, 15, 21–23]). The *arbitrary pattern formation* problem, that is forming any pattern given in input, has also been studied [3, 16, 17, 20, 25, 26, 28]. Characterizations of patterns formable from a given initial configuration under various conditions have been given in [18, 26]. Formation problems have been investigated also for specific patterns in the case of oblivious robots, in particular for the circle [7, 10, 12]. Another problem that has been studied is that of moving in formation, called *flocking* [19, 24].

At a more general level, the research has focused on the characterization of which patterns are possible under what conditions [3, 16, 18, 26]. For example, if there is agreement about the local coordinate system (e.g., a compass), oblivious robots can form any pattern even if they are totally asynchronous [16]. It has been recently shown [27] that oblivious semi-synchronous robot can form exactly the same patterns that non-oblivious robot can, with one exception: point formation by two robots. These results indicate that, in most settings, simple robots can form a *single* geometric pattern, however arbitrary, in spite of their obliviousness. In other words, obliviousness is not a limiting factor to form a *single* pattern.

This naturally brings to the front the question of whether the robots can form not just a single pattern but a *series of patterns* in a particular order, and if so, what series can be formed. Notice that obliviousness limits what can be remembered. On the other hand, to enable a series of pattern to be formed, a protocol must guarantee that a robot that wakes up in an arbitrary configuration can still join the others in performing the required tasks. Thus, a formable series of patterns provides some form of memory in an otherwise memoryless system of robots. The problems examined here are integral components of the crucial research question on what are the computational limits imposed by the robots being oblivious.

1.2 Contributions and Organization

We study the series of patterns that may be formed by oblivious robots starting from any arbitrary configuration. An immediate observation (c.f. Lemma 1) is that no algorithm that terminates within a finite time can completely form any non-trivial series of distinct patterns; as explained later, an adversary can force any protocol to produce at most a single pattern, by appropriately choosing the initial configuration. Thus, in this paper, we concentrate on *periodic* (or cyclic) series S^∞, i.e. the periodic repetition of a finite series S of distinct patterns.

We consider the case of anonymous identical robots as well as that of distinguishable robots, i.e. robots having visibly distinct identities. We also consider the case of robots that are visibly indistinguishable from each other but each robot has distinct (invisible) identity; this allows the robots to execute distinct algorithms. When the robots have distinct visible identities, we prove that any series of distinct patterns can be formed provided that there are sufficiently many robots. The same result holds for robots having invisible but distinct identities. In case of anonymous robots, the series that may be formed depends on the symmetry in the initial configuration, quantified by the parameter called *symmetricity*. We characterize the series of patterns formable by anonymous, oblivious and unoriented robots. We also consider the special case when the robots agree on directions.

The paper is organized as follows. In the next section the model, terminology and basic properties are introduced. In Section 3 we study the most general (thus weaker) case of anonymous robots. The strongest case when robots have distinct visible identities is examined in Section 4. Finally the series of patterns formable when the robots have distinct but invisible identities is investigated in Section 5. We conclude the paper in Section 6 with some directions for future research.

2 The Model

2.1 The Robots

Each robot in the system has sensory capabilities allowing it to determine the location of other robots in the plane, relative to its own location. The robots also have computational capabilities which allow them to compute the location to move to. Each robot follows an identical algorithm that is preprogrammed into the robot. This algorithm may contain description of patterns that the robots are required to form. The robots may start from arbitrary locations in the plane.

The behavior of the robots can be described as follows. Each step taken by a robot consists of three stages: LOOK, COMPUTE and MOVE as described below. Between two steps taken by a robot, the robot may go to sleep for an arbitrary amount of time. We assume that time is discretized into rounds and at each round an arbitrary subset of the robots (selected by an adversarial scheduler) are active. The robots that are active in a round complete exactly one step in that round. This model of synchronization is called the *semi-synchronous* or SSYNC model in the literature and there exists many variants of this model depending on assumptions about the powers or limitations of the scheduler or the adversary. In our case, we impose only the following fairness constraint on the scheduler: In any infinite execution, each robot must be activated in infinitely many rounds.

We now describe the actions of the robot during each step.

- During the LOOK stage of a step, an active robot r gets a complete snapshot of the environment showing the current location of all the other robots. These locations are observed by robot r in terms of the local coordinate system and unit distance used by robot r at the time of observation. The coordinate system used by a robot may change at the beginning of each LOOK-COMPUTE-MOVE step, but remains invariant during the step[1].
- During the COMPUTE stage, an active robot executes an algorithm that determines its next location. The algorithm takes as input the information obtained from the LOOK operation and the identifier of the robot (if any), and outputs the next location of the robot in terms of the robot's current coordinate system. Note that the actions of each robot r depend only on the current configuration of robots as seen by r and do not depend on the previous history of moves (i.e. the robots are oblivious).
- During the MOVE stage of the step, the robot moves to the location computed during the COMPUTE stage. We assume rigidity of movements (as in [12, 21, 22, 24]) which means that a robot always reaches its intended destination in each step regardless of the distance.

The local coordinate system of a robot r in round t is denoted by $Z_{r,t}$. The origin of the coordinate system $Z_{r,t}$ is always the current location of the robot. The robots may not agree on a common sense of direction, but they agree on left-right orientation with respect to any given direction, i.e. the robots can distinguish clockwise from counter-clockwise. The visual capabilities of the robots allows them to detect multiplicity (i.e. they can count how many robots are at the same location). The robots may have distinct identities, and these identities may be visible, allowing other robots to identify and distinguish between robots; or, the robots may be all identical.

We denote by n the number of robots in the system and the i-th robot will be denoted by r_i. The robots are viewed as points in the plane. This means that multiple robots may occupy the same location in a plane. In order to describe our algorithms and the global configuration of robots during the algorithm, we shall use a fixed coordinate system[2]; in this system, the location of robot r_i is denoted by $L(r_i) = (x_i, y_i)$ and the Euclidean distance between the robots r_i and r_j, by $d(r_i, r_j)$.

A *configuration* of the n robots on the plane is denoted by the multi-set $\gamma = \{(label(r_i), L(r_i)) : 1 \leq i \leq n\}$ where $label(i)$ is the identity (or label) assigned to robot r_i. If the robots are anonymous then $label(r_i) = 1$, for all i. On the other hand, if the robots are all distinct then we assume that $label(r_i) = i$, for all $i \in \{1, 2, \ldots, n\}$. The configuration at a specific time t is denoted by $\gamma(t)$. Given a configuration γ, we denote by $L(\gamma)$ the set of points

---

[1] This models any deviations in sensory apparatus used by a robot in between two steps, specially when the robot physically moves across the plane.

[2] The robots themselves are not aware of this global coordinate system.

occupied by robots in the configuration γ, i.e. $L(\gamma) = \{(x, y) : \exists l, (l, x, y) \in \gamma\}$. Recall that multiple robots may occupy the same location, thus the cardinality of $L(\gamma)$ could be strictly less than n.

Given any set of points on the plane, the smallest enclosing circle SEC of the points is the circle of minimum diameter such that every point is either on or in the interior of this circle. For a configuration γ, we define $\mathrm{SEC}(\gamma)$ to be the smallest enclosing circle for the set of points in $L(\gamma)$.

2.2 Patterns, Series and Formation

A *pattern P* is represented by a set of distinct points (x_1, y_1), (x_2, y_2), ..., (x_s, y_s) in the two-dimensional Euclidean plane. A pattern P_i is said to be *isomorphic* to a pattern P_j if P_j can be obtained by a combination of translation, rotation and scaling of pattern P_i. Two patterns that are not isomorphic to each other are said to be *distinct*. The *size* of a pattern P_i is its cardinality, and is denoted by $n_i = \mathrm{size}(P_i)$. We define some special patterns below:

- `POINT`: The pattern consisting of a single point.
- `TWO-POINTS`: The only possible pattern consisting of exactly two points.
- `POLYGON(k)` : For any $k \geq 3$, this is the pattern consisting of points $p_1, p_2, \ldots, p_k$ that are vertices of a regular convex polygon of k sides.

For any pattern P, with $\mathrm{size}(P) > 1$, the *bounding circle* of P is the smallest circle that encloses all points in P. We define the *symmetricity* $\rho(P)$ to be the largest integer $q \geq 1$ such that the area inside the bounding circle of P can be partitioned into exactly q equiangular sectors (each making an angle of $2\pi/q$ at the center), and containing the sets of points $s_1, s_2, \ldots, s_q$ respectively, where each s_{i+1} is a rotation of s_i with respect to the center of the circle. We define $\rho(\texttt{POINT})$ to be infinity.

We say that a system of robots R have *formed* the pattern P, if the current configuration γ is such that $L(\gamma)$ is isomorphic to P. Two configurations γ_i and γ_j are said to be *analogous* if $L(\gamma_i)$ is isomorphic to $L(\gamma_j)$ (i.e. the two configurations form the same pattern P).

We are interested in ordered series of patterns that can be formed by a system of robots. We consider two types of pattern series: A *linear pattern series L* is any (possibly infinite) ordered sequence $S = \langle P_1, P_2, \ldots \rangle$ of patterns, where P_i and P_j are non-isomorphic whenever $i \neq j$. A *cyclically ordered series* is any periodic sequence $S^\infty = \langle P_1, P_2, \ldots, P_m \rangle^\infty$ where $S = \langle P_1, P_2, \ldots, P_m \rangle$ is a finite linear pattern series.

A system of robots executing an algorithm $\mathcal{A}$, starting from a configuration $\gamma(t_0)$ is said to *completely form* a pattern series $S = \langle P_1, P_2, P_3 \ldots, \rangle$ if during any possible execution of $\mathcal{A}$, there exists time instances $t_1, t_2, t_3 \ldots$, where $t_0 < t_j < t_{j+1}$ such that for all $1 \leq j \leq |S|$, $L(\gamma(t_j))$ is isomorphic to P_j. A pattern series S is *completely formable* if there exists an algorithm $\mathcal{A}$ for a system of $k > 0$ robots such that starting from any initial configuration, the system of k robots completely forms S. It is immediate that no *finite* linear pattern series is completely formable by a terminating algorithm:

Lemma 1 *Given any finite linear series of patterns* $S = \langle P_1, P_2, P_3 \ldots, P_m \rangle$, *where* $m \geq 2$, *no deterministic algorithm* $\mathcal{A}$ *that terminates in finite time can completely form S from all possible starting configurations.*

Proof In any deterministic algorithm for oblivious robots, actions taken by the robots depend only on the current configuration. By contradiction, let $\mathcal{A}$ be a deterministic protocol that always completely forms the finite linear series of patterns $S = \langle P_1, P_2, P_3 \ldots, P_m \rangle$, where $m \geq 2$. Consider an execution of $\mathcal{A}$ from an arbitrary initial configuration γ. Let γ_t be the configuration when the algorithm terminates. Consider now the execution of $\mathcal{A}$ when the starting configuration is γ_t, Because of obliviousness, the robots can not distinguish between the two executions; thus, this execution will immediately terminate without forming S: a contradiction.

The above lemma does not forbid the formation of trivial series consisting of a single pattern. However for longer series ($m \geq 2$), any terminating algorithm can form only a suffix of S, depending on the starting configuration. This means that, for any linear series $S = \langle P_1, P_2, P_3 \ldots, P_m \rangle$, any protocol that just forms the single pattern P_m will be deemed to be forming S. It may be however possible to construct non-terminating algorithms that form the entire given series: this is the case if S^∞ is completely formable. Thus our focus will be on determining which cyclic series are completely formable by a team of oblivious robots.

3 General Case: Anonymous Robots

3.1 Preliminaries

In this section, the robots are assumed to be anonymous and oblivious, having no other additional capabilities. Central to the anonymous case is the notion of symmetry in a configuration, which is quantified using the concept of *centred view*, and *symmetricity*, a slight modification of the notion of local view and of symmetricity introduced in [26].

Definition 1 The centred view of a robot r at a time t is defined with respect to the coordinate system $Z_{r,t}$, as the set of tuples $CV_r(t) = \{(x_i, y_i, k_i) : k_i > 0 \text{ robots are located at } (x_i, y_i)\}$.

In the above definition, the coordinate system $Z_{r,t}$ of robot r is computed with respect to the current configuration, as follows. Let γ be the current configuration of robots. If a robot r_i is not located at the centre c of the smallest enclosing circle $SEC(\gamma)$, it builds its coordinate system $Z_{r_i,t}$ by considering its own position as the origin $(0,0)$ and by considering the position of the center c as the point with coordinate $(1,0)$. On the other hand, if robot r_i is located at the centre of the smallest enclosing circle, its coordinate system $Z_{r_i,t}$ has still its own position as origin, and the robot r_j whose view $CV_{r_j}(t)$ is minimum among all the other robots is considered to be at coordinate $(1,0)$. Clearly, if all robots are located at c, then $CV_{r_i}(t) = \{(0,0,k)\}$ for all robots r_i.

According to the above definition, no information about the local coordinate system of the robots is included in the centred views[3] because such information is neither globally known nor necessarily consistent between two rounds. In the following, when no ambiguity arises, we will refer to the *centred view* simply with the term *view*. Our definition ensures that the *view* of a robot is independent of the local coordinate system used by the robot. Thus, robots that are collocated have identical views, i.e. $L(r_i) = L(r_j) \implies CV(r_i) = CV(r_j)$.

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

An obvious consequence of anonymity is that from a configuration γ consisting of anonymous robots at w distinct locations, a configuration γ' 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.

Property 1 From a configuration γ consisting of anonymous robots at w distinct locations, it is not always possible reach a configuration γ' where the robots occupy more than w distinct locations.

(a) (b) (c)
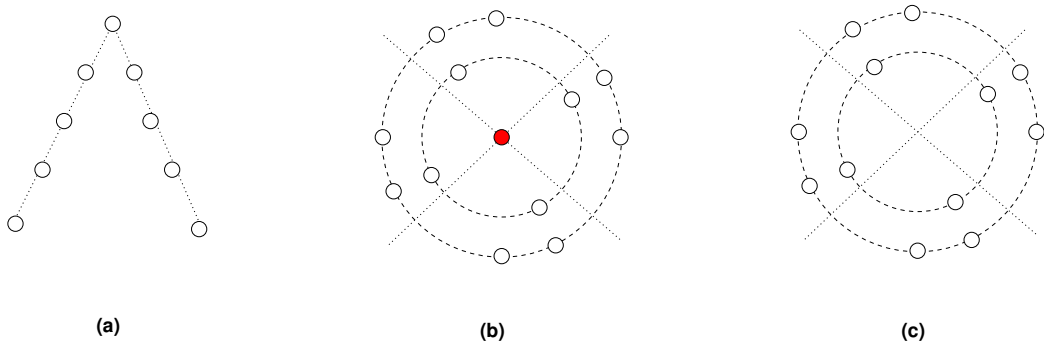
Fig. 1 A configuration γ where robots occupy distinct locations, such that: **(a)** The symmetricity $\rho(\gamma) = 1$ and each robot has a distinct view; **(b)** The symmetricity $\rho(\gamma) = 1$ and only one robot (the shaded one) has a unique view; **(c)** The symmetricity $\rho(\gamma) = 4$.

Based on our definition of views of robots, we define the symmetricity of a configuration γ. Examples of configurations with various symmetricities are shown in Figure 1.

---

[3] unlike the definition of views in [26].

Definition 2 The *symmetricity*, $\rho(\gamma)$ of a configuration γ consisting of n robots, is the largest integer q such that for each robot r, the number of robots having the same view as r (including itself) is a multiple of q; i.e. $\rho(\gamma) = \max\{q : \forall r_i \in \gamma, q \text{ divides } |\{r_j \in \gamma : CV(r_j) = CV(r_i)\}|\}$

The following properties follow from the above definition and the earlier results in [26].

Property 2 For any configuration γ containing n robots $\rho(\gamma)$ divides n.

Property 3 For any configuration γ where robots occupy **distinct locations**, the following holds: (i) If a robot is located at the center of $SEC(\gamma)$ then $\rho(\gamma) = 1$; (ii) If $\rho(\gamma) = q > 1$, then the robots are located at the vertices of regular polygons of q sides, each of which is concentric to $SEC(\gamma)$.

Lemma 2 *If the current configuration γ has symmetricity $q = \rho(\gamma)$ then, for any algorithm, there exists an execution where all subsequent configurations γ' satisfy $\rho(\gamma') = l \cdot q$, $l \geq 1$.*

Proof Any deterministic algorithm must specify the actions of a robot based on its current view. Thus, two robots having the same view would take the same action (if they are activated simultaneously). For the sake of argument, we assume an adversary that decides which robots are activated at each step. Whenever a robot r is activated, the adversary also activates all other robots that have the same view as r. The new location computed by each such robot r' with respect to its coordinate system $Z_{r'}$, would be the same as location computed by robot r with respect to its coordinate system Z_r. In other words, any two robots having identical views would continue to have identical views after moving to the new location. In this case, the symmetricity of the new configuration must be a multiple of the previous one.

3.2 Robots starting from Distinct Locations

As mentioned, Property 1 restricts the size of patterns in any formable series of patterns. To form repetitively any series S of patterns, all the patterns in S should be of the same size. Thus, we consider only patterns of size n, where n is the number of robots. Each robot starts from a distinct location and during the pattern formation, no two robots should occupy the same location (i.e. we can not allow points of multiplicity). Due to Property 1 and Lemma 2, we know the following impossibility result:

Lemma 3 *A cyclic series of distinct patterns $\langle P_1, P_2, ... P_m \rangle^\infty$ is formable only if $size(P_i) = size(P_j)$ and $\rho(P_i) = \rho(P_j)$, $\forall i, j \in \{1, 2, ... m\}$.*

We now show that exactly the above type of series are formable, by providing algorithms for forming any such series. Note that there exists exactly one pattern P such that $\rho(P) = size(P) = n$ and this is the pattern POLYGON(n). Thus, no non-trivial series may be formed starting from any configuration having symmetricity exactly equal to n. In the following, we will consider configurations with $\rho(\gamma) < n$. The formation algorithm for forming the type of series mentioned in the lemma, 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 in the algorithms, we define these special configurations.

Definition 3 (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 $\mathcal{C}_1$ containing all the robots, has diameter more than three times the diameter of the circle $\mathcal{C}_2$ that encloses all robots except those at the pivot; (ii) $\mathcal{C}_1$ and $\mathcal{C}_2$ intersect at exactly one point: the point directly opposite the pivot (called the *base-point*).

Definition 4 (SCC) A configuration γ containing n robots is called *q-symmetric-circular* or, SCC(q), $1 < q < n$, if: (i) the smallest enclosing circle $\mathcal{C}_1 = SEC(\gamma)$ has exactly q robots on its circumference forming a regular polygon; (ii) all the other robots lie on or in the interior of a smaller circle $\mathcal{C}_2$ that is concentric to $\mathcal{C}_1$ such that $Diameter(\mathcal{C}_1) \geq (5 + \sin^{-1}(\pi/q)) \cdot Diameter(\mathcal{C}_2)$; (iii) there are no robots at the center of $SEC(\gamma)$.

In both configurations, the former circle ($\mathcal{C}_1$) is called the *primary enclosure* while the latter circle ($\mathcal{C}_2$) is called the *secondary enclosure*. The ratio of the diameter of the primary enclosure over the diameter of the secondary enclosure is called the *stretch* of the configuration. For a BCC configuration, the point on the secondary enclosure directly opposite the base-point is called the *frontier-point*. Starting from an asymmetric configuration of robots, it is possible to form a BCC by moving a single robot to the pivot location (see Figure 2).

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, as we show below. Another property that can be shown is that starting 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:
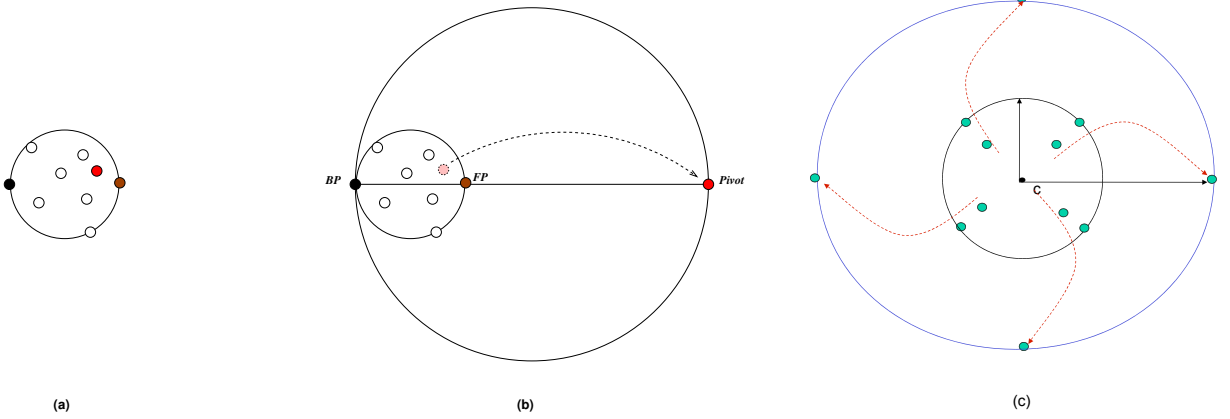
Fig. 2 (a) An asymmetric configuration of robots and the smallest enclosing circle. (b) A bi-circular configuration formed by moving one of the robots. (c) An SCC(q) configuration with $q = 4$.

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

Proof If $q = 1$ then the configuration γ is asymmetric and thus, it is possible to elect a leader among the robots (e.g. the robot with the minimum view); this elected robot can move to a pivot position to form the BCC configuration having the required stretch. In this case the lemma holds. For the rest of the proof, we assume that $1 < q < n$ and we show algorithms to reach one of the configurations (i) or (ii). First notice that, since the robots start from distinct locations and $\rho(\gamma) > 1$, no robots are located at the center c of SEC(γ). Further, all robots lie on circles concentric to SEC(γ) and the robots can be partitioned into classes (each containing exactly q robots) such that the robots in the same class have identical views and robots in distinct classes have distinct views (see Property 3). Consider the class R of robots (called *leaders*) that have the minimum view among all robots, according to some predefined ordering on the views. Note that there exist robots that do not belong to R (If all robots were in R, then the symmetricity would be $\rho(\gamma) = n$: a contradiction). Further, we can choose the class R of leader robots in such a way that some of the remaining robots are located on SEC(γ). The algorithm instructs only the robots in R to move. Each robot in R computes its next destination on the circle $\mathcal{C}_1$ whose diameter is k times the diameter D of the smallest circle ($\mathcal{C}_2$) enclosing the remaining robots (Note that $\mathcal{C}_1$ is identical to SEC(γ)). If robot $r \in R$ is activated by the scheduler it moves to the farthest point on $\mathcal{C}_1$ that lies on the straight line joining c to r.

First consider the case when exactly one robot $r \in R$ is activated by the scheduler and it moves to $\mathcal{C}_1$. In this case, the resulting configuration would be BCC with stretch $k' = (k+1)/2$ as in the lemma (Note that $k' > 3$ as required in the definition of BCC). Now, let us consider the case when exactly $q' > 1$ robots from the set R, are activated by the scheduler (where q' is a factor of q as in the lemma) and these robots form a regular polygon of q' sides centred at c. When all these robots move to the computed circle $\mathcal{C}_1$, they would continue to form a regular polygon of q' sides. The resulting configuration is SCC(q') and the lemma holds in this case too. We now consider the case when either the number of robots that move to $\mathcal{C}_1$ is co-prime with q or, these robots do not form a regular polygon. In both these cases the symmetry is broken, i.e. the new configuration has symmetricity one. This new configuration γ^* is neither BCC nor SCC ; However we show that the robots can easily revert to a bi-circular configuration (BCC). First, notice that configuration γ^* can be identified by each robot as a failed attempt to form an SCC configuration. Let us call as *failed* robots, the robots that moved to the circle $\mathcal{C}_1$. Due to the following facts, each of the failed robots is twice as far from any other robot, compared to the robots that did not move, i.e. those that are on or inside $\mathcal{C}_2$.

1. The distance between any two failed robots is at least

$$k \times D \times \sin(\pi/q) \geq (2 + 5\sin(\pi/q)) \times D > 2D$$

2. The distance between a failed robot and a robot on or inside $\mathcal{C}_2$, is at least $(k-1)D/2 > 2D$.

Thus, it is possible to identify the *failed* robots by looking at configuration γ^*. Moreover, there exists a unique ordering on the failed robots (since configuration γ^* has symmetricity one). According to this ordering, each of

the failed robots, one after the other, returns to some unoccupied location in the interior of $\mathcal{C}_2$; By choosing the ordering, the algorithm can ensure that each intermediate configuration has symmetricity one. When only one of the failed robots remains outside the circle $\mathcal{C}_2$, then the resulting configuration is BCC with stretch k' as in the lemma. Thus we have arrived at the required configuration (ii) and the lemma holds.

The only remaining case we need to consider is when $q* < q$ robots from the set R move to the circle $\mathcal{C}_1$, but the resulting configuration γ^* has symmetricity q', where q' is a factor of both q and $q*$. In this case, again it is possible to identify the configuration as a failed attempt at forming SCC(q), and the failed robots can be identified, as before. Moreover, the failed robots can be classified into distinct classes having q' robots each. In the next step of the algorithm, one of these classes is chosen and each robot in that class is instructed to move to some point inside $\mathcal{C}_2$ that lies on the line segment from current location of the robot to the center of $\mathcal{C}_1$. This process is repeated until there is exactly one class of q' robots that are located on the outer circle $\mathcal{C}_1$. The resulting configuration is SCC(q') with the appropriate stretch as required by the lemma. Thus in a! ll cases, we arrive at a configuration of type (i) or (ii) and the lemma holds.

Lemma 5 *Starting from a configuration of type SCC(q), q > 1, with n robots occupying distinct locations we can form any pattern P such that size(P)= n and the symmetricity $\rho(P) = q \cdot a$, for some integer $a \geq 1$.*

Proof We begin forming the pattern P within the secondary enclosure ($\mathcal{C}_2$) of the SCC configuration, while the robots on $\mathcal{C}_1$ remain stationary. The points in P are mapped to locations on or in the interior of $\mathcal{C}_2$, in such a way that the circle $\mathcal{C}_2$ corresponds with the SEC (P). The robots are assigned to these locations. The circle $\mathcal{C}_2$ is partitioned into q equiangular sections and the assignment of robots within each sector is unique. There are at least q robots on $\mathcal{C}_2$ which do not need to move as they correspond to q points in SEC (P). Thus, the configuration SCC(q) is maintained during the movement of these robots. As a final step, the robots located on $\mathcal{C}_1$ need to move inside $\mathcal{C}_2$ to complete the pattern. During this step the SCC configuration may be broken (since all the robots on $\mathcal{C}_1$ may not be activated simultaneously). However, the robots can still identify the resulting configuration as pattern P with at most $q-1$ points missing and an equal number of additional robots. These additional robots can be distinguished as they are located a large distance apart from each-other (at least twice the distance between any two robots that form the pattern). Moreover these robots can be uniquely mapped to the missing points to complete the pattern P.

Lemma 6 *The following properties hold for BCC configurations: (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 P of size n can be formed.*

Proof (i) In a bi-circular configuration, there is a unique diameter of the primary enclosure (the diameter containing the pivot). We can define the positive x-axis as the line containing this diameter, in the direction from the base-point to the pivot. Due to the agreement on left-right orientation, we can now define the line perpendicular to the x-axis in the left direction, to be the positive y-axis. Thus, the base-point represents the origin and we have a unique coordinate system, where the length of the diameter of the secondary enclosure is taken as the unit distance.

(ii) Due to the agreement on a unique coordinate system, the robots can be ordered lexicographically according to their coordinates, as r_1, r_2, ..., r_n where r_1 is the robot at BP and r_n is the robot at the pivot position. If r_{n-1} is not at FP, it moves to the FP position (note that there can not be any other robot at this location). While r_1, r_{n-1} and r_n remain in these positions, the BCC configuration is maintained. The other robots compute their destination according to the following algorithm. The points in the pattern P are mapped to locations in the bi-circular configuration such that the bounding circle of pattern P coincides with the secondary enclosure $\mathcal{C}_2$ of the configuration and the base-point coincides with the lexicographically smallest point p_i on the bounding circle of P. Notice that this mapping is unique and every robot can obtain the same set $\Gamma(P)$ of locations on or inside $\mathcal{C}_2$ that correspond to points in the pattern P. Let us assume first that at least three points of P lie on the bounding circle BC(P). The robot r_{n-1} is assigned to the location lexicographically largest among those on the bounding circle BC(P) of P. If there are more than three points on BC(P), the robot r_n is assigned to the location that is second-largest (lexicographically) among these; Otherwise the robot r_n is assigned to the location lexicographically largest among those in the interior of BC(P). Now, the remaining robots r_2 to r_{n-2} are assigned to the remaining locations in $\Gamma(P)$, according to the lexicographic order. The robots r_2 to r_{n-2} move to their respective destination in such a way as to maintain the relative order among them based on their positions. Now the robot r_{n-1} moves to its assigned location (which lies on SEC2). After this move the BCC configuration is still maintained since there are at least three robots on the SEC2 at distinct locations. At a final step, the robot r_n can move to its destination to complete the pattern P. We now consider the case when only two points of P are on BC(P), which implies

that these two points are on the opposite ends of a diameter of BC(P). This case is simpler since the robots r_1 and r_{n-1} as defined above, are already on the required locations (after the first step). Thus the robots r_2 to r_{n-2} move to their respective destinations without disturbing the BCC configuration. As a final step, robot r_n moves to its assigned location to complete the pattern P.

Based on the above results, we now describe the algorithm for forming a cyclic series of distinct patterns $S^\infty = \langle P_1, P_2, \ldots P_m \rangle^\infty$, with n robots starting from a configuration γ, under the condition that $\forall i$ size$(P_i) = n$ and $\rho(P_i) = q = \rho(\gamma)$.

We only need to consider the case $q < n$ and $n > 2$, since for $q = n$ or for $n = 2$, there is a single possible pattern: the regular n-gon and the robots already form that. The scenarios $(q = 2, n = 3)$ is impossible due to Property 2. The scenario $(q = 1, n = 3)$ is easily solved because every transformation from one pattern to another requires only one robot to move and since the symmetricity is one, the algorithm can select exactly one robot to move in each configuration. Thus, we now focus on the case when $n \geq 4$ and $q < n$.

Let F be a function that maps each pattern $P_i \in S$ to a real number $k_i = F(P_i)$ that satisfies the condition of Lemma 5. To signal the formation of pattern P_i, one of the following configurations is unambiguously used: either 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 P_i is being formed. By Lemmas 6 and 5, the robots can then form pattern P_i. 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 $F(P_{i+1})$ for the next pattern P_{i+1} in the series. Using this technique, the robots can move from one pattern to the next, and thus they can form the required series of patterns.

From Lemma 3 and from the algorithm described above, we obtain the following characterization.

Theorem 1 *A set of n anonymous robots starting from distinct locations in an arbitrary configuration γ, can form a cyclic series of distinct patterns $\langle P_1, P_2, \ldots, P_m \rangle^\infty$, each pattern of size n, if and only if $\forall i, j \in \{1, 2, \ldots m\}$ $\rho(P_i) = \rho(P_j) = a \cdot \rho(\gamma)$ for some integer $a \geq 1$.*

3.3 Special Case: Agreement on Directions

In this section, as a special case we consider robots that agree on directions (i.e. they have common notion of North, South, East and West). This is the case if each robot is provided with a consistent compass, for example. Thus, the robots in this section, besides agreeing on a common clockwise direction, also agree on the directions of a fixed coordinate system. We say such robots have common *sense of direction*.

In this case, as long as the robots occupy distinct locations, there exist a total order on the robots. However whenever two robots gather at the same point, we lose the order relationship between these two robots. For robots with this additional capability, we have the following results.

Lemma 7 *With n anonymous robots having common sense of direction, any single pattern P of size $n' \leq n$ can be formed, if the robots start from distinct locations.*

Proof As mentioned before, there is a total order on the n robots, say $r_1, r_2, \ldots, r_n$ based on their locations (e.g. ordered left to right and then bottom to top). Suppose the points $p_1, p_2, \ldots p_{n'} \in P$ are also ordered similarly. Thus, the location of r_1 and r_2 can be matched to points p_1 and p_2 and all other other robots r_3 to r_n simply move to the points p_3 to $p_{n'}$, in this order (i.e. robots r_{n+1} to r_n all move to the same location $p_{n'}$). During these movements, the ordering of the robots is preserved, thus every robot can unambiguously determine the location where it should move to, to form the pattern P.

In case all robots do not start from distinct locations, it is easy to see that any pattern P of size n' is formable whenever at least n' out of the n robots are initially in mutually distinct locations. We now show which series of patterns are formable starting from any arbitrary configuration.

Theorem 2 *With n anonymous robots having a common sense of direction and occupying w distinct locations, we can form any cyclic series of distinct patterns $S^\infty \langle P_1, P_2, \ldots P_m \rangle$, if and only if $\forall i \in \{1, 2, \ldots m-1\}$ size(P_i) size$(P_{i+1}) \leq w$.*

Proof The "only if" part follows from Property 1. We only need to show how to form the given series. Let $r_1, r_2, \ldots, r_w$ be the order among robots based on their locations. Note that robots located at the same place share the same identity. We use the technique of fixed ratios, where robots r_1, r_2, and r_w form a specific ratio to signal the

formation of a pattern P_i. We employ a function F that associates each pattern P_i to some real number $F(P_i) > 1$ and to form the pattern P_i we maintain a configuration where the ratio of distances $d(r_1, r_l)/d(r_1, r_2) = F(P_i)$ where r_l denotes the last robot in the current configuration. Note that such a configuration, called configuration $FormRatio(F(P_i))$, can be formed in one step, by movement of one or more robots from the location $L(r_w)$ to the new location satisfying the ratio constraint. Each robot can determine which pattern is being formed by computing this ratio. Once the configuration $FormRatio(F(P_i))$ is formed the pattern P_i can be formed easily using the same techniques as in the proof above.

4 Visibly Distinct Robots

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 tuples of the form (j, x_j, y_j) where $1 \leq j \leq n$, $j \neq i$ and (x_j, y_j) is the location of the j-th robot, with respect to the local coordinate system of robot r_i. The *view* of a robot contains the information observed by the robot, including the identities and locations of robots. Thus, the view of each robot is unique in this case, and even in absence of agreement on directions, the symmetry among the robots can be broken by the use of distinct labels. In other words, there can be 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.

As we showed in section 3.3, having an order on the robots allows us to form any pattern of size $n' \leq n$. However, for labelled robots, the order among is preserved even if the robots are not in distinct location. So, the assumption that the robots start from distinct locations, is no longer necessary.

Lemma 8 *With n robots having visibly distinct identities, starting from arbitrary locations, any single pattern P of size $n' \leq n$ points can be formed.*

Proof This is achieved by Algorithm 1. The case for $P = \texttt{POINT}$ is trivial; all robots except r_1 simply move to the location of r_1. Let us consider patterns where $size(P) > 1$. If robots r_1 and r_2 are at the same location, then r_2 will be the first robot to move. Once these two robots are in distinct locations, they remain there until the end of the algorithm. Taking $L(r_1)$ as point $p_1 \in P$ and $L(r_2)$ as point $p_2 \in P$, the locations corresponding to all the other points in the pattern can be uniquely determined with respect to these two fixed points. Whenever the robot r_i, $i > 2$ becomes active, it can determine the correct location, corresponding to the point p_i (or $p_{n'}$ if $i > n'$) and move there. Thus, after every robot has executed at least one computation cycle, the pattern P would be formed.

Algorithm 1: Form-Pattern

/* Algorithm for single pattern P */
INPUT: $P = (p_1, p_2, \ldots, p_{n'})$, $ID = i$
begin
 if $P = \texttt{POINT}$ *and* $i > 1$ **then**
  | Move to the location of robot r_1
 else
  if $i = 2$ *and* r_1 *is colocated with* r_2 **then**
   ⌊ Move to a location distinct from robot r_1 ;
  if $i > 2$ *and* r_1 *is not colocated with* r_2 **then**
   | $i \leftarrow \text{Min}(i, size(P))$;
   | Consider the coordinate system Z such that, $L_Z(r_1) = p_1$ and $L_Z(r_2) = p_2$;
   ⌊ Move to the location corresponding to $p_i \in P$;

When there is only one robot, the only pattern that can be formed is obviously $\texttt{POINT}$. With $n = 2$ robots, only two patterns can be formed: $\texttt{POINT}$ and $\texttt{TWO-POINTS}$ and it is easy to form the cyclic series $(\texttt{POINT}, \texttt{TWO-POINTS})^\infty$, by movement of a single robot (say r_2). There are no other possible series of patterns for $n = 2$. The more interesting cases occur when there are at least three robots (i.e., $n \geq 3$), in this case any series of distinct patterns $S^\infty = \langle P_1, P_2, \ldots P_m \rangle^\infty$ can be formed, with the only restriction that each pattern P_i has at most n points. We describe below an algorithm that achieves this.

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 algorithm serving as fixed points of reference for the other robots. If the robots r_1 and r_2 are initially collocated

---

Algorithm 2: FormSeries

```
/* Algorithm for forming a series S^∞ with visibly distinct robots */
INPUT: S = ⟨P₁, P₂, ..., Pₘ⟩, ID = i
```

begin

    case $ID = 2$

        if *all robots except r_n are colocated* **then**   `/* Pre-Point configuration */`

             Do nothing;

        else if *r_1 is colocated with r_2* **then**

             Move to a location distinct from robot r_1 ;

        else if *all robots except r_2 and r_n are colocated* **then**

             if $\exists P_j \in S : P_j = \texttt{POINT}$ **and** $d(r_1, r_2)/d(r_1, r_n)$ *equals* $F(P_j)$ **then**

                  Move to the location of robot r_1;

    case $2 < ID < n$

        if *r_1, r_2 and r_n are at distinct locations* **then**

             $w \leftarrow d(r_1, r_2)/d(r_1, r_n)$;

             if $\exists j \in \{1, 2, \ldots m\} : F(P_j) = w$ **then**

                  Execute **Form-Pattern** (P_j, ID) ;

    case $ID = n$

        if *all robots except r_n are colocated* **then**   `/* Pre-Point configuration */`

             Move to the same location as the other robots;

        else if *all robots except r_2 are colocated* **then**   `/* Post-Point configuration */`

             if $\exists j \in \{1, 2, \ldots, m\} : P_j = \texttt{POINT}$ **then**

                  Move to a location such that the ratio $d(r_1, r_2)/d(r_1, r_n) = F(P_{(j \mod m)+1})$;

             else

                  Move to a location such that $d(r_1, r_2)/d(r_1, r_n) = F(P_1)$;

        else if *r_1 is not colocated with r_2* **then**

             $t \leftarrow d(r_1, r_2)/d(r_1, r_n)$;

             if $\exists j \in \{1, 2, \ldots, m\} : F(P_j) = t$ **then**

                  if *all other robots are in correct location to form pattern P_j* **then**

                      Move to the appropriate location to form P_j

             else if *Current configuration corresponds to pattern $P_j \in S$* **then**

                  Move to a location such that $d(r_1, r_2)/d(r_1, r_n) = F(P_{(j \mod m)+1})$;

             else

                  `// Start to form pattern P₁`

                  Move to a location such that $d(r_1, r_2)/d(r_1, r_n) = F(P_1)$;

---

then robot r_2 must move away from r_1 during the first step of the algorithm. As before, we use some known function F to map each pattern P_j to a distinct real number $w_j = F(P_j)$, $w_j \in (1, \infty)$. Before forming pattern P_j, robot r_n moves to a location between r_1 and r_2 such that the ratio of distances $\frac{dist(r_1, r_2)}{dist(r_1, r_n)}$ 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 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 P_l in the series , or is an intermediate configuration which signals the formation of P_l (i.e. where r_1, r_2, and r_n maintain a ratio of $w_j = F(P_l)$). The function F must be chosen in such a way that the ratio $\frac{dist(r_1, r_2)}{dist(r_1, r_n)}$ 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.

In order to include POINT in the series of patterns formed, we need to make some modifications to our scheme. Notice that the penultimate configuration just before forming POINT must necessarily have all robots except one at the same location. The same holds for the configuration immediately after forming POINT . In these two situations, we can not use the ratio of distances to signal the formation of a pattern. So, we use the following convention. When forming the POINT pattern, the robot r_n is the last robot to join and when breaking away from the POINT pattern, robot r_2 is the first robot to move. We define the following special configurations:

(i) *PrePoint*: Robots r_1 to r_{n-1} in the same location and r_n in a different location.

(ii) *PostPoint*: All robots except r_2 in the same location and robot r_2 in a different location.

(iii) *TwoPoints*: Robots r_2 to r_n in the same location and r_1 in a different location.

(iv) *FormRatio(w)*: This is the set of configurations where robots r_1, r_2, r_n lie on distinct locations on the same line such that the ratio of distances $dist(r_1, r_2)/dist(r_1, r_n) = w$. (The other robots may be located anywhere.)

The algorithm for forming a given series $S = \langle P_1, P_2, \ldots, P_m \rangle^\infty$ is called *Form-Series* and is sketched below and presented in Algorithm 2. The algorithm recognizes the above configurations as special configurations. In particular, the *PrePoint* configuration is recognized as the configuration immediately before forming POINT . In this case, the algorithm proceeds to form POINT. Similarly, the algorithm recognizes the special configuration *PostPoint* as the configuration immediately after POINT formation. In this case, the algorithm proceeds to form the next pattern $P_{(i \mod m)+1}$ if $P_i = $ POINT $\in S$. On the other hand, if POINT does not belong to the series S then the algorithm proceeds to form the first pattern P_1 in S. Notice that the configurations *PrePoint*, *PostPoint* and *TwoPoints* are analogous; however only the configuration *TwoPoints* is considered to correspond to the pattern TWO-POINTS. Whenever the current configuration corresponds to some pattern $P_i \in S$, the algorithm proceeds to form the next pattern $P_{(i \mod m)+1}$ by first forming a configuration of type *FormRatio(t)* where $t = F(P_{(i \mod m)+1})$.

Lemma 9 *Given a cyclic series of distinct patterns $S^\infty = \langle P_1, P_2, \ldots, P_m \rangle^\infty$ Algorithm* Form-Series *forms S^∞ if $size(P_i) = n_i \leq n$, for all $i = 1, 2, \ldots, m$.*

Proof During the algorithm, the robot system goes through a series of configurations $\gamma_0, \gamma_1, \ldots$, where for all $i > 1$, either γ_i corresponds to some pattern $P_i \in S \bigcup \{$POINT$\}$ or γ_i is one of the following configurations: *PrePoint*, *PostPoint*, *TwoPoints*, or any configuration of the type *FormRatio(w)* for $w = F(P_i)$, $1 \leq i \leq m$. If the initial configuration is none of those described above, then the system reaches such a configuration in just one or two steps with a single movement of robot r_2 or r_n or both (one after the other). If all robots are collocated, r_2 moves away to form configuration *PostPoint*. If r_1 and r_2 are initially collocated, then r_2 first moves out and followed by robot r_n to form configuration *FormRatio*$(F(P_1))$. Otherwise r_1 and r_2 are initially separated and in this case only r_n needs to move to form *FormRatio*$(F(P_1))$.

We now need to show that from a configuration γ_i that is one of the types discussed above, the algorithm eventually forms the complete series of patterns. It is easy to see that if γ_i is of type FormRatio$(F(P_j))$, $1 \leq j \leq m$, then the algorithm forms the pattern P_j. On the other hand if γ_i corresponds to some pattern P_j in S, $P_j \neq$ POINT then γ_{i+1} is a configuration of type *FormRatio(w)* where $w = F(P_{(j \mod m)+1})$ which implies that eventually pattern $P_{(j \mod m)+1}$ will be formed. This implies that the algorithm forms the complete series in this case. Note that if γ_i is *TwoPoints*, it is considered to correspond to the pattern TWO-POINTS and the thus the above argument holds in that case too. This leaves us with only the special cases when γ_i is either *PrePoint*, *PostPoint* or corresponds to POINT . If γ_i is *PrePoint* then γ_{i+1} would correspond to POINT and γ_{i+2} would be *PostPoint*. This further implies that γ_{i+3} would be the configuration *FormRatio(t)* where $t = F(P_j)$ if $P_{j-1} = $ POINT and $t = 1$ otherwise. This combined with the previous argument implies that the algorithm forms the series S even in these special cases. □

The Algorithm *Form-Series* forms any cyclic series $\langle P_1, P_2, \ldots, P_m \rangle^\infty$ for $n \geq 3$ robots. For $n = 2$ robots, it is easy to form the only possible series $\langle$POINT, TWO-POINTS$\rangle^\infty$, by movement of a single robot (as mentioned before). For $n = 1$, no non-trivial series are possible. To summarize, we have the following result:

Theorem 3 *With $n > 1$ robots having distinct visible identities, any cyclic series of distinct patterns $\langle P_1, P_2, \ldots, P_m \rangle^\infty$ is formable if and only if for all i, $1 \leq i \leq m$, $size(P_i) \leq n$.*

5 Distinct Robots with Invisible Identities

In this section, we consider robots that have distinct identities but 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 identifier.

Consider first the case when there are at least four robots (i.e. $n \geq 4$). The BCC configuration, defined for the anonymous case, is used here as well to signal the formation of specific patterns in a series. However since dense points are allowed in this case, there may be more than one robots at the pivot and at the base-point of the bi-circular configuration.

Lemma 10 *From any arbitrary configuration γ with $|L(\gamma)| \geq 3$, 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 P starting from a bi-circular configuration of stretch k_i is as follows: As 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} would move to the base-point and the frontier-point respectively. Now, these three robots remain in their location while the other robots move to the required positions for forming pattern P. The positions are assigned in the following manner. The points in the pattern P are mapped to locations in the bi-circular configuration such that the bounding circle of pattern P coincides with the secondary enclosure of the configuration and the base-point coincides with the lexicographically smallest point p_i on the bounding circle of P, i.e., $p_i \in BC(P)$ and $p_i \leq p_j$, for any $p_j \in BC(P)$. Notice that this mapping is unique. Let $\Gamma(P)$ be the unique mapping obtain by each robot (i.e., the locations that correspond to points in the pattern P). The elements of $\Gamma(P)$ 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 $\mathcal{C}_2$ precede those that are located in the interior of $\mathcal{C}_2$. For $1 \leq i \leq size(P)$ robot r_i is assigned the ith location in $\Gamma(P)$ and for $size(P) < j \leq n$ robot r_j is assigned to the last location in $\Gamma(P)$.

The algorithm that implements the above strategy is called algorithm *Form-Series-2* (See Algorithm 3). During the formation of a pattern P_i, the algorithm ensures that the BCC configuration is maintained by keeping the robots r_1, r_{n-1} and r_n stationary at the BP, FP and pivot positions, respectively. When all other robots have moved to their assigned location, robot r_{n-1} finally moves to its assigned location. If r_{n-1} is assigned a location inside the secondary enclosure $\mathcal{C}_2$, then all the points on the circle $BC(P_i)$ are already occupied by robots; So, the circle $\mathcal{C}_2$ (and thus the BCC) is preserved after the move of r_{n-1}. Otherwise, if r_{n-1} is assigned another position on $\mathcal{C}_2$, then there must be already two other robots on the circle $\mathcal{C}_2$ (since $n \geq 4$) and thus, the BCC is preserved after the move of r_{n-1} (Since any three points uniquely determine the only circle passing through them). Thus, after the move of r_{n-1}, the BCC configuration of appropriate stretch is still maintained. Hence robot r_n can unambiguously move to the required position to complete the pattern.

When the BCC configuration is initially formed, the robots r_1 and r_{n-1} may not be present at the base-point(BP) and frontier-point(FP) respectively. In that case, these robots move to these locations during the first step when they are activated. Only after this the robots that were originally at BP and FP are allowed to move to other locations. In case r_1 is at FP and r_{n-1} is at BP, then none of these robots can move without breaking the BCC configuration. In this special case, the robots r_1 and r_{n-1} simply reverse their roles (i.e. r_{n-1} performs the algorithm for r_1 and vice versa).

For the above algorithm, the special configuration *PrePoint* is defined similarly as before, while the configuration *PostPoint* is defined as the configuration where r_1 is alone and all other robots are together. The configuration *TwoPoints* that corresponds to TWO-POINTS is defined as the configuration where r_1 and r_n are together and all other robots are collocated at a separate location. Since $n \geq 4$, these three special configurations can be distinguished from each-other by both robot r_1 and robot r_n (which are the only two robots that make the decisive move for changing from one pattern to the next).

We now introduce a few lemmas to show that Algorithm Form-Series-2 is correct.

Lemma 11 *Given a series of patterns $S = \langle P_1, P_2, \ldots, P_m \rangle$ as input, Algorithm* Form-Series-2 *starting from any arbitrary configuration, reaches a configuration γ, such that $L(\gamma)$ is isomorphic to some $P_i \in S$*

Proof During any execution of the algorithm 3, the series of configurations satisfies the condition that each configuration γ other than the initial configuration is one of the following:

(i) $L(\gamma)$ is isomorphic to $P_i \in S$, such that $size(P_i) > 2$.
(ii) γ is a bi-circular configuration with stretch $t_i = F(P_i)$.
(iii) $|L(\gamma)| = 1$ (i.e. all robots are collocated)
(iv) $|L(\gamma)| = 2$ and r_1 is alone. (The configuration after POINT formation)
(v) $|L(\gamma)| = 2$ and r_n is alone. (The configuration before POINT formation)
(vi) $|L(\gamma)| = 2$ and r_1 is collocated with only r_n. (The configuration for TWO-POINTS)

If the initial configuration is not one of the above then, during the next active cycle of robot r_n, it moves to a location such that configuration is BCC(t_1) (Note that this is possible because there are at least two robots other than r_n that are located at distinct locations). If POINT, TWO-POINTS $\notin S$, then all subsequent configurations will be of the type (i) or (ii). If POINT $\notin S$ and the initial configuration is of the type (iii) or (v), then robot r_1 moves to a location distinct from its current location, such that after the move, at least two of the robots other than r_n are at distinct locations. Starting from such a configuration (including configurations of type (iv) and (vi)) robot r_n moves to form the configuration BCC(t_1). (Note that if the configuration has robot r_i, $1 < i < n$ alone and all other robots together, then both r_1 and r_n could potentially move at the same time. However, the resulting

---

Algorithm 3: FormSeries2

/* Algorithm for forming a series S^∞; Robots with invisible identities */
INPUT: $S = \langle P_1, P_2, \ldots, P_m \rangle$, $k[j] = F(P_j), 1 \le j \le m$, $ID = i$
begin

    /* Let Conf be the current configuration of robots as viewed by r_i.   */
    /* Let loc be the current location of the robot r_i.   */
    case $i = 1$ /* Algorithm for robot r_1 */

        if *Conf is BCC(k[j]) for some $j \in [1, m]$* **then**
            if *loc $\ne$ BP* **and** *FP is unoccupied* **then**
                Move to FP;
            else if *loc $\ne$ BP* **and** *FP is occupied by at least one other robot* **then**
                Move to BP;
            else if *loc = FP* **and** *$n - 2$ other robots are occupying at least $n_j - 2$ points of P_j* **then**
                MoveToPosition($P_j, n - 1$);

        else if *all robots are collocated* **then**   /* i.e. L(Conf) $\simeq$ POINT */
            Move to an unoccupied location;
        else if *all robots except one are at loc* **and** *POINT $\notin$ S* **then**   /* maybe only r_n is alone */
            Move to an unoccupied location within current SEC;

    case $ID = n - 1$ /* Algorithm for robot r_{n-1} */
        if *Conf is BCC(k[j]) for some $j \in [1, m]$* **then**
            if *loc $\ne$ FP* **and** *loc $\ne$ BP* **then**
                Move to FP;
            else if *loc = BP* **and** *multiplicity(BP) > 1* **then**
                Move to FP;
            else if *loc = FP* **and** *$n - 2$ other robots are occupying at least $n_j - 2$ points of P_j* **then**
                MoveToPosition($P_j, n - 1$);

    case $ID = n$ /* Algorithm for robot r_n */
        if *Conf is BCC(k[j]) for some $j \in [1, m]$* **then**
            if *all other robots are occupying at least $n_j - 1$ points of P_j* **then**
                MoveToPosition(P_j, n);

        else if *all robots except one are at loc* **then**
            if *P_j = POINT for some $j \in [1, m]$* **then**
                Move to pivot to form BCC ($k[(j \mod m) + 1]$);
            else
                Move to pivot to form BCC ($k[1]$);

        else if *all other robots are at the same location $l \ne$ loc* **and** *POINT $\in$ S* **then**
            Move to the location occupied by all other robots;
        else if *L(Conf) matches P_j for $j \in [1, m]$, $P_j \ne$ POINT* **then**
            Move to pivot to form BCC ($k[(j \mod m) + 1]$);
        else
            Move to pivot to form BCC ($k[1]$);

    otherwise   /* Algorithm for all other robots, i.e. $1 < ID < n - 1$ */
        if *Conf is BCC(k[j]) for some $j \in [1, m]$* **and** *BP, FP are occupied* **then**
            if *at least $i - 1$ robots are occupying at least $\min(i - 1, n_j)$ points of P_j* **then**
                if *loc $\ne$ BP* **or** *multiplicity(BP) > 1* **then**
                    MoveToPosition(P_j, i);

---

configuration would still be $\text{BCC}(t_i)$, since r_1 is inside the current SEC.) On the other hand, if POINT $\in S$ and the current configuration is of type (iv), then robot r_n moves to form the configuration $\text{BCC}(t_i)$ where P_{i-1} =POINT. If TWO-POINTS $\in S$ and the current configuration is of type (vi), then robot r_n moves to form the configuration $\text{BCC}(t_i)$ where P_{i-1} =TWO-POINTS. Thus, starting from any configuration, we eventually reach a configuration of type $\text{BCC}(t_i)$ for some $t_i = F(P_i)$, $P_i \in S$. Once the configuration $\text{BCC}(t_i)$ is formed, each robot can correctly determine the location that it needs to occupy in order to form pattern P_i. The robots move to respective locations in order. While robots r_2 to r_{n-2} are moving to their respective locations, robots r_1, r_{n-1} and r_n occupy the base-point, frontier-point and pivot locations, ensuring that the bi-circular configuration is maintained. Note that when the robot r_{n-1} moves to its designated location, the bi-circular configuration is still maintained. This holds because the points on SEC2 are occupied earlier than the points in the interior of SEC2 and, since $n \ge 4$, there must be

---

Procedure MoveToPosition(P,i)

/* Algorithm to move robot r_i to the correct position for forming P */
INPUT: Pattern=P, ID=i
begin
  $n \leftarrow$ size(P);
  $\Gamma \leftarrow$ MAP(Conf,P);
  Sort Γ such that points on the SEC precede those inside SEC and $\Gamma[1]$ is the BP;
  if $i = k$ **and** $n = 2$ **then**   /* To form TWO-POINTS, r_n joins r_1 */
   | Move to location $\Gamma[1]$;
  else if $i \geq n$ **then**   /* robots r_n to r_n go to the final point */
   | Move to location $\Gamma[n]$;
  else
   | Move to location $\Gamma[i]$;

---

at least three robots on the SEC2 (including robots r_1 to r_{n-1}). In the final step, robot r_n moves from the pivot location to its assigned location and the pattern P_i.

Now, we show that after forming P_i, Algorithm Form-Series-2 forms the next pattern $P_{(i \mod m)+1}$, for any $i \in [1, m]$.

Lemma 12 *From a configuration γ representing $P_i \in S$, the algorithm eventually reaches a configuration γ' such that $L(\gamma')$ is isomorphic to $P_{(i \mod m)+1}$.*

Proof We only need to show that starting from the configuration representing P_i, a configuration of the type BCC(t_j), $j = (i \mod m) + 1$ is eventually formed. The rest follows from the previous Lemma.

Suppose $P_i =$ POINT, then all the robots are collocated. In this only robot r_1 is allowed to move. Once robot r_1 moves away, we have the configuration (iv) above. This is recognized as the configuration just after POINT formation. So, robot r_n moves to the pivot location to form BCC(t_j). If $P_i =$ TWO-POINTS then the configuration is of type (vi) mentioned above. In this case, robots r_1 and r_2 occupy distinct locations, so the SEC of all robots other than r_n is well-defined and can be computed by robot r_n. Thus, robot r_n can move to form BCC(t_i). In all other cases (i.e. for any other pattern), there are at least two robots other than r_n which occupy distinct locations. Thus, robot r_n can form configuration BCC(t_j) by directly moving to the corresponding pivot location.

From Lemmas 11 and 12, we have:

Theorem 4 *Algorithm Form-Series-2 executed by $n \geq 4$ robots forms any cyclic series of distinct patterns $S^\infty = \langle P_1, P_2, \ldots, P_m \rangle^\infty$, if size($P_i$)$\leq n$, for $i = 1, 2, \ldots, m$.*

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. For the case of $n = 3$, the transformations between any two patterns of size 3 is straightforward and requires the movement of a single robot, as mentioned before. 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 contrast to the scenario in Section 4, it is not possible for any single robot to distinguish between the special configurations *PrePoint*, *PostPoint* and *TwoPoints*. This difficulty can be overcome by using the technique presented recently in [5]. The idea is to allow robot r_3 to move in each of these three configurations. Recall that robot r_3 is alone in the configuration PrePoint and t! hus can distinguish this configuration from the other two configurations. In order to distinguish between the configurations PostPoint and TwoPoints, robot r_3 moves to a location forming a particular scalene triangle with respect to the other robots such the smallest of the interior angles is at the location of the robot that was earlier collocated with r_3. If the previous configuration was PostPoint, this robot would be r_1 and otherwise this robot would be r_2. Thus, the other robots can now distinguish between these two scenarios and take the appropriate action to form the next pattern in the series. With this minor modification, the algorithm from Section 4 can be used to form any series of patterns of size at most three, for $n = 3$ robots. In conclusion:

Theorem 5 *With $n > 1$ robots having distinct invisible identities, a cyclic series of distinct patterns $\langle P_1, P_2, \ldots, P_m \rangle^\infty$ is formable if and only if size(P_i)$\leq n$, $1 \leq i \leq m$.*

6 Conclusions

We studied the formation of geometric patterns by a system of oblivious mobile robots in a plane and showed that a set of oblivious, asynchronous robots may form a cyclic series of distinct patterns. Thus it is possible to encode the state of the system in the form of a pattern and "remember" this global state information even though the individual robots have no memory of the past. This implies we can build a self-stabilizing system of mobile robots that can perform a sequence of tasks in a given order. The computational power of such robot systems is an important area of investigation.

In this paper we considered three types of robots—robots having distinct and visible identities, robots having invisible identities and also robots that completely anonymous (and indistinguishable from each other). If the robots have distinct visible identities then any series of patterns can be formed provided that there are sufficiently many robots. The same result holds for robots having invisible but distinct identities. However, in the case of anonymous robots, the series of patterns that may be formed depends on the symmetricity of the initial configuration.

The power of the robots in our model comes from their unlimited vision and mobility. If the movements of the robots are bounded to a maximum distance δ that the robot can move in each step, as in the ATOM model [26], then we could modify our algorithms as follows. First the robots would use some convergence algorithm to move closer to each other, while avoiding collisions. Once all robots are within a circle of small diameter (depending on the value of δ, assuming that it is known), then robots can then execute the same algorithms as in this paper without ever having to move a distance longer than δ, i.e. all movements would be within a circle of diameter δ. A similar approach has been used recently in [28] where the robots are assumed to have limited visibility.

A challenging problem that is left open by the current investigation is how to form a sequence of patterns in an asynchronous setting. For example, in the asynchronous CORDA model studied in the literature, forming even a single pattern seems difficult unless either the robots share a common coordinate system [16] or the points of the pattern are visibly marked by some external agent [17].

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006.
2. H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transaction on Robotics and Automation*, 15(5):818–828, 1999.
3. H. Ando, I. Suzuki, and M. Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *IEEE Symposium on Intelligent Control*, pages 453–460, 1995.
4. Y. Asahiro, S. Fujita, I. Suzuki, and M. Yamashita. A self-stabilizing marching algorithm for a group of oblivious robots. In *12th International Conference on Principles of Distributed Systems (OPODIS)*, pages 125–144, 2008.
5. Z. Bouzid and A. Lamani. Robot networks with homonyms: The case of patterns formation. *arXiv Report*, 1105.5817, 2011.
6. F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Princeton University Press, 2009.
7. I. Chatzigiannakis, M. Markou, and S. E. Nikoletseas. Distributed circle formation for anonymous oblivious robots. In *3rd Workshop on Efficient and Experimental Algorithms (WEA)*, pages 159–174, 2004.
8. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *SIAM Journal on Computing*, 41(4):829–879, 2012.
9. R. Cohen and D. Peleg. Local spreading algorithms for autonomous robot systems. *Theoretical Computer Science*, 399(1-2):71–82, 2008.
10. X. Défago and S. Souissi. Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. *Theoretical Computer Science*, 396(1-3):97–112, 2008.
11. Y. Dieudonné, S. Dolev, F. Petit, and M. Segal. Deaf, dumb, and chatting asynchronous robots. In *13th Int. Conf. on Principles of Distributed Systems (OPODIS)*, pages 71–85, 2009.
12. Y. Dieudonné, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. *ACM Transactions on Autonomous and Adaptive Systems*, 3(4), 2008.
13. A. Efrima and D. Peleg. Distributed models and algorithms for mobile robot systems. In *33nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 70–87, 2007.
14. P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots*. Morgan & Claypool, 2012.
15. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1-3):147 – 168, 2005.
16. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, 407(1-3):412 – 447, 2008.
17. N. Fujinaga, H. Ono, S. Kijima, and M. Yamashita. Pattern formation through optimum matching by oblivious corda robots. In *14th International Conference on Principles of Distributed Systems (OPODIS)*, pages 1–15, 2010.
18. N. Fujinaga, Y. Yamauchi, S. Kijima, and M. Yamashita. Asynchronous pattern formation by anonymous oblivious mobile robots. In *Proc. 26th International Symposium on Distributed Computing (DISC)*, pages 312–325, 2012.
19. V. Gervasi and G. Prencipe. Coordination without communication: the case of the flocking problem. *Discrete Applied Mathematics*, 144(3):324–344, 2004.
20. N. Gordon, I. A. Wagner, and A. M. Bruckstein. Discrete bee dance algorithms for pattern formation on a grid. In *IEEE/WIC International Conference on Intelligent Agent Technology (IAT)*, pages 545–549, 2003.

21. T. Izumi, T. Izumi, S. Kamei, and F. Ooshita. Feasibility of polynomial-time randomized gathering for oblivious mobile robots. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):716–723, 2013.
22. T. Izumi, Y. Katayama, N. Inuzuka, and K. Wada. Gathering autonomous mobile robots with dynamic compasses: An optimal result. In *21st International Symposium on Distributed Computing (DISC)*, pages 298–312, 2007.
23. J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. Parts 1 and 2. *SIAM Journal on Control and Optimization*, 46(6):2120–2147, 2007.
24. S. Souissi, T. Izumi, and K. Wada. Oracle-based flocking of mobile robots in crash-recovery model. *Theoretical Computer Science*, 412(33):4350–4360, 2011.
25. K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotics Systems*, 13:127–139, 1996.
26. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.
27. M. Yamashita and I. Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theoretical Computer Science*, 411(26-28):2433 – 2453, 2010.
28. Y. Yamauchi and M. Yamashita. Pattern formation by mobile robots with limited visibility. In *20th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 201–212, 2013.