Optimum Circle Formation by Autonomous Robots

Subhash Bhagat and Krishnendu Mukhopadhyaya

Abstract This paper considers a constrained version of the *circle formation* problem for a set of asynchronous, autonomous robots on the Euclidean plane. The *circle formation* problem asks a set of autonomous, mobile robots, initially having distinct locations, to place themselves, within finite time, at distinct locations on the circumference of a circle (not defined a priori), without colliding with each other. The *constrained circle formation* problem demands that in addition the maximum distance moved by any robot to solve the problem should be minimized. A basic objective of the optimization constrain is that it implies energy savings of the robots. This paper presents results in two parts. First, it is shown that the constrained circle formation problem is not solvable for oblivious asynchronous robots under *ASYNC* model even if the robots have rigid movements. Then the problem is studied for robots which have $O(1)$ bits of persistent memory. The initial robot configurations, for which the problem is not solvable in this model, are characterized. For other configurations, a distributed algorithm is presented to solve the problem for asynchronous robots. Only one bit of persistent memory is needed in the proposed algorithm.

Keywords Swarm robots · Asynchronous · Circle formation
Robots with persistent lights

1 Introduction

A *robot swarm* consists of small, autonomous, indistinguishable, inexpensive mobile robots. Robots in such a distributed system work cooperatively to accomplish some common task which cannot be done by a single large robot. The robots are autonomous (they work without any centralized control), homogeneous (all of them

S. Bhagat (✉) · K. Mukhopadhyaya
Advanced Computing and Microelectronics Unit,
Indian Statistical Institute, Kolkata, India
e-mail: sbhagat_r@isical.ac.in

K. Mukhopadhyaya
e-mail: krishnendu@isical.ac.in

R. Chaki et al. (eds.), *Advanced Computing and Systems for Security*,
Advances in Intelligent Systems and Computing 666,
https://doi.org/10.1007/978-981-10-8180-4_10

have same capabilities) and anonymous (they are indistinguishable by their appearances and nature). All of them execute the same algorithm. In general, robots lack explicit communication capability. The robots can implicitly communicate with each other by sensing the positions of other robots in the system, using endowed sensors. The system does not have any global coordinate axes. Each robot owns a local coordinate system having origin at its current position. The local coordinate systems of two different robots may have different directions and orientations of coordinate axes and unit distances. In general, the robots do not remember any piece of information of their previous computational cycles, i.e. they are oblivious.

A robot has one of the two states at any point of time: active or inactive. Initially, all robots are inactive. On activation, a robot executes its computational cycle consisting of *Look–Compute–Move* phases. During the *Look* phase, a robot takes snapshot of its surrounding environment, using its sensing capability, to obtain the positions of other robots. Considering the input from the *Look* phase, the *Compute* phase outputs a destination point for the robot. Finally, the robot moves towards the destination point during the *Move* phase. An idle robot remains silent without performing any course of action. Robots may be endowed with some additional capabilities or may have some common agreements in order to solve different coordination problems. The memory model assumes that the robots possess a constant amount of additional persistent memories to remember their current states. The implementation of such persistent memory is done by externally visible lights. These lights use a constant number of colours. The colours are predefined to indicate different states of the robots [3, 8]. The robots may have some agreement on the directions and orientations of local coordinate axes. They may share a common handedness or chirality (clockwise direction).

Three types of basic models are used. These models are defined according to the schedules of the operations and activation of the robots. *Asynchronous* (*ASYNC or CORDA*) model [13] is the most general one in which robots are activated arbitrarily and independently of each other. The time duration of the operations by the robots is unpredictable but finite. This implies that a robot may have done its computations on obsolete data. Due to this unpredictability, the problems become more difficult to solve in this model. The second model is the *semi-synchronous* (*SSYNC or ATOM*) [15] model. In *SSYNC* model, time is discretized into several rounds and the robots are activated in these rounds. A subset of robots becomes activated all together in a round and performs their operations instantaneously in that round. During movements, a robot is not observed by other robots in the system. The subset of robots activated in a round is not known in advance. In *fully synchronous* (*FSYNC*) model, all robots become activated in all rounds. This work assumes that scheduler activates each robot infinitely often, i.e. a scheduler is a fair [4].

Under these settings, a variety of geometric problems have been studied by the researchers. These problems include *gathering*, *arbitrary pattern formation*, *circle formation*, etc. The *circle formation* problem is defined in the following manner: a set of robots, occupying positions in the Euclidean plane, should work cooperatively to occupy distinct positions on the circumference of a circle not known a priory and this should be done within finite time. The *constrained circle formation* problem requires that while solving the circle formation problem, the maximum distance moved by any robot should be minimized.

1.1 Earlier Works

In literature, different solutions for the *circle formation* problem have been proposed under different schedulers and assumptions on the capabilities of the robots. The basic objective of these works is to propose solutions which minimize the sets of capabilities for the robots. The circle formation problem is solvable in *ASYNC* model, when robots have unlimited visibility, and the solution requires no extra assumption on the capabilities of the robots. Under limited visibility model, different solutions have been proposed with different sets of assumptions. Sugihara and Suzuki proposed a heuristic algorithm to form a circle of given radius under limited visibility [14]. Dutta et al. solved the circle formation problem for robots represented as unit discs (*fat* robots) under limited visibility model [7]. *Uniform circle formation* is another variation of the circle formation problem in which robots are asked to place themselves on the boundary of a circle such that they are equally spaced from each other. Suzuki and Yamashita proposed an algorithm for uniform circle formation for non-oblivious robots [16]. Defago and Konogaya showed that in *SSYNC* model, it is possible to converge towards a uniform circle [5]. Flocchini et al. solved the uniform circle formation problem when system has $n \neq 4$ robots [9]. Mamino and Viglietta solved the problem for $n = 4$ robots [11]. Peleg was the first to proposed the idea of using externally visible lights [12]. Das et al. characterized the computational powers of the models in which robots have externally visible lights [3]. Flocchini et al. solved the *rendezvous* problem in two different setting: (a) the robots use the lights only for remembering its own internal state and (b) they use lights to communicate with other robots its current state [10]. In memory model, solutions for the *mutual visibility* problem were also proposed [6]. A solution for the circle formation problem in the obstructed visibility model (when robots are not transparent) was proposed in [6]. None of the works in the literature have considered the constrained version of the circle formation problem.

2 Our Contribution

This work presents a study of the *constrained circle formation* problem for a set of autonomous mobile robots. The contributions of this paper are in two folds. While the circle formation problem is solvable for an arbitrary set of asynchronous robots without any extra assumption, we have shown that the constrained circle formation problem for a set of asynchronous oblivious robots is not solvable even if robots have rigid movements and both axes agreements. A characterization of the set of robot configurations for which the problem is not solvable is presented. Then, we have presented a distributed algorithm to solve the problem in admissible configurations for asynchronous robots. The algorithm uses only one bit of persistent memory. The robots do not have any form of agreements in their coordinate axis systems or chirality or constrains in movement patterns. In this weak setting, we have solved the constrained circle formation problem for asynchronous robots which use only two colours starting from an admissible initial configurations. The solution ensures collision-less movements of the robots. To the best of our knowledge, this work is the first to study the constrained circle formation problem for asynchronous robots. One of the implications of the constrained version of circle formation problem is energy efficiency.

3 General Model and Definitions

The paper considers a set of autonomous, homogeneous, anonymous, asynchronous robots under the *ASYNC* (*CORDA*) model. The robots are considered as points in the infinite Euclidean plane. A robot can freely move on the plane. Each robot owns a local coordinate system centred at its current position. Two distinct robots may not have same directions and the orientations of the axes and unit distances. The directions and the orientations of the axes for a robot may change with positions. Furthermore, the robots do not share a common chirality. A robot uses its local coordinate systems to locate the positions of the other robots in the system. Initially, no two robots share same point. Each robot has unlimited visibility range. A robot has non-rigid movement in which it may be stopped by an adversary before reaching its destination. However, when it moves, it moves at least a distance δ towards its destination point if it does not reach its destination where $\delta > 0$ is a constant. This assumption ensures that a robot reaches its destination within finite time. It is assumed that the robots have no knowledge about the value of δ.

- **Configuration of the robots**: Let $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ be the set of n robots. Let $r_i(t)$ be the point occupied by r_i at time t. Let $\mathcal{R}(t) = \{r_1(t), \ldots, r_n(t)\}$ denote the robot configuration and $\widetilde{\mathcal{R}}$ be the set of all such configurations. It is assumed that in the initial configuration $\mathcal{R}(t_0)$, there is no multiplicity point (a point occupied by multiple robots).
- The closed-line segment between two points p and q includes these two points, and it is denoted by $\overline{pq}$. The open-line segment between p and q excludes these two points and is denoted by (p, q). Let $|p, q|$ denote the distance between two points p and q. Let $X \setminus Y$ denote the set difference of two sets X and Y. When measuring the angle between two line segments, we consider the angle which is has value than or equal to π.
- **Smallest enclosing circular annulus**: Let $SECA(t)$ denote the smallest enclosing circular annulus of the points in $\mathcal{R}(t)$ and $\mathcal{O}_t$ denote its centre. Let $C_{out}(t)$ and $C_{in}(t)$ denote the circles forming the outer and inner boundaries respectively of

$SECA(t)$. Let $C_{opt}(t)$ denote the circle which is equally distanced from $C_{out}(t)$ and $C_{in}(t)$ and the distance of $C_{opt}(t)$ from $C_{out}(t)$ and $C_{in}(t)$ is denoted by l_{opt}. The annular region between the circles $C_{out}(t)$ and $C_{in}(t)$ (excluding the circumferences of $C_{out}(t)$ and $C_{in}(t)$) is denoted by ANL. When there is no ambiguity, ANL, $C_{out}(t)$ and $C_{in}(t)$ are used to denote the sets of robots lying within the annular region ANL, on the circles $C_{out}(t)$ and $C_{in}(t)$), respectively. For each robot $r_i \in \mathcal{R}$, let $rad_i(t)$ denote the half line or starting from O_t and passing through $r_i(t)$.
- Let $S(t)$ denote one of the two sets $C_{out}(t)$ and $C_{in}(t)$ which contains more number of robots, i.e. $S(t) = arg\ max\{|C_{out}(t)|, |C_{in}(t)|\}$.
- **Different robot configurations**: We define the following sub-classes of $\widetilde{\mathcal{R}}$:
  - $\boldsymbol{\mathcal{E}}$: A configuration $\mathcal{R}(t)$ belongs to this class if $\exists\ r_i(t), r_j(t) \in \mathcal{R}(t)$ such that either (i) $r_i(t) \in \mathcal{C}_{out}(t)$ and $r_j(t) \in \mathcal{C}_{in}(t)$ and $rad_i(t) = rad_j(t)$ or (ii) $r_i(t) \in \mathcal{C}_{out}(t) \cup \mathcal{C}_{in}(t)$ and $r_j(t) \in \mathcal{C}_{opt}(t)$ and $rad_i(t) = rad_j(t)$.
  - $\boldsymbol{\mathcal{SR}}$: It contains all configurations $\mathcal{R}(t)$ which are rotationally symmetric and $|ANL| < 3$ for $\mathcal{R}(t)$.
  - $\boldsymbol{\mathcal{CL}}$: A configuration $\mathcal{R}(t)$ belongs to this class if all the robot positions in $\mathcal{R}(t)$ lie on a single line, i.e. all of them are collinear.
  - $\boldsymbol{\mathcal{M}}$: It contains all configurations $\mathcal{R}(t)$ which contain at least one multiplicity point.
  - $\boldsymbol{\mathcal{H}_{\leq 7}}$: A configuration $\mathcal{R}(t)$ is in this class if $|\mathcal{R}(t)| \leq 7$ and $|ANL| < 3$.
  - $\boldsymbol{\mathcal{U}}$: The class is defined by $\mathcal{U} = \mathcal{E} \cup \mathcal{SR} \cup \mathcal{CL} \cup \mathcal{M} \cup \mathcal{H}_{\leq 7}$.
  - $\boldsymbol{\widetilde{\mathcal{R}}_s}$: The class is defined by $\widetilde{\mathcal{R}}_s = \widetilde{\mathcal{R}} \backslash \mathcal{U}$.

To solve the constrained circle formation problem, we use the following results from pp. 163–167 of the textbook [1]:

Result 1 *For a configuration $\mathcal{R}(t)$, the smallest enclosing circular annulus $SECA(t)$ can be computed in polynomial time [1].*

Result 2 *For a nonlinear configuration $\mathcal{R}(t)$, the smallest enclosing circular annulus $SECA(t)$ has finite radius [1].*

Result 3 *For a nonlinear configuration $\mathcal{R}(t)$, the smallest enclosing circular annulus $SECA(t)$ has any one of the following properties: (i) $C_{out}(t)$ contains at least three points of $\mathcal{R}(t)$ and $C_{in}(t)$ contains at least one point of $\mathcal{R}(t)$ or (ii) $C_{out}(t)$ contains at least one point of $\mathcal{R}(t)$ and $C_{in}(t)$ contains at least three points of $\mathcal{R}(t)$ or (iii) both of $C_{out}(t)$ and $C_{in}(t)$ contains at least two points of $\mathcal{R}(t)$ [1].*

Observation 1 *The circle $C_{opt}(t)$ of a configuration $\mathcal{R}(t)$ uniquely minimizes the maximum distance from any point in $\mathcal{R}(t)$ to its circumference.*

The above observation implies that $C_{opt}(t_0)$ is the unique solution of the constrained circle formation problem for an initial configuration $\mathcal{R}(t_0)$.

Observation 2 *If a configuration $\mathcal{R}(t)$ has exactly one line of symmetry $\mathcal{L}_1$, one can define the positive direction $\mathcal{L}_1^+$ along $\mathcal{L}_1$.*

Following theorem is given without a proof:

Theorem 1 *For an initial configuration $\mathcal{R}(t_0) \in \mathcal{U}$, the constrained circle formation problem, in general, is not solvable, even if robots have persistent memory.*

4 Circle Formation Without Persistent Memory

This section presents a study of the constrained circle formation problem under a memoryless model. We provide a negative result in this setting.

Theorem 2 *The constrained circle formation problem for oblivious, asynchronous robots is deterministically unsolvable in the ASYNC model, even if robots have rigid motion.*

Proof If possible, let $\mathcal{A}$ be an algorithm which solves the constrained circle formation problem for oblivious robots. Consider an initial robot configuration $\mathcal{R}(t_0)$ as depicted in Fig. 1a. The circle $C_{opt}(t_0)$ is the desired one to be formed by the robots. All the robots should move along the line segments joining their current positions to $\mathcal{O}_{t_0}$. Now suppose that the robot r_i computes $p_i(t_0)$ and moves to this point (since scheduler is asynchronous adversary only chooses r_i for movement, the movements of the robots are rigid). This movement of r_i changes the configuration to $\mathcal{R}(t')$ as shown in Fig. 1b. Since the robots are oblivious, $\mathcal{A}$ would consider $\mathcal{R}(t')$ as a fresh initial configuration and instruct the robots to form $C_{opt}(t')$. This would cause all the robots to deviate from their original paths and would violate the optimization criteria. Hence, the theorem is true.

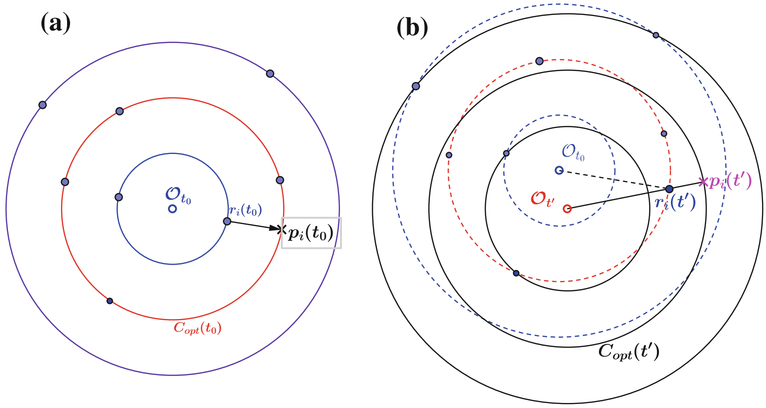

Fig. 1 An illustration of the counter-example in Theorem 2

5 Circle Formation with Persistent Memory

From observation 1, in an arbitrary initial robot configuration $\mathcal{R}(t_0) \in \widetilde{\mathcal{R}}_s$, circle $C_{opt}(t_0)$ is the unique solution for the *constrained circle formation* problem. When robots move, this circle may not remain invariant. We devise strategies in which the robots can recognize $C_{opt}(t_0)$, even if the configuration is changed. Each robot owns a single bit of persistent memory. This persistent memory is implemented via externally visible light which assumes two different colours to indicate two disjoint states. These colours do not change automatically (i.e. persistent). The lights are used with two objectives: one to store a robot's own state and other to broadcast its current state (both for communication and internal memory) [3]. A robot can identify colours of all the lights. Apart from the colour, all robots are oblivious, i.e. they do not carry any piece of information from previous cycles.

5.1 States of the Robots

Different colours of externally visible lights are used by the robots to indicate their states. Let $\mathcal{X}$ denote the set of these colours. The robots use two colours *off* and *on*, i.e. $\mathcal{X} = \{off, on\}$. The colour *on* indicates that a robot is in any one of the following states (i) active state and waiting for some other robots to turn their light *on* or to move (ii) the robot is on the circle $C_{opt}(t_0)$. The colour *off* indicates any one of the remaining states.

5.2 Algorithm Move()

Let r_i be a robot, and it wants to move to the circumference of $C_{opt}(t)$ such that the optimization criteria of the problem are also satisfied. Let $UP(t)$ be the annular region in between the circles $C_{out}(t)$ and $C_{opt}(t)$ (including the boundary of $C_{out}(t)$ and excluding the boundary of $C_{opt}(t)$) and $LOW(t)$ be the annular regions in between the circles $C_{opt}(t)$ and $C_{in}(t)$ (including the boundary of $C_{in}(t)$ and excluding the boundary of $C_{opt}(t)$). Let $C_i(t)$ denote the circle passing through $r_i(t)$ and having centre at $\mathcal{O}_t$. Let p_i be the point of intersection between $rad_i(t)$ and $C_{opt}(t)$. Let $u_i(t)$ be the intersection point between $rad_i(t)$ and $C_{out}(t)$. Let v_i be the intersection point between $rad_i(t)$ and $C_{in}(t)$. The corridor of r_i, denoted by $Cor_i(t)$, is defined as follows: (i) if $r_i(t)$ lies in $UP(t)$, then the corridor is the annular region between the circles $C_{opt}(t)$ and $C_i(t)$ (excluding the two boundaries) (Fig. 2a) and (ii) if $r_i(t)$ lies in $LOW(t)$, then the corridor is the annular region between the circles $C_{opt}(t)$ and $C_i(t)$ (including the boundary of $C_{opt}(t)$ and excluding the boundary of $C_i(t)$) (Fig. 2b). We say the $Cor_i(t)$ is free (i) for a robot in $UP(t)$ if $Cor_i(t)$ does not contain any robot position and (ii) for a robot r_i in LOW if all the robots in $Cor_i(t)$ lies on the

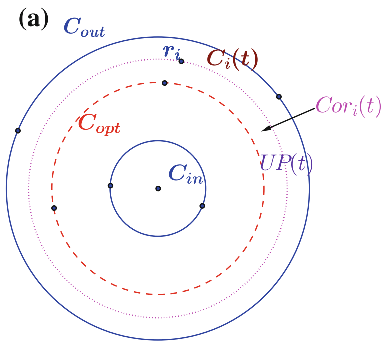

Fig. 2 An example of $cor_i(t)$: **a** r_i is in $UP(t)$ and $Cor_i(t)$ is free **b** r_i is in $LOW(t)$ and $Cor_i(t)$ is not free

circle $C_{opt}(t)$. Robot r_i moves towards the circumference of $C_{opt}(t)$ in the following way:

- ***Cor*$_i$*(t)* is free**: If there is no robot at p_i, robot r_i moves straight towards p_i along $rad_i(t)$. Otherwise, robot r_i moves to the destination point computed in the following way:
  (i) Suppose, r_j is the robot lying at p_i. Let $\{rad_k(t), rad_s(t)\}$ be the two adjacent rays to $rad_i(t)$. Without loss of generality, suppose, $\angle r_i(t)\mathcal{O}_t r_k(t) \geq \angle r_i(t)\mathcal{O}_t r_s(t)$.
  (ii) Let d and l be two distances defined as follows: (a) if $r_i(t)$ lies in $UP(t)$, then d and l are the two distances of $r_i(t)$ from $u_i(t)$ and p_i, respectively; (b) if $r_i(t)$ lies in $LOW(t)$, then d and l are the two distances of $r_i(t)$ from $v_i(t)$ and p_i, respectively. Let $h = l + \frac{d}{2^x}$, where x is the number of robots in $(r_i(t), u_i(t))$ if $r_i(t)$ lies in $UP(t)$ or $(r_i(t), v_i(t))$ if $r_i(t)$ lies in $LOW(t)$.
  (iii) Let $\hat{C}_i(t)$ be the circle having centre at $r_i(t)$ and radius h. Let m_i be the intersection point between $\mathcal{C}_{opt}(t)$ and $\hat{C}_i(t)$ such that m_i lies in the wedge defined by the angle $\angle r_i(t)\mathcal{O}_t r_k(t)$.
  (iv) Let $a_i(t)$ be the point on $C_{opt}(t)$ in the wedge defined by the angle $\angle r_i(t)\mathcal{O}_t r_k(t)$ such that $\angle r_i(t)\mathcal{O}_t a_i(t) = \frac{1}{3}\angle r_i(t)\mathcal{O}_t r_k(t)$. Let q_i denote the closest point among m_i and $a_i(t)$ from $p_i(t)$. The destination point of r_i is the middle point of the $arc(p_i(t), q_i)$ on the circle $\mathcal{C}_{opt}(t)$.
- ***Cor*$_i$*(t)* not is free**: Robot r_i does nothing.

5.3 Algorithm OptCircle()

It is assumed that (i) the initial configuration $\mathcal{R}(t_0) \in \widetilde{\mathcal{R}}_s$, (ii) each robot in the system has light *off* initially and (iii) $n \geq 8$. We use result 2, result 3 and observation 1 to solve the problem. The facts stated in the result 3 are used to make $C_{opt}(t_0)$ invariant under the movements of the robots until $C_{opt}(t_0)$ becomes recognizable by the robots.

For an initial configuration $R(t_0)$, if ANL contains more than 2 robots, then all the robots in ANL compute and move to the circumference of $C_{opt}(t_0)$. Once they are on the circle $C_{opt}(t_0)$, they turn their lights *on* to make the circle recognizable to the other robots not on $C_{opt}(t_0)$. Once at least three robots on $C_{opt}(t_0)$ turn their lights *on*, the other robots compute the circle passing through the robots having light *on*, i.e. $C_{opt}(t_0)$ and move towards the circumference of the circle. Otherwise, robots are selected from the two circles $C_{int}(t_0)$ and $C_{out}(t_0)$ and are moved within ANL in such way that within finite time ANL contains at least three robots. Since the robots are asynchronous and the number of persistent lights are only two, the main challenge lies in the selection and the movements of the robots so that (i) no forbidden configuration is created due to the movements of the robots before ANL contains at least three robots; (ii) no deadlock or livelock is created during the execution of the algorithm; (iii) robots do not collide during their movements; and (iv) the annulus of the initial configuration remains same.

Let r_i be an arbitrary robot in $\mathcal{R}$. If there are at least three co-circular robots with lights *on* in $\mathcal{R}(t)$, then we are done. Robot r_i computes the circle passing through the robots having lights *on*. If r_i does not lie on this circle, it moves towards the circumference of the circle without changing its light. Otherwise, r_i does nothing. Now suppose there are less than three robots on $C_{opt}(t_0)$ having lights *on*. Depending upon the current position and configuration, robot r_i performs any one of the following actions:

- $|ANL| \geq 3$: If r_i is in ANL and $r_i \notin \mathcal{C}_{opt}(t)$, robot r_i moves towards the circumference of $C_{opt}(t)$ according to algorithm $Move()$ and it does not change its light. If $r_i \in \mathcal{C}_{opt}(t)$, all the robots not lying on $C_{opt}(t)$ have light *off* and $C_{opt}(t)$ contains less than three robots with lights *on*, robot r_i turns its light *on* and does not move. In the rest of the cases, r_i does nothing.
- $|ANL| < 3$: The main strategy here is to select robots from $C_{in}(t) \cup C_{out}(t)$ and move them within ANL so that ANL contains at least three robots within finite time and the annulus $SECA(t)$ remains same during the process. The robots follow algorithm $Move()$ to reach their respective destination points.
  - $r_i \in ANL$: If $r_i \in \mathcal{C}_{opt}(t)$, robot r_i does nothing. Otherwise, it does not change its light and moves towards the circumference of $\mathcal{C}_{opt}(t)$.
  - $r_i \notin ANL$: In this case, robot r_i lies on the boundary of the annulus $SECA(t)$. Following are the possible scenarios:
    - $*|ANL| = 2$: If r_i has light *on* and there is another robot with light *on*, robot r_i moves towards $C_{opt}(t)$. Otherwise, robot r_i computes $S(t)$ and acts according to the followings:
      - · $\mathcal{R}(t)$ **is asymmetric**: Since $\mathcal{R}(t)$ is asymmetric, the robot positions in $\mathcal{R}(t)$ are orderable [2]. If $r_i \in S(t)$, there is no robot with light *on* and r_i has highest order among the robots in $S(t)$, and it moves towards $C_{opt}(t)$ without changing colour of its light.

        Otherwise, robot r_i does nothing.

· **$\mathcal{R}(t)$ has one line of symmetry**: Suppose $\mathcal{L}$ is the line of symmetry. Suppose, $r_i \in S(t)$. If r_i lies on $\mathcal{L}^+$, robot r_i moves towards $C_{opt}(t)$ without changing the colour of its light. If $\mathcal{L}$ does not pass through any robot in $S(t)$ and r_i is one of the closest robots to $\mathcal{L}^+$, robot r_i turns its light *on* and does not move. In rest of the cases, r_i does nothing.

$*$**$|ANL| = 1$**: Suppose there are two robots on $C_{out}(t) \cup C_{in}(t)$ with lights *on*. If r_i has light *on*, it moves towards $C_{opt}(t)$. Otherwise, it does nothing. If there are no two robots on $C_{out}(t) \cup C_{in}(t)$ with lights *on*, robot r_i computes $S(t)$ and acts according to the followings:

· **$\mathcal{R}(t)$ is asymmetric**: If $r_i \in S(t)$ and it has highest order or the second highest order among the robots in $S(t)$, it turns its light *on* and does not move. In the rest of the cases, robot r_i does nothing.

· **$\mathcal{R}(t)$ has one line of symmetry**: If $r_i \in S(t)$, r_i does not lie on $\mathcal{L}$ and it is one of the closest to $\mathcal{L}^+$, robot r_i turns its light *on* and does not move. Otherwise, r_i does nothing.

$*$**$|ANL| = 0$**: Robot r_i computes $S(t)$. First, suppose there are two robots with lights *on*. Let $A = \{r_j, r_k\}$ be the two robots with lights *on*. Let $S_2(t) = arg\ max\{|C_{out}(t) \backslash A|, |C_{in}(t) \backslash A|\}$. If r_i has light *off* and $r_i \in S_2(t)$, it does any one of the followings: (a) $\mathcal{R}(t)$ is asymmetric and r_i has highest order among the robots in $S_2(t)$, it moves towards $C_{opt}(t)$; (b) $\mathcal{R}(t)$ has one line of symmetry and r_i is one of the closest robots to $\mathcal{L}^+$, among the robots in $S_2(t)$, it moves towards $C_{opt}(t)$. In both the cases, r_i does not change its light. Otherwise, it does nothing. Next, suppose there is at most one robot with light *on*. If r_i has highest order or second highest order among the robots in $S(t)$, it changes its light to *on* and does not move. In rest of the cases, it does nothing.

5.4 *Correctness of OptCircle*()

We prove that *OptCircle*() solves the constrained circle formation problem within finite time.

Lemma 1 *Algorithm Move() provides collision-free robot movements during the execution of OptCircle().*

Proof During the execution of *Move*(), robots first order themselves and then move towards $C_{opt}(t_0)$ according to that order. Thus, the robots lying on the same ray do not collide. The destination of a robot r_i lies on the $\frac{1}{3}$ section of the wedge defined by the larger angle with the neighbouring $rad_i(t)$. Thus, two robots on two different rays also do not collide. This implies that the robots have collision-free movements. Also, the destination point of a robot r_i lies within the circle having radius l_{opt} and centre at $r_i(t)$. This implies that the optimization criteria of the circle formation problem is satisfied by the movements of the robots. □

Lemma 2 *Suppose, in a configuration* $\mathcal{R}(t) \in \widetilde{R}_s$, $t \geq t_0$, $|ANL| < 3$. *During the execution of OptCircle(), there exists a time* $t' \geq t$ *such that* $|ANL| \geq 3$ *in the configuration* $\mathcal{R}(t')$ *and the circle* $C_{opt}(t')$ *is same as* $C_{opt}(t)$.

Proof Consider a configuration $\mathcal{R}(t)$ with $|ANL| < 3$. We prove the lemma analysing each case separately. Note that if the annulus remains same during the movements of the robots, so does $C_{opt}(t)$.

- $|ANL| = 2$: In this case, at least one and at most two robots from $S(t)$ move inside the annulus $SECA(t)$. Thus, within finite time, ANL contains at least three robots. Since $n \geq 8$ and $|ANL| = 2$, the set $S(t)$ contains at least three robots. Thus by result 3, the removal at most two robots from $S(t)$ does not change the annulus $SECA(t)$.
- $|ANL| = 1$: In this case at least two and at most three robots from $S(t) \cup S_2(t)$ move within ANL. This makes $ANL| \geq 3$, in finite time. Since $n \geq 8$ and $|ANL| = 1$, the set $S(t)$ contains at least four robots. At most, two robots from the set $S(t)$ are removed. Thus, the result of 3 implies that $C_{opt}(t)$ does not change, during the movements of these robots.
- $|ANL| = 0$: In this case, at least three and at most four robots from $S(t) \cup S_2(t)$ are selected and moved inside $SECA(t)$. This makes $ANL| \geq 3$, in finite time. Since $n \geq 8$ and $|ANL| = 0$, each of the sets $S(t)$ and $S_2(t)$ contains at least four robots. By the same arguments as above, the circle $C_{opt}(t)$ remains intact during the movements of these robots.

Hence, the lemma is true. □

Lemma 3 *Suppose, in a configuration* $\mathcal{R}(t) \in \widetilde{R}_s$, $t \geq t_0$, $|ANL| \geq 3$. *During the execution of OptCircle(), there exists a time* $t' \geq t$ *such that* $C_{opt}(t)$ *contains at least three robots with lights on, in the configuration* $\mathcal{R}(t')$. *Furthermore,* $C_{opt}(t)$ *is the unique circle in* $\mathcal{R}(t')$ *containing at least three robots on its circumference with lights on.*

Proof Let r_i be a robot in ANL in the configuration $\mathcal{R}(t)$. When r_i becomes active, if it finds itself on $C_{opt}(t)$, it takes any one of the following decisions (i) there is at least one robot not on $C_{opt}(t)$ with light *on* or there are at least three robots on $C_{opt}(t)$ with robot light *on*, robot r_i does nothing until the robots having lights *on* reach $C_{opt}(t)$, and (ii) all the robots, not lying on $C_{opt}(t)$, have lights *off* and $C_{opt}(t)$ contains less than three robots with lights *on*, robot r_i turns its light *on*. If r_i is not on $C_{opt}(t)$, it moves towards $C_{opt}(t)$. Thus, if $C_{opt}(t)$ contains less than three robots, within finite time, it will have at least three robots on its boundary. This implies that there exists a time $t' \geq t$ such that $C_{opt}(t)$ contains at least three robots with lights *on*, in $\mathcal{R}(t')$. The second part of the lemma follows from the case (i) and case (ii) above. Hence, the lemma is true. □

Lemma 4 *Given an initial configuration* $\mathcal{R}(t_0) \in \widetilde{R}_s$, *algorithm OptCircle() solves the constrained circle formation problem for a set of asynchronous robots.*

Proof By Lemma 2 and 3, there exists $t > t_0$ such that in $C_{opt}(t_0)$ is the unique circle in $\mathcal{R}(t)$ containing at least three robots on its circumference with lights *on*. Once this is done, the circle $C_{opt}(t_0)$ is uniquely recognizable by the robots even when all the robots move towards it. The robots simply compute the circle which contains at least three robots with lights *on* and then move towards it using algorithm $Move()$. Algorithm $Move()$ assures collision-free robot movements. Also, during the movements, the robots satisfy the optimization criteria of the problem. Thus, within finite time all the robots reach $C_{opt}(t_0)$ satisfying the optimization criteria. Hence, the lemma is true. □

From above results, we have the following theorem.

Theorem 3 *The constrained circle formation problem is solvable in the ASYNC model for an initial configuration $\mathcal{R}(t_0) \in \widetilde{R}_s$, when robots have externally visible lights with only 2 distinct colours.*

6 Conclusion

This paper presents a study of the constrained circle formation problem for asynchronous autonomous mobile robots. For oblivious robots, it is proved that the problem is not solvable under *ASYNC* model even when the robots have rigid movements. For robots having persistent memory, the initial robot configurations, which the problem is not solvable, are identified. For rest of the configurations, an algorithm is proposed which solves the problem for asynchronous robots which have exactly one bit of persistent memory. Following are the possible future directions of the problem: (i) relaxation of the exact optimality in the constrain considered in this work; (ii) study of the problem when robots develop faults; and (iii) the extension of the problem to the three-dimensional Euclidean space.

References

1. Berg, M., Cheong, O., Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer, Santa Clara, USA (2008). TELOS
2. Chaudhuri, G.S., Mukhopadhyaya, K.: Leader election and gathering for asynchronous fat robots without common chirality. J. Discrete Algorithms **33**, 171–192 (2015)
3. Das, S., Flocchini, P., Prencipe, G., Santoro, N., Yamashita, M.: The power of lights: synchronizing asynchronous robots using visible bits. In: IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), pp. 506–515 (2012)
4. Défago, X., Gradinariu, M., Messika, S., Raipin-Parvédy, P.: Fault-tolerant and self-stabilizing mobile robots gathering. In: Proceeding of 20th International Symposium on Distributed Computing, pp. 46–60 (2006)
5. Défago, X., Konagaya, A.: Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In: Proceedings of the Second ACM International Workshop on Principles of Mobile Computing, POMC 2002, pp. 97–104. ACM, New York, USA (2002)

6. Di Luna, G.A., Flocchini, P., Chaudhuri, S.G., Santoro, N., Viglietta, G.: Robots with lights: overcoming obstructed visibility without colliding. In: Proceeding of 16th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2014), pp. 150–164 (2014)
7. Dutta, A., Chaudhuri, S.G., Datta, S., Mukhopadhyaya, K.: Circle formation by asynchronous fat robots with limited visibility. In: Proceedings of International Conference on Distributed Computing and Internet Technology (ICDCIT), pp. 83–93 (2012)
8. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers (2012)
9. Flocchini, P., Prencipe, G., Santoro, N., Viglietta, G.: Distributed computing by mobile robots: solving the uniform circle formation problem. In: The 18th International Conference on Principles of Distributed Systems (OPODIS 2014), pp. 217–232 (2014)
10. Flocchini, P., Santoro, N., Viglietta, G., Yamashita, M.: Rendezvous of two robots with constant memory. In: Proceeding of International Colloquium on Structural Information and Communication, Complexity, pp. 189–200 (2013)
11. Mamino, M., Viglietta, G.: Square formation by asynchronous oblivious robots. In: Proceedings of the 28th Canadian Conference on Computational Geometry (CCCG), pp. 1–6 (2016)
12. Peleg, D.: Distributed coordination algorithms for mobile robot swarms: new directions and challenges. In: Distributed Computing—IWDC 2005. Lecture Notes in Computer Science, vol. 3741, pp. 1–12. Springer, Berlin (2005)
13. Prencipe, G.: Instantaneous actions versus full asynchronicity: controlling and coordinating a set of autonomous mobile robots. In: Proceeding of 7th Italian Conference on Theoretical Computer Science, pp. 154–171 (2001)
14. Sugihara, K., Suzuki, I.: Distributed motion coordination of multiple mobile robots. In: Proceedings of IEEE International Symposium on Intelligent Control, pp. 138–143 (1990)
15. Suzuki, I., Yamashita, M.: Formation and agreement problems for anonymous mobile robots. In: Proceeding of 31st Annual Conference on Communication, Control and Computing, pp. 93–102 (1993)
16. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: formation of geometric patterns. SIAM J. Comput. **28**, 1347–1363 (1999)