On the Computational Power of Oblivious Robots: Forming a Series of Geometric Patterns*

Shantanu Das
LIF, Aix-Marseille University
13453 Marseille
France
shantanu.das@acm.org

Paola Flocchini
School of Info. Tech. & Engg.
University of Ottawa
Canada
flocchin@site.uottawa.ca

Nicola Santoro
School of Computer Science
Carleton University
Canada
santoro@scs.carleton.ca

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

ABSTRACT

We study the computational power of a distributed system consisting of simple autonomous robots moving on the plane. The robots are endowed with visual perception but do not have any means of explicit communication with each other, and have no memory of the past. In the extensive literature it has been shown how such simple robots can form a *single* geometric pattern (e.g., a line, a circle, etc), however arbitrary, in spite of their obliviousness. This brings to the front the natural research question: what are the real computational limits imposed by the robots being oblivious? In particular, since obliviousness limits what can be remembered, under what conditions can oblivious robots form a *series* of geometric patterns? Notice that a series of patterns would create some form of memory in an otherwise memory-less system. In this paper we examine and answer this question showing that, under particular conditions, oblivious robot systems can indeed form series of geometric patterns starting from any arbitrary configuration. More precisely, we study the series of patterns that can be formed by robot systems under various restrictions such as anonymity, asynchrony and lack of common orientation. These results are the first strong indication that oblivious solutions may be obtained also for tasks that intuitively seem to require memory.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles; F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; F.1.1 [**Models of Computation**]: Unbounded-action devices

General Terms

Algorithms,Theory

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 obtains information about these positions in terms of 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 systems used by two different robots. Each robot computes its next location based on the visual information (the vision of a robot is unrestricted) and moves to this location. There is no central coordinator; the robots are identical, they have the same capabilities, and execute the same (deterministic) algorithm.

These systems have been extensively investigated by researchers from robotics, AI, control and, more recently, distributed computing (e.g., [1, 3, 5, 6, 7, 9, 12, 13, 14, 15, 16, 22]). A particular subclass 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. A system of oblivious robots is inherently *self-stabilizing* in the sense that the robots are allowed to start from any arbitrary state. Hence there is a strong interest in such systems (e.g., [4, 3, 6, 7, 12, 13, 14, 16, 22]). The present paper considers robots which act deterministically. Designing deterministic algorithms for oblivious robots is specially challenging and some simple tasks such as the gath-

*This paper has been partially supported by NSERC Discovery Grants, and by a Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

PODC'10, July 25–28, 2010, Zurich, Switzerland.

ering of two robots, are known to be impossible for oblivious robots [22].

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, or scattering uniformly in a given region. 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, 3, 8, 10, 7, 11, 17, 19, 20]). The *arbitrary pattern formation* problem, that of forming any pattern given in input, has also been studied [2, 16, 21, 22].

In the case of *oblivious* robots, formation problems have been investigated for specific patterns, in particular the circle [6, 12, 14, 18]. At a more general level, the research has focused on the characterization of which patterns are possible under what conditions [2, 16, 22]. 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]. In the semi-synchronous model, it has been recently shown [23] that oblivious robots can form exactly the same patterns that non-oblivious robots 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 patterns 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 real computational limits imposed by the robots being oblivious. In other words, *what are the limiting powers of obliviousness*?

1.2 Contributions and Organization

This paper studies the computational power of a system of oblivious robots in the semi-synchronous model. We consider different scenarios, starting from the case of completely anonymous robots, to that of visibly indistinguishable but ordered set of robots and finally, distinctly labeled robots. In each case, we study the series of patterns that may be formed by such robots. An immediate observation is that no *finite* (non-trivial) series of distinct patterns can be completely formed in finite time starting from arbitrary initial configurations—as explained later, an adversary can force any protocol to produce at most a single pattern. Thus the focus of this paper is forming *periodic* (or cyclic) series S^{∞}, i.e. the periodic repetition of a finite series S of distinct patterns. The results we obtain are immediately generalizable to infinite *aperiodic* series.

We prove that if the robots have distinct visible identities then any series can be formed provided that there are sufficient number of robots. The same result holds for robots having invisible but distinct identities (except for the case when there are exactly three robots). In case of anonymous robots, the series that may be formed depends on the symmetricity of the initial configuration. 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 more general (thus weaker) case of anonymous robots. The strongest case when robots have distinct visible identities is examined in Section 4. Finally the series formable when the robots have distinct but invisible identities is investigated in Section 5. Due to space limitations, some of the proofs are omitted.

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. Each step is instantaneous but between two steps, the robot may be inactive for an arbitrary amount of time. This model of asynchrony was introduced in [22] and is sometimes called the *semi-synchronous* model or *ATOM* model. In this model, time is discretized into rounds and at each round, an arbitrary subset of the robots (selected by a scheduler) are active. The robots that are active in a round complete exactly one step in that round.

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. During the COMPUTE stage, an active robot executes an algorithm that determines its next location based on the information obtained from the LOOK operation and the identifier of the robot (if any). During the MOVE stage of the step, the robot moves to the location computed during the COMPUTE stage. Notice that movements of robots are instantaneous, since the complete step is completed within one instance of time.

The robots may not agree on a common sense of direction, but they agree on a unique sense of orientation, i.e. the robots can distinguish clockwise from counterclockwise. 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 k 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 and robots in distinct locations do not obstruct the view of one another. The visual capabilities of the robots allows them to detect multiplicity and they can count how many robots are at the same location. In order to describe our algorithms and the global configuration of robots during the algorithm, we shall use a fixed coordinate system[1]; 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 k robots on the plane is denoted by the multiset $\gamma = \{(label(r_i), L(r_i)) : 1 \leq i \leq k\}$ 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 $i \in \{1, 2, \ldots, k\}$. The configuration at a specific time t is denoted by $\gamma(t)$. Given a configuration γ, we denote by $L(\gamma)$ the set of distinct points occupied by robots in the configuration γ, i.e. $L(\gamma) = \{(x, y) : \exists l, (l, x, y) \in \gamma\}$. The set $L(\gamma))$ contains at least one and at most k points.

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. The diameter of a circle C is denoted by D(C). For a configuration γ, we define SEC(γ) to be the SEC for the points in $L(\gamma)$. In any given configuration γ, the information obtained by a robot by looking at its surroundings is called the *view* of the robot.

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), $\ldots$, (x_n, y_n) $n \geq 1$, 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 uniform scaling of pattern P_i. The *size* of a pattern P_i is its cardinality and will be denoted by n_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`(n) : For any $n \geq 3$, this is the pattern consisting of points $p_1, p_2, \ldots, p_n$ that are vertices of a regular convex polygon of n sides.

For any pattern P of size $n > 1$, we define the *symmetricity* $\rho(P)$ to be the largest integer $q \geq 1$ such that the smallest enclosing circle containing P can be partitioned into exactly q equiangular sectors 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 ρ(`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. A *linear pattern series* 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 *cyclicly 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 2.1. *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.*

The above lemma follows from the observation that the system may start from a configuration analogous to one forming pattern P_m, for instance. In that case, the algorithm would terminate immediately.

This means that, for any non-trivial (i.e. $m \geq 2$) finite series S, a terminating algorithm can form only a suffix of S. Thus, we are interested in constructing non-terminating algorithms that completely form a given series S, i.e. they form the cyclic series S^∞. Our focus will be on determining which cyclic series are completely formable by a team of oblivious robots.

3. ANONYMOUS ROBOTS

3.1 Preliminaries

In this section, the robots are assumed to be anonymous and oblivious, having no other additional capabilities. The algorithm followed by each robot must be identical and depends only on the current configuration as viewed by the robot. If two robots in the same location are activated simultaneously, they may both move to the same location. Since the activation schedule is decided by an adversary, this implies the following property:

Property 3.1. *From a configuration γ consisting of anonymous robots at w distinct locations, we may not reach a configuration γ' where the robots occupy more than w distinct locations.*

If two robots start from symmetric positions within a configuration, then these two robots may take similar actions and this symmetry may be maintained (if the activation schedule is decided by an adversary). This notion of symmetry in a configuration is quantified using the concept of *symmetricity*, that was introduced in [22]. We now formally define the *view* of a robot, distinguishing the cases when the robot lies or does not lie on the center of the smallest enclosing circle in a configuration.

Definition 3.1. *If robot r is not located at the center c of the SEC of the current configuration γ then its view $\mathcal{V}(r)$*

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

is defined as follows. Consider the coordinate system Z_r defined by taking the location of r as the origin $(0,0)$ and the point c as $(1,0)$, and represent the locations of all robots according to this coordinate system. The view $\mathcal{V}(r)$ is then the set of tuples $\{(x_i, y_i, k_i) : k_i > 0$ robots are located at $(x_i, y_i)\}$.

DEFINITION 3.2. *If robot r is located at the center c of the SEC of the current configuration γ then its view $\mathcal{V}(r)$ is defined as follows. If there are robots that are not located at c, select any robot r' whose view is minimum among all such robots. Consider the coordinate system Z_r defined by taking the location of r as the origin $(0,0)$ and the location of r' as $(1,0)$, and represent the locations of all robots according to this coordinate system. The view of r, $\mathcal{V}(r)$ is then the set of tuples $\{(x_i, y_i, k_i) : k_i$ robots are located at $(x_i, y_i)\}$. On the other hand, if there are no robots outside of c, then $\mathcal{V}(r) = \{(0,0,k)\}$.*

The above definition requires a total order on distinct views, which can be obtained as follows. The elements of $\mathcal{V}(r)$ can be ordered lexicographically to obtain an ordered sequence $Q(\mathcal{V}(r))$, for each robot $r \in \gamma$. For any two robots r_1 and r_2, the ordered sequences $Q(\mathcal{V}(r_1))$ and $Q(\mathcal{V}(r_2))$ contain the same number of elements and these sequences can be ordered lexicographically. So, $\mathcal{V}(r_1) < \mathcal{V}(r_2)$ if and only if $Q(\mathcal{V}(r_1))$ is lexicographically smaller than $Q(\mathcal{V}(r_2))$.
Note that our definitions of view differ from the one used in the previous literature, notably in [22] where the *views* contained information not only about the locations of other robots with respect to the location of r, but also information about coordinate systems used by the robots. Recall that in our scenario, the coordinate system used by a robot is not invariant during the execution of the algorithm[2]. Thus, in our definition, the *view* of a robot must be independent of the coordinate system used by the robot. In the following, we shall denote by $mult(r)$ the multiplicity of the location occupied by r.

PROPERTY 3.2. *Given a configuration γ that does not contain any robots at the center c of SEC, we observe the following properties:*
(1) Robots that are collocated have identical views
(2) If $\mathcal{V}(r_i) = \mathcal{V}(r_j)$ then $mult(r_i) = mult(r_j)$.
(3) There exists an integer $q \geq 1$ such that (i) for any robot r, there are exactly $q \cdot l$ robots that have the same view as robot r, where $l = mult(r)$, (ii) q is the size of any maximal subset R of robots that have identical views but are located at mutually distinct locations, and (iii) if $q \geq 3$ then the q robots $r_1, r_2, \ldots, r_q \in R$ occupy the vertices of a regular convex polygon of q sides whose center is c [3].

Based on our definition of views of robots, we define the symmetricity of a configuration γ. (See Figure 1 for an illustration.)

DEFINITION 3.3. *The* symmetricity, *$\rho(\gamma)$ of a configuration γ consisting of k 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. $\rho(\gamma) = \max\{q : \forall r_i \in \gamma, q$ divides $|\{r_j \in \gamma : \mathcal{V}(r_j) = \mathcal{V}(r_i)\}|\}$*

[2] i.e. each time a robot wakes up and execute the LOOK operation, it may use a different coordinate system than the one in previous steps of the same robot.
[3] In the special case of $q = 2$, the robots r_1, r_2 occupy the endpoints of the diameter of a circle centered at c

PROPERTY 3.3. *For any configuration γ where robots occupy $w > 1$ locations, $\rho(\gamma)$ divides w.*

LEMMA 3.1. *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 simultanoeusly). 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

Property 3.1 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 k, where k is the number of robots. Each robot starts from a distinct location and during the pattern formation, no two robots should occupt the same location (i.e. we can not allow points of multiplicity). Due to Property 3.1 and Lemma 3.1, we know the following impossibility result:

LEMMA 3.2. *A cyclic series of distinct patterns $\langle P_1, P_2, \ldots P_m\rangle^\infty$, is formable only if $size(P_i) = size(P_j)$ and $\rho(P_i) = \rho(P_j)$, $\forall\, i, j \in \{1, 2, \ldots m\}$.*

We now show that pattern series of the above type can indeed be formed, provided that the initial configuration is not more symmetric than the patterns. We first define some special types of configurations that can be easily identified by the robots.

DEFINITION 3.4. *(BCC): A configuration is called a* bi-circular configuration *(BCC) if the locations of the robots satisfy the following condition. There is one unique location called the* pivot *location, such that the smallest enclosing circle (SEC) containing all the robots has a diameter more than thrice the diameter of the SEC containing all robots except those at the pivot. Further the two circles intersect at exactly one point (the point directly opposite the pivot) which is called the* base-point *(BP). The former circle is called the* primary enclosure *(or, SEC1) while the latter is called the* secondary enclosure *(or, SEC2). The point on the secondary enclosure directly opposite the base-point is called the* frontier-point *(FP).*

The ratio of the diameter of the primary enclosure over the diameter of the secondary enclosure is called the *stretch* of the bi-circular configuration. (See Figure 2.)

DEFINITION 3.5. *A configuration γ containing k robots is called q-symmetric-circular configuration or, SCC[q], $1 < q < k$, if the following three conditions hold:*

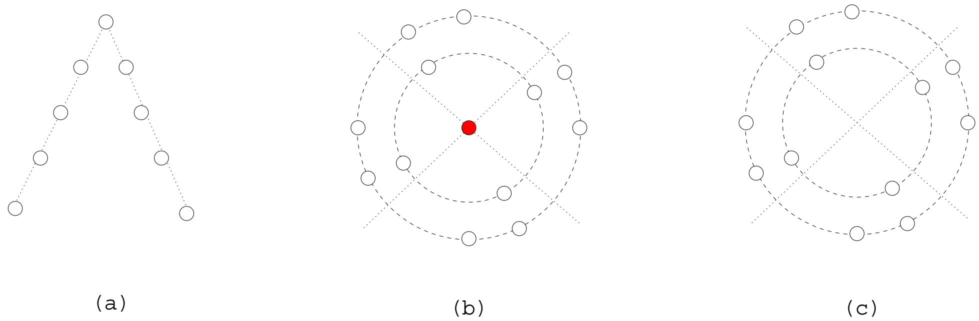

Figure 1: A configuration C with (a) symmetricity one, where robots at distinct locations have distinct views; (b) symmetricity one where only one robot has unique view; (c) symmetricity $\rho(C) = 4$.

(1) The smallest circle enclosing all robots (called SEC1) has exactly q points on its circumference that are occupied by robots.
(2) All the other robots (except those on the SEC1) lie on or in the interior of a smaller circle called SEC2 that is concentric to SEC1 such that $D(SEC1) \geq (5+\sin^{-1}(\pi/q)) * D(SEC2)$.
(3) There are no robots in the center of SEC1.

As before, the former circle (SEC1) is called the *primary enclosure* while the latter (i.e. SEC2) 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 q-symmetric-circular configuration.

LEMMA 3.3. *Starting from any configuration γ with symmetricity $\rho(\gamma) = q$, $1 < q < k$, and for any $t \geq (5 + \sin^{-1}(\pi/q))$ we can reach a configuration γ' such that either (i) γ' is SCC[q'] having stretch t, where $q' > 1$ is a factor of q, or, (ii) γ' is BCC having stretch $t' = (t + 1)/2$.*

LEMMA 3.4. *Starting from a configuration of type SCC[q], $q > 1$, with k robots occupying distinct locations we can form any pattern P such that the symmetricity $\rho(P) = q \cdot a$, $a \geq 1$ and size(P)= k.*

LEMMA 3.5. *In any bi-circular configuration, the robots can agree on a unique coordinate system.*

PROOF. 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. □

Note that agreement on a coordinate system implies that there exists a unique way of ordering robots. Thus, it is easy to form any single pattern as long as the BCC configuration is maintained. We now describe the algorithm for forming a cyclic series of distinct patterns $\langle P_1, P_2, ...P_m\rangle^\infty$, starting from a configuration γ, such that $\forall i$, size$(P_i) = k$, and $\rho(P_i) = q = \rho(\gamma)$. (We only need to consider the case $q < k$, since for $q = k$ there is a single possible pattern: the regular k-gon and the robots already form that.) Let F be a function that maps each pattern P_i to a real number $t_i = F(P_i)$. To signal the formation of pattern P_i, we use either the configuration SCC[x] with stretch t_i, where x is any factor of q or, the configuration BCC with stretch $t'_i = (t_i + 1)/2$. Due to Lemma 3.3 it is possible to form one of these configurations starting from arbitrary configurations of symmetricity q. By computing the stretch of the configuration, the robot can identify which pattern is being formed. During the formation of the pattern P_i, at each intermediate configuration each robot can uniquely identify the pattern. Once the pattern has been formed the resulting configuration has symmetricity q. It is thus again possible to form the configuration SCC[x] or BCC having the appropriate stretch for the next pattern P_{i+1} in the sequence. Using this technique, the robots can move from one pattern to the next, without requiring any memory of the past.

From Lemma 3.2 and from the algorithm described above, we obtain the following characterization.

THEOREM 3.1. *With k anonymous robots starting from distinct locations in an arbitrary configuration γ, we can form a cyclic series of distinct patterns $\langle P_1, P_2, ...P_m\rangle^\infty$, each of size k, if and only if $\rho(P_i) = \rho(P_j) = \rho(\gamma) \cdot a$, $a \geq 1$, $\forall i, j \in \{1, 2, \ldots m\}$.*

3.3 Special Case: Agreement on Directions

In this section, we assume that, besides agreeing on a common clock-wise direction, the robots also agree on the directions of a fixed coordinate system. This allows us to break the symmetry between the robots. Notice that 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 3.6. *With k anonymous robots having common sense of direction, any single pattern P of size $n \leq k$ points can be formed, if the robots start from distinct locations.*

PROOF. As mentioned before, there is a total order on the k robots, say $r_1, r_2, \ldots, r_k$ 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_k simply move to the

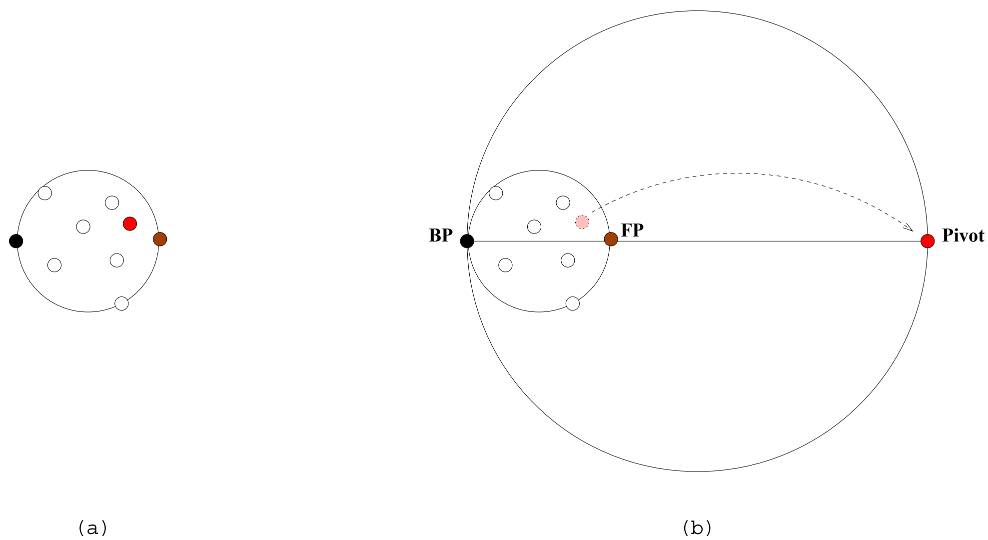

Figure 2: (a) An arbitrary configuration of robots and the smallest enclosing circle. (b) A bi-circular configuration.

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 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 k robots are initially in mutually distinct locations. We now show which series of patterns are formable starting from any arbitrary configuration.

THEOREM 3.2. *With k anonymous robots having a common sense of direction and occupying w distinct locations, we can form any cyclic series of distinct patterns $\langle P_1, P_2, \ldots P_m \rangle^\infty$, if and only if size($P_i$) = size($P_{i+1}$) $\leq w$, for all i, $1 \leq i < m$.*

PROOF. The "only if" part follows from Property 3.1. We only need to show how to form the given series. Let r_1,r_2,..., 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 $d(r_1, r_l)/d(r_1, r_2) = F(P_i)$ (r_l denotes the last robot). 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

When there is no agreement on directions, the symmetry among the robots can be broken by the use of distinct labels. In this section, we assume that each robot r_i has a unique identity ID_i and any other robot can read this identity even from a distance. Without loss of generality, we can assume that $ID_i = i$ (i.e. the i-th robot is numbered i). In this case, the view of each robot is unique as it contains information about both the identities and locations of the other robots. Thus, there can be no symmetric configurations. In this case, we can allow the robots to form points of multiplicity, since the robots can be separated later, if required (i.e. Property 3.1 does not hold anymore).

As we showed in section 3.3, having an order on the robots allows us to form any pattern of size $n \leq k$. However, for labelled robots, the order is preserved even if the robots are not in distinct location.

LEMMA 4.1. *With k robots having visibly distinct identities, any single pattern P of size $n \leq k$ points can be formed.*

PROOF. This is achieved by Algorithm 1. The case for P = `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. □

We now consider the series of patterns that may be formed by visibly distinct robots. For forming any interesting series of patterns there must be at least three robots. For only one robot, no non-trivial series are possible. For $k = 2$ robots, it is easy to form the only possible series $\langle$`POINT`, `TWO-POINTS` $\rangle^\infty$, by movement of a single robot (say r_2). We show below that with $k \geq 3$ robots, we can form any series of distinct

Algorithm 1: Form-Pattern

```
/* Algorithm for single pattern P */
INPUT: P =(p_1, p_2, ..., p_n) , ID = i
begin
    if P = 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 ← 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 ∈ P;
```

patterns $S = \langle P_1, P_2, \ldots, P_m\rangle$, with the only restriction that each pattern P_i has at most k points. Let us first consider a series of patterns not containing POINT.

As before we use a function F to associate each pattern P_i to a real number $t_i = F(P_i)$, $t_i \in (1, \infty)$. When forming the pattern P_i, the last robot r_k moves to a location between the first two robots r_1 and r_2 such that the ratio of distances $d(r_1, r_2)/d(r_1, r_k)$ is exactly equal to t_i. Any robot can compute this ratio by looking at the current configuration of robots and thus, it can determine which pattern P_j in the series is being formed. The ordered set of robots r_3 to r_k can be assigned to points in the pattern P_j. The robots r_1, r_2, and r_k remain stationary until each of the other robots has moved into the correct position, and then robot r_k moves to complete the pattern. Notice that robots r_1 and r_2 do not need to move as the system transforms from one pattern to the next. This algorithm works for any finite series of patterns not containing the POINT pattern. 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_k 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_{k-1} in the same location and r_k 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_k in the same location and r_1 in a different location.
(iv) *FormRatio*(t): This is the set of configurations where robots r_1, r_2, r_k lie on distinct locations on the same line such that the ratio of distances $d(r_1, r_2)/d(r_1, r_k) = t$. (The other robots may be located anywhere.)

The algorithm for forming a given series $S^\infty = \langle P_1, P_2, \ldots P_m \rangle^\infty$ is called *Form-Series*: it is sketched below and reported in Algoritm 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 $t = F(P_{(i \mod m)+1})$.

Algorithm 2: FormSeries

```
/* Algorithm for forming a series S^∞ with visibly distinct
robots */
INPUT: S = ⟨P_1, P_2, ..., P_m⟩, ID = i
begin
    case ID = 2
        if all robots except r_k are colocated then
            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_k are colocated
        then
            if ∃P_j ∈ S : P_j = POINT and d(r_1, r_2)/d(r_1, r_k)
            equals F(P_j) then
                Move to the location of robot r_1;
    case 2 < ID < k
        if r_1, r_2 and r_k are at distinct locations then
            t ← d(r_1, r_2)/d(r_1, r_k) ;
            if ∃j ∈ [1, m] : F(P_j) = t then
                Execute Form-Pattern (P_j, ID) ;
    case ID = k
        if all robots except r_k are colocated then
            Move to the same location as the other robots;
        else if all robots except r_2 are colocated then
            if ∃j ∈ {1, 2, ..., m} : P_j = POINT then
                Move to a location such that the ratio
                d(r_1, r_2)/d(r_1, r_k) = F(P_{(j mod m)+1});
            else
                Move to a location such that
                d(r_1, r_2)/d(r_1, r_k) = F(P_1);
        else if r_1 is not colocated with r_2 then
            t ← d(r_1, r_2)/d(r_1, r_k) ;
            if ∃j ∈ {1, 2, ..., 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 ∈ S then
                Move to a location such that
                d(r_1, r_2)/d(r_1, r_k) = F(P_{(j mod m)+1});
            else
                // Start to form pattern P_1
                Move to a location such that
                d(r_1, r_2)/d(r_1, r_k) = F(P_1);
```

LEMMA 4.2. *Given a finite series of distinct patterns $S = \langle P_1, P_2, \ldots, P_m\rangle$ Algorithm* Form-Series *forms S^∞ if size(P_i) $= n_i \le k$, for all $i = 1, 2, \ldots, m$.*

PROOF. During the algorithm, the robot system transforms through a series of configurations γ_0, γ_1, ..., where for all $i > 1$, either γ_i corresponds to some pattern $P_i \in S \bigcup \{\texttt{POINT}\}$ or γ_i is one of the following configurations: *PrePoint*, *PostPoint*, *TwoPoints*, or any configuration of the type *FormRatio*(t) for $t = F(P_i)$, $1 \leq i \leq m$. 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_k 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_k to form configuration *FormRatio*$(F(P_1))$. Otherwise r_1 and r_2 are initially seperated and in this case only r_k 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*(t) where $t = F(P_{(j \mod m)+1})$ 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. □

THEOREM 4.1. *With $k \geq 1$ robots having distinct visible identities, we can form any finite series of distinct patterns $\langle P_1, P_2, \ldots, P_m\rangle$ if and only if for all i, $1 \leq i \leq m$, size(P_i) $= n_i \leq k$.*

5. DISTINCT ROBOTS WITH INVISIBLE IDENTITIES

In this section, we consider a model that is between the two extremes considered before. We assume that the robots are ordered with labels $1, 2, 3, \ldots, k$ 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 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 labels.

When there are only two robots, this scenario is equivalent to the previous one when the robots had visible identities and thus, the same results follow for that particular case. However, for $k \geq 3$, we have slightly different results.

5.1 Exactly Three Robots ($k = 3$)

Let γ be any configuration consisting of three robots r_1, r_2, r_3. Then either $L(\gamma)$ is a triangle or it is isomorphic to `POINT` or `TWO-POINTS`. There are three types of configurations where the robots occupy exactly two distinct locations (depending on which of the three robots is alone). In the following, we shall use the notation $C(i, j; l)$, $i, j, l \in \{1, 2, 3\}$ to represent any configuration (of three robots) where robots r_i and r_j are collocated, but the third robot r_l is not collocated with the others. If all three robots are at distinct locations then such a configuration is called a triangular configuration. We now show the following possibility result:

LEMMA 5.1. *With $k = 3$ robots, we can form a series S^∞ where S has the following forms:*
(i) $S = \langle P_1, P_2, \ldots, P_m,$ `POINT`, `TWO-POINTS`$\rangle$,
(ii) $S = \langle P_1, P_2, \ldots, P_m,$ `TWO-POINTS`, `POINT`$\rangle$, or
(iii) any subsequence of the above series,
such that size(P_i)= 3 for $i = 1, 2, \ldots, m$ and for any $i \neq j$, P_i and P_j are non-isomorphic.

We briefly describe here how the above series are formed. Let us consider the series of the first type. Note that each pattern P_i consists of exactly three points and so the transformation from P_i to P_{i+1} is trivially achieved by the movement of only one robot (say r_3). Thus, there is no intermediate configuration during these transformations. To form `POINT` from a triangular configuration (representing pattern P_m), robots r_2 and r_3 (in this order) move to the location of r_1. Thus the intermediate configuration is $C(1, 2; 3)$. To form `TWO-POINTS` from `POINT`, robot r_1 moves away from the others. Thus, the configuration representing `TWO-POINTS` is $C(1; 2, 3)$ which can be distinguished from the previous intermediate configuration $C(1, 2; 3)$, by the robot r_3. To form pattern P_1 again, robot r_3 now moves away from r_2 to the appropriate location.

The series of the second type can be formed in a similar manner. In this case, we need to form `POINT` from `TWO-POINTS` and this can be achieved by a single movement of robot r_1. During the whole series, there is only one intermediate configuration and this occurs when transforming from `POINT` to P_1. This intermediate configuration is $C(1, 3; 2)$ which can be distinguished from the `TWO-POINTS` configuration $C(1; 2, 3)$, by the robot r_1. Thus, robot r_1 can unambiguously determine the next location that it has to move to.

The fact that the three configurations of type $C(i, j; l)$ can not be distinguished by any single robot, implies the following impossibility result:

LEMMA 5.2. *Given any non-trivial series $\langle P_1, P_2, \ldots, P_m\rangle$, where size($P_i$)> 2 and any l, $1 \leq l < m$, three robots can not form a series of the form*

$$\langle P_1, P_2, \ldots, P_l, \texttt{POINT}, P_{l+1}, P_{l+2}, \ldots, P_m, \ \texttt{TWO-POINTS}\rangle^\infty$$

Thus, we have a characterization of series that can be formed with $k = 3$ robots having invisible identities. In the following the notation $S^{[0/1]}$ denotes either zero or one occurrence of the series S of patterns.

THEOREM 5.1. *With exactly $k = 3$ robots, we can form a series S^∞ iff S is a cyclic rotation of*
$\langle P_1, P_2, \ldots, P_m\rangle^{[0/1]} \langle \texttt{POINT}\rangle^{[0/1]} \langle \texttt{TWO-POINTS}\rangle^{[0/1]}$ or,
$\langle P_1, P_2, \ldots, P_m\rangle^{[0/1]} \langle \texttt{TWO-POINTS}\rangle^{[0/1]} \langle \texttt{POINT}\rangle^{[0/1]}$
where $m \geq 1$, size(P_i)= 3 for $i = 1, 2, \ldots, m$ and for any $i \neq j$, P_i and P_j are non-isomorphic.

5.2 More than three Robots ($k > 3$)

We now consider the case when there are at least four robots. In this case, the impossibility from Lemma 5.2 no longer holds. As in previous sections, a system of more than three robots may go through many intermediate configurations during the transition from one pattern to the next. The intermediate configurations must encode information about the pattern to be formed. In the following, we use again the BCC configuration, defined earlier, to signal the formation of specific patterns in a series. In this case we allow points of multiplicity and we shall ensure that there is at least one robot each at the pivot and at the base-point of the bicircular configuration.

LEMMA 5.3. *From any arbitrary configuration γ with $|L(\gamma)| \geq 3$, we can form a bi-circular configuration of any given stretch $t > 3$, by movement of a single robot. (This single robot will place itself in a pivot position).*

We now describe the technique for forming any given pattern P_i starting from a bi-circular configuration of stretch t_i. (As mentioned before, the bi-circular configuration can be formed by robot r_k jumping to the pivot location). Once the robots are in bi-circular configuration BCC(t_i), robot r_1 and robot r_{k-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 the pattern P. We assign the robots to these positions 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 and every robot can obtain the same set $\Gamma(P)$ of locations that correspond to points in the pattern P. (Note that the coordinates of the points in $\Gamma(P)$ are assigned with respect to the fixed coordinate system defined by the bicircular configuration.) The elements of $\Gamma(P)$ are sorted in such a way that the first point is the base-point and all points which lie on the SEC2 precede those that are located in the interior of SEC2. For $1 \leq i \leq n$ robot r_i is assigned the ith location in $\Gamma(P)$ and for $n < j \leq k$ robot r_j is assigned the n-th location in $\Gamma(P)$.

The algorithm that implements the above strategy is called *Form-Series-2* and is reported in 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_{k-1} and r_k stationary at the BP, FP and pivot positions. When all other robots have moved to their assigned location, robot r_{k-1} finally moves to its assigned location. If r_{k-1} is assigned a location inside the SEC2, then all the points on the bounding circle are already occupied by robots; So, the SEC2 (and thus the BCC) is preserved after the move of r_{k-1}. Otherwise, if r_{k-1} is assigned another position on the SEC2, then there must be already two other robots on the SEC2 (since $k \geq 4$) and thus, the BCC is preserved after the move of r_{k-1} (Any three points uniquely determine the only circle passing through them). Thus, after the move of r_{k-1}, the BCC configuration of appropriate stretch is still maintained. Hence robot r_k can unambiguously move to the required position to complete the pattern.

When the BCC configuration is initially formed, the robots r_1 and r_{k-1} may not be present at the base-point(BP) and

Algorithm 3: FormSeries2

```
/* Algorithm for forming a series S^∞; Robots with invisible
identities */
INPUT: S = ⟨P_1, P_2, ..., P_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(t_j) for some j ∈ [1, m] then
            if loc ≠ BP and FP is unoccupied then
                Move to FP;
            else if loc ≠ BP and FP is occupied by at least
            one other robot then
                Move to BP;
            else if loc = FP and k − 2 other robots are
            occupying at least n_j − 2 points of P_j then
                MoveToPosition(P_j, k − 1);
        else if all robots are collocated then        /* i.e.
        L(Conf) ≃ POINT  */
            Move to an unoccupied location;
        else if all robots except one are at loc and POINT
        ∉ S then              /* maybe only r_k is alone */
            Move to an unoccupied location within current
            SEC;
    case ID = k − 1    /* Algorithm for robot r_{k−1}    */
        if Conf is BCC(t_j) for some j ∈ [1, m] then
            if loc ≠ FP and loc ≠ BP then
                Move to FP;
            else if loc = BP and multiplicity(BP) > 1
            then
                Move to FP;
            else if loc = FP and k − 2 other robots are
            occupying at least n_j − 2 points of P_j then
                MoveToPosition(P_j, k − 1);
    case ID = k              /* Algorithm for robot r_k */
        if Conf is BCC(t_j) for some j ∈ [1, m] then
            if all other robots are occupying at least n_j − 1
            points of P_j then
                MoveToPosition(P_j, k);
        else if all robots except one are at loc then
            if P_j = POINT for some j ∈ [1, m] then
                Move to pivot to form BCC (t_{(j mod m)+1});
            else
                Move to pivot to form BCC (t_1);
        else if all other robots are at the same location
        l ≠ loc and POINT ∈ S then
            Move to the location occupied by all other
            robots;
        else if L(Conf) matches P_j for j ∈ [1, m], P_j ≠
        POINT then
            Move to pivot to form BCC (t_{(j mod m)+1});
        else
            Move to pivot to form BCC (t_1);
    otherwise /* Algorithm for all other robots, i.e.
    1 < ID < k − 1   */
        if Conf is BCC(t_j) for some j ∈ [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 ≠ BP or multiplicity(BP) > 1 then
                    MoveToPosition(P_j, i);
```

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_{k-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_{k-1} simply reverse their roles (i.e. r_{k-1} remains stationary at BP until the pattern is formed, while r_1 moves from the FP only in the penultimate stage, to go to the location assigned to r_{k-1}).

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_k are together and all other robots are collocated at a separate location. Since $k \geq 4$, these three special configurations can be distinguished from eachother by both robot r_1 and robot r_k.

LEMMA 5.4. *Algorithm Form-Series-2 executed by $k \geq 4$ distinct robots forms the given series $S = \langle P_1, P_2, \ldots, P_m \rangle$ of distinct patterns, if size(P_i)$\leq k$, for $i = 1, 2, \ldots, m$.*

Thus we can conclude:

THEOREM 5.2. *With $k \geq 4$ robots having distinct invisible identities, a series of distinct patterns $\langle P_1, P_2, \ldots, P_m \rangle$ is formable if and only if size(P_i)$\leq k$, $1 \leq i \leq m$.*

6. REFERENCES

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. on Computing*, 36(1):56–82, 2006.

[2] H. Ando, S. Fujita, I. Suzuki, and M. Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *IEEE Symposium of Intelligent Control*, pages 453–460, 1995.

[3] 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.

[4] Y. Asahiro, S. Fujita, I. Suzuki, and M. Yamashita. A self-stabilizing marching algorithm for a group of oblivious robots. In *12th Int. Conf. on Principles of Distributed Systems (OPODIS)*, pages 125–144, 2008.

[5] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[6] 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.

[7] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Automata, Languages and Programming, 30th International Colloquium, (ICALP'03)*, pages 1181–1196, 2003.

[8] R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM J. on Computing*, 34(6):1516–1528, 2005.

[9] R. Cohen and D. Peleg. Local spreading algorithms for autonomous robot systems. *Theoretical Computer Science*, 399(1-2):71–82, 2008.

[10] J. Czyzowicz, L. Gasieniec, and A. Pelc. Gathering few fat mobile robots in the plane. *Theoretical Computer Science*, 410(6-7):481–499, 2009.

[11] X. Défago, M. Gradinariu, S. Messika, and P. R. Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In *20th Int. Symposium on Distributed Computing (DISC)*, pages 46–60, 2006.

[12] 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.

[13] 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.

[14] Y. Dieudonné, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. *ACM Transactions on Autonomous and Adaptive Systems*, 3(4), 2008.

[15] 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.

[16] P. Flocchini, G. Prencipe, and N. Santoro. Self-deployment of mobile sensors on a ring. *Theoretical Computer Science*, 402(1):67 – 80, 2008. Algorithmic Aspects of Wireless Sensor Networks.

[17] 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.

[18] 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.

[19] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. parts 1 and 2. *SIAM J. on Control and Optimization*, 46(6):2120–2147, 2007.

[20] S. Souissi, X. Défago, and M. Yamashita. Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. *ACM Transactions on Autonomous and Adaptive Systems*, 4(1), 2009.

[21] K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotics Systems*, 13:127–139, 1996.

[22] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. on Computing*, 28(4):1347–1363, 1999.

[23] M. Yamashita and I. Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theoretical Computer Science*, 411(26-28):2433 – 2453, 2010.