Abstract

Arbitrary pattern formation (APF) by mobile robots is studied by many in literature under different conditions and environment. Recently it has been studied on an infinite grid network but with full visibility. In opaque robot model, circle formation on infinite grid has also been studied. In this paper, we are solving APF on infinite grid with asynchronous opaque robots with lights. The robots do not share any global co-ordinate system. The main challenge in this problem is to elect a leader to agree upon a global co-ordinate where the vision of the robots are obstructed by other robots. Since the robots are on a grid, their movements are also restricted to avoid collisions. In this paper, the aforementioned hardness are overcome to produce an algorithm that solves the problem.

Keywords

Distributed Algorithm, Arbitrary Pattern Formation, Compact Line Formation, Opaque Robots, Autonomous Robots, Luminous Robots, Asynchronous, Look-Compute-Move Cycle, Infinite Grid.
1. Introduction

Arbitrary Pattern Formation (APF) Problem is a classical problem in swarm robotics which deals with fundamental coordination problem of multi-robot systems. The problem is to design an algorithm which will be used by each autonomous mobile robot of a robot swarm that will guide the robots to form any specific pattern that is given to the robots initially as input. In this problem, the robots are modeled as autonomous (there is no central control), anonymous (the robots do not have any unique identifiers), homogeneous (all robots execute the same algorithm) and identical (robots are indistinguishable by their appearance). All the robots can freely move on the plane. Each robot has sensing capability by which they can perceive the location of other robots on the plane. The robots do not have any global coordinate system (each robot has its own local coordinate system) and they operate in Look-Compute-Move (LCM) cycles. In the Look phase, a robot takes a snapshot of its surroundings. In the Compute phase, a robot process the information got from the Look phase and in the Move phase a robot moves to another position (a robot also might stay still in this phase) depending on the output of the Compute phase.

1.1. Earlier Works

The problem of Arbitrary Pattern Formation was first introduced in [16] and after that this problem has been studied many times in the literature ([2, 4–14, 17–19]). Initially the problem has been studied only assuming that the robots do not have obstructed visibility. But in a more practical setting, when more than two robots are in a straight line, a robot with camera sensors can only see its adjacent robots (at most two robots). These robots are known as opaque robots. In [4], APF has been solved considering opaque robots with visible lights which can assume 6 persistent colors. They have also assumed one axis agreement for each robots. This model of luminous robots has first been introduced in [15] by Peleg et al. The visible lights can be used by robots as a means of communication and persistent memory. In [4], the robots are considered to be point robots. But in real life scenario, robots are physical entities that have certain dimensions. So in [2], the problem of APF has been considered and solved using fat robots with luminous robots having 10 colors.

In a plane, the robots can move freely in any direction. So collision can be avoided by the robots by comparably easy techniques. So solving APF in specific network (eg. grid) is quite interesting itself where it is not so easy to avoid collisions. In [3], the authors have considered this problem and produced an algorithm with robots having full visibility on an infinite grid in OBLLOT model. Furthermore, in obstructed visibility model, a problem of circle formation on an infinite grid has been solved in [1] with opaque luminous robots with 7 colors. In this paper, we are considering the problem of arbitrary pattern formation on an infinite grid with luminous robots with obstructed visibility.
1.2. Problem description and our contribution

This paper deals with the arbitrary pattern formation problem on an infinite grid using opaque luminous robots with 8 colors. The robots operate in LCM cycles under an adversarial asynchronous scheduler. The robots are autonomous, anonymous, identical and homogeneous. They move only through the edges of the grids and the movement is instantaneous for each robot (i.e a robot can only be seen on a grid point). Initially the robots are placed arbitrarily on the grid. From this configuration, they need to move to a target configuration or, Pattern (a set of target coordinates) without collision. The robots have one axis agreement and does not have agreement on global coordinate (each robot has its own local coordinate).

The main difficulty of the problem is visibility. As APF is closely related to the Leader Election problem, without seeing the whole configuration it is quite hard to elect a leader and thus design an algorithm to solve the problem. Depending on the local view of each robot, the algorithm is need to be designed. In this paper, the described algorithm does so. Also another difficulty was to avoid collision between robots while they are moving. We removed this difficulty by using a technique where the robots will move sequentially.

The problem described in this paper is also very practical in nature. The restricted movement and also the obstructed visibility, these practical scenarios are considered here. The algorithm described in this paper solves the above mentioned APF problem in total $O(kD)$ moves in the worst case, where $k$ is the number of robots on the grid and $D$ is $\max\{m, n, M, N, k\}$ ($m, n$ are the height and width of smallest enclosing rectangle of the initial configuration; $M, N$ are the same for the target configuration).

2. Model and Definitions

2.1. Model

Robots: Robots are autonomous, anonymous, homogeneous and identical. They are deployed on a two-dimensional infinite grid where each of them is initially positioned on distinct grid points. They do not have a common notion of direction. The robots have an agreement over the positive direction of X-axis i.e, all the robots have an agreement over left and right. They do not have any agreement over the Y-axis. Here the robots do not have access to any global coordinate system other than the agreement over the positive direction of X-axis. The total number of robots is not known to them. The robots are assumed to be dimensionless and modeled as points.

Look-Compute-Move cycles: The robot, when active, operates according to the Look-Compute-Move cycle. In the Look phase, a robot takes the snapshot of the positions of all the robots represented in its own local co-ordinate system. Then the robot performs computation and compute the next position and a light according to a deterministic algorithm i.e., the Compute phase. In the Move phase, it will
either move unit length to the desired location along a straight line or make a null move.

**Scheduler:** We assume that the robots are controlled by an asynchronous adversarial scheduler. This implies that the amount of time spent in LOOK, COMPUTE, MOVE, and inactive states by different robots is finite but unbounded and unpredictable. As a result, the robots do not have a common notion of time, and the configuration perceived by a robot during the LOOK phase may significantly change before it actually makes a move.

**Movement:** The movement of robots are restricted only along grid lines from one grid point to one of its four neighboring grid points. Robots’ movements are assumed to be instantaneous in discrete domains. Here we assume that the movements are instantaneous. The robots are always seen on grid points, not on edges.

**Visibility:** The robots visibility is unlimited but by the presence of other robots it can be obstructed. A robot $r_i$ can see another robot $r_j$ if and only if there are no robots on the straight line segment $r_ir_j$.

**Lights:** Each robot is equipped with an externally visible light, which can assume a $O(1)$ number of predefined lights. The robots communicate with each other using these lights. The lights are not deleted at the end of a cycle, but otherwise, the robots are oblivious. The lights used in our algorithm are \{off, terminal1, symmetric, decider, call, leader1, leader, done\}.

### 2.2. Notations and Definitions

We have used some notations throughout the paper. A list of these notations along with their definitions are mentioned in the following table.

| Symbol | Description |
|--------|-------------|
| $L_1$  | First vertical line on left that contains at least one robot. |
| $L_V(r)$ | The vertical line on which the robot $r$ is located. |
| $L_H(r)$ | The horizontal line on which the robot $r$ is located. |
| $L_I(r)$ | The left immediate vertical line of robot $r$ which has at least one robot on it. |
| $R_I(r)$ | The right immediate vertical line of robot $r$ which has at least one robot on it. |
| $H_O^L(r)$ | Left open half for the robot $r$. |
| $H_C^L(r)$ | Left closed half for the robot $r$ (i.e $H_O^L(r) \cup L_V(r)$). |
| $H_O^B(r)$ | Bottom open half for the robot $r$. |
| $H_C^B(r)$ | Bottom closed half for the robot $r$ (i.e $H_O^B(r) \cup L_H(r)$). |
| $H_O^U(r)$ | Upper open half for the robot $r$. |
| $H_C^U(r)$ | Upper closed half for the robot $r$ (i.e $H_O^U(r) \cup L_H(r)$). |
| $L_{t_{j-1}}$ | The horizontal line below the target position $t_j$. |
| $K$ | The horizontal line passing through the middle point of the line segment between two robots with light decider or terminal1 on the same vertical line. |
| $L_{H1}$ | The immediate horizontal line above the robot with light leader. |
Some additional definitions are needed to be explained which will be useful later.

**Configuration:** Let us consider a team of robots placed on an simple undirected connected graph \( G = (V, E) \). Let us define a function \( f : V \to \{0\} \cup \mathbb{N} \), where \( f(v) \) is the number of robots placed on vertex \( v \). The graph \( G \) together with the function \( f \) is called a configuration which is denoted by \( C = (G, f) \). For any time \( T \), \( C(T) \) will denote the configuration of the robots at time \( T \).

For a graph \( G = (V, E) \), \( \phi : V \to V \) is an automorphism if \( \phi \) is a bijection and \( \phi(u)\phi(v) \) is adjacent iff \( u \) and \( v \) are adjacent \( \forall u, v \in V \). All the automorphisms of \( G \) form a group denoted by \( \text{Aut}(G) \). Similarly we can define an automorphism \( \phi \) for a configuration \( (G, f) \) where \( \phi \in \text{Aut}(G) \) and \( f(u) = f(\phi(u)), \forall u \in V \). All automorphisms on \( (G, f) \) form a group denoted by \( \text{Aut}(G, f) \).

**Symmetric configuration:** For any configuration \( C = (G, f) \), we can define the group \( \text{Aut}(C) \). \( \phi(v) = v, \forall v \in V \) is called a trivial symmetry. Every non trivial \( \phi \in \text{Aut}(C) \) is called a symmetry of \( C \). Note that all symmetric configurations of a configuration \( C \) is basically generated by some translations, rotations and reflections. Translation shifts all the vertices by the same amount. Since the number of robots in the configuration \( C \) is finite it is easy to see that there is no translation in \( \text{Aut}(C) \). Reflections are defined by some axis or line of reflection. It can be vertical, horizontal or diagonal. The angle of rotation can be \( 90^\circ \) or \( 180^\circ \). The center of rotation can be a vertex of the grid, center of an unit square or a center of an edge.

**Stable Configuration:** A configuration \( C \) is called a stable configuration if the following conditions are satisfied in \( C \).

1. There are two robots with light **decider** on same vertical line and all other robots in \( C \) have light **off**.
2. The vertical and horizontal line on which the robots with light **decider** are located don’t have any other robots.
3. The robots with light **decider** have no robots on left open half and also their upper closed half or bottom closed half have no other robots.

**Leader Configuration:** A configuration \( C \) is called a leader configuration if the following conditions are satisfied in \( C \).

1. There are exactly one robot with light **leader** and all other robots have light **off**.
2. The vertical line and the horizontal line on which the robot with light **leader** is located do not have other robots.
3. The robots with light **leader** has no robots on left open half and also upper open half or bottom open half is empty.

**Compact Line:** A line is called compact if there is no unoccupied grid position between any two robots on that line.

**Terminal Robot:** A robot \( r \) is called a terminal robot if \( L_V(r) \cap H \) is empty, where \( H \in \{H_O^U(r), H_O^B(r)\} \).
Symmetry of a vertical line $L$ w.r.t $K$: Let $L$ be a vertical line of the grid and $\lambda$ be a binary sequence defined on $L$ such that $j$-th term of $\lambda$ is defined as follows:

$$\lambda(j) = \begin{cases} 1 & \text{if } \exists \text{ a robot on the } j\text{-th grid point from } K \cap L \text{ on the line } L. \\ 0 & \text{otherwise.} \end{cases}$$

By definition of $\lambda$, it follows that there are two such values of $\lambda$, say $\lambda_1$ and $\lambda_2$. We say that the line $L$ is symmetric with respect to $K$ if $\lambda_1 = \lambda_2$. For future, whenever symmetry of a line is mentioned, it is assumed that it means the symmetry of the line with respect to $K$.

**Dominant half:** A robot $r$ is said to be on the dominant half if the following conditions are satisfied:

1. $R_I(r)$ is not symmetric with respect to $K$.
2. If lexicographically $\lambda_1 > \lambda_2$ on $R_I(r)$, then $r$ and the portion of $R_I(r)$ corresponding to $\lambda_1$ lie on same half plane delimited by $K$.

3. The Algorithm

The main result of the paper is Theorem 1. The proof of the ‘only if’ part is the same as in case for point robots, proved in [4]. The ‘if’ part will follow from the algorithm presented in this section.

**Theorem 1.** For a set of opaque luminous robots having one axis agreement, APF is deterministically solvable if and only if the initial configuration is not symmetric with respect to a line $K$ such that 1) $K$ is parallel to the agreed axis and 2) $K$ is not passing through any robot.

For the rest of the paper, we shall assume that the initial configuration $C(0)$ does not admit the unsolvable symmetry stated in Theorem 1. Our algorithm works in two stages. In the first stage, Leader Election, the robots will agree on a leader. Since there are no common agreement on a global coordinate system, the robots will not be able to agree on the embedding of the pattern on the grid. Thus leader election is necessary for robots to agree on a global coordinate. This stage is further divided in two phases namely Phase 1 and Phase 2. Next in the pattern formation stage, the robots will move to form the input pattern embedded on the grid. The stages are described in details in 3.1 and 3.2.

3.1. Leader Election

3.1.1. Phase 1

The procedure Phase 1 starts from the initial configuration with the aim of forming a stable configuration or achieving a configuration with a robot with light leader. Initially all the robots are located on an infinite grid with light off. On waking, each robot $r$ will check if the robot is terminal and if their open left half has no other robots and no robot leader in $R_I(r)$. Note that there will be at most two and at
Algorithm 1: Phase 1

Procedure Phase1()

1. $r \leftarrow$ myself
2. if $r$’s light = off then
3. if $r$ is terminal and there is no robot in $H_1^O(r)$ and no robot leader1 in $R_1(r)$ then
4. $r$’s light $\leftarrow$ terminal1
5. Move left
6. else if there are exactly two robots in $L_1(r)$ having light terminal1 and $r$ is on $K$ then
7. $r$’s light $\leftarrow$ leader1
8. else if $r$’s light = terminal1 then
9. if no robots on $K \cap R_1(r)$ then
10. $r$’s light $\leftarrow$ symmetric
11. else if $r$ is in dominant half then
12. $r$’s light $\leftarrow$ leader1
13. else if there is a robot on $K \cap R_1(r)$ with light leader1 then
14. $r$’s light $\leftarrow$ off
15. else if $r$ is singleton on $L_1(r)$ and all robots in $R_1(r)$ are off then
16. $r$’s light $\leftarrow$ leader1
17. else if $r$’s light = symmetric then
18. if there is a robot with light symmetric on $L_1(r)$ then
19. $r$’s light $\leftarrow$ symmetric
20. else if there is a robot with light leader1 or off on $L_1(r)$ then
21. $r$’s light $\leftarrow$ off
22. else if $r$ is on $K \cap R_1(r)$ with light leader1 then
23. $r$’s light $\leftarrow$ leader1
24. else if there is a robot with light symmetric or decider on $L_1(r)$ then
25. if there is other robot both in $H_1^O(r)$ and $H_1^L(r)$ then
26. move vertically opposite to $r'$
27. else
28. $r$’s light $\leftarrow$ decider

least one such robot. These robots will change their lights to terminal1 and move left (Figure 1).

If there is only one robot (say, $r$) with light terminal1 on $L_1$ and light of all robots on $R_1(r)$ are off, then $r$ will change its light to leader1 (Figure 2, 3). This might also happen due to asynchrony of the system that there are two robots with terminal1 light but one (say, $r_1$) in $L_1$ and the other one (say, $r_2$) is still in $R_1(r_1)$. In this case, the robot on $L_1$ will see that all robots on $R_1(r_1)$ do not have lights off and will wait for $r_2$ to reach $L_1$.

If there are two robots (say, $r_1$ and $r_2$) on $L_1$ with terminal1 light, then observe that both the robots $r_1$ and $r_2$ and all the robots on $R_1(r_1)$ ($= R_1(r_2)$) can recognise the line $K$. Now, if there exists a robot (say $r_1$) occupying the grid point $R_1(r_1) \cap K$, 2022/05/09; 02:03 str. 7/26
then $r_1$ changes its light to leader1 and the robots with light terminal1 changes their light to off after seeing the robot $r_2$ with light leader1 (Figure 4, 5). If the grid point $R_I(r_1) \cap K$ is empty, then the robots $r_1$ and $r_2$ check the symmetry of the line $R_I(r_1) =$ $R_I(r_2)$. If $R_I(r_1)$ is not symmetric, then the robot $r_i$ ($i = 1$ or, 2), which is on the dominant half changes its light to leader1 (Figure 6, 7). On the other hand if $R_I(r_1)$ is symmetric, then both the robots $r_1$ and $r_2$ change their lights from terminal1 to symmetric (Figure 8, 9). Note that, due to asynchrony it might happen that one robot, let’s say $r_1$ changes its light to symmetric before $r_2$. Then $r_1$ does not move until it sees another robot ($r_2$) on $L_V(r_1)$ with light symmetric. This technique prevents the configuration from getting symmetric in this phase. Now after seeing another robot ($r_2$) on $L_V(r_1)$ with symmetric light, $r_1$ moves vertically opposite of $r_2$ until the closed upper half or the closed bottom half of $r_1$ has no other robots (similar argument can be given for $r_2$) (Figure 10). After reaching their designated positions both the robots $r_1$ and $r_2$ change their light from symmetric to decider, achieving a stable configuration (Figure 11). Note that, due to asynchrony it might happen that one robot, let’s say $r_1$ reaches to its designated position and changes its light to decider before $r_2$.

The following Lemmas 1, 2 and 3 and Theorem 1 justify the correctness of the Algorithm 1.

**Lemma 1.** If the initial Configuration $C(0)$ has exactly one robot on $L_1$, then $\exists T_1 > 0$ such that there will be exactly one robot with light leader1 in $C(T_1)$.

**Proof.** Let us assume the initial configuration $C(0)$ has exactly one robot (say, $r$) on $L_1$. According to the Algorithm 1, when $r$ wakes it will see that it is a terminal robot, the open left half is empty and no leader1 in $R_I(r)$. Then it changes the light to terminal1 and moves left. Note that, after $r$ moves left, $L_1$ now denotes the new vertical line where $r$ is located. Now on waking again, $r$ sees that it is the only robot on $L_1$ with light terminal1 and all other robots on $R_I(r)$ having light off. In this
case, \( r \) changes the light to \texttt{leader1}. Note that all the other robots in this case have the light \texttt{off} throughout the Algorithm 1. So during the execution of Algorithm 1, if this case occurs, there will be exactly one robot with light \texttt{leader1} (Figure 2, 3).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image1.png}
\caption{Single robot on \( \mathcal{L}_1 \) with light \texttt{terminal1}.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image2.png}
\caption{The single robot with light \texttt{terminal1} changes its light to \texttt{leader1}.}
\end{figure}

\textbf{Lemma 2.} If the initial configuration \( \mathcal{C}(0) \) has exactly two robots on \( \mathcal{L}_1 \), then \( \exists T' > 0 \) such that \( \mathcal{C}(T') \) is either a stable configuration or there is exactly one robot with light \texttt{leader1} in \( \mathcal{C}(T') \).

\textit{Proof.} Let \( r_1 \) and \( r_2 \) be two robots on \( \mathcal{L}_1 \) in the initial configuration \( \mathcal{C}(0) \). In this case, both the robots change their lights to \texttt{terminal1} and move left. Note that after moving left, \( \mathcal{L}_1 \) now denotes the vertical line on which the robots are located now. Due to asynchrony of the system, two cases may occur:

\textbf{Case 1:} Both the robots \( r_1 \) and \( r_2 \) reaches \( \mathcal{L}_1 \) before waking again. In this case, \( r_1 \) and \( r_2 \) can see each other and can calculate the line \( K \). Now if there is another robot (say \( r_3 \)) on the intersection of \( K \) and \( \mathcal{R}_I(r_1) \) (\( \mathcal{R}_I(r_1) = \mathcal{R}_I(r_2) \)), then \( r_1 \) changes the light to \texttt{leader1}. Note that \( r_1 \) can see both \( r_1 \) and \( r_2 \), so it can calculate the line \( K \) itself. After seeing \( r_1 \) with light \texttt{leader1}, the robots with light \texttt{terminal1} change their lights to \texttt{off} (Figure 4, 5).

On the other hand, if there is no robot on the intersection of \( K \) and \( \mathcal{R}_I(r_1) \), then \( r_1 \) and \( r_2 \) both check the symmetry of \( \mathcal{R}_I(r_1) \) with respect to the line \( K \). If \( \mathcal{R}_I(r_1) \) is asymmetric with respect to \( K \), then the robot in the dominant half (\( r_1 \) or \( r_2 \)) will change it’s light to \texttt{leader1} (Figure 6, 7).

Otherwise if \( \mathcal{R}_I(r_1) \) is symmetric with respect to \( K \), then \( r_1 \) and \( r_2 \) change their light to \texttt{symmetric} (Figure 8, 9). Note that due to asynchrony, it may happen that one robot (say, \( r_1 \)) changes light to \texttt{symmetric} while \( r_2 \) still has the light \texttt{terminal1}. In this case, \( r_1 \) will not move until \( r_2 \) wakes and changes it’s light to \texttt{symmetric}, after
Figure 4. Both the robots with light \textit{terminal1} see a robot on the right next occupied vertical line and on the line $K$.

Figure 5. The robot on $K$ changes its light to \textit{leader1} and next the \textit{terminal1} robots see robot \textit{leader1} and change their lights to \textit{off}.

Figure 6. The robots with light \textit{terminal1} see that the right next occupied vertical line is not symmetric.

Figure 7. The robot with light \textit{terminal1} on the dominant half changes the light to \textit{leader1}.

seeing the \textit{symmetric} light of $r_1$. After both the robots $r_1$ and $r_2$ changed their lights to \textit{symmetric}, they move vertically in the opposite direction of each other until one of the region $H_{U1}$ or $H_{B1}$ has no other robots (Figure 10). After this step, both $r_1$ and $r_2$ change their lights from \textit{symmetric} to \textit{decider}, thus reaching a stable configuration (Figure 11). Note that the robots will not check the symmetry of $\mathcal{R}_I(r_1)(=\mathcal{R}_I(r_2))$ if the light of robots are \textit{symmetric}.

Case 2: Let’s assume $r_1$ is on $L_1$ and awake before another robot $r_2$ with \textit{terminal1} light is yet to reach on $L_1$. In this case, $r_1$ checks $\mathcal{R}_I(r_1)$ and finds out all the robots except one ($r_2$ with light \textit{terminal1}) have light \textit{off}. In this case,
Figure 8. The terminal1 robots see that the right next occupied vertical line is symmetric.

Figure 9. Robots with light terminal1 change their lights to symmetry.

Figure 10. The robots with light symmetry move opposite of each other until they find either bottom or upper closed half has no other robots.

Figure 11. After reaching designated positions where upper closed or bottom closed half has no other robots, the robots with light symmetry change lights to decider to reach a stable configuration.

$r_1$ waits until $r_2$ reaches $L_1$ and then by similar argument like in case 1, either reaches a configuration with a robot having light leader1 or reaches a stable configuration.

So after the execution of Algorithm 1, the configuration is either stable or has exactly one robot with light leader1.

Lemma 3. If the initial configuration $C(0)$ has more than two robots on $L_1$, then $\exists T'' > 0$ such that $C(T'')$ is either a stable configuration or there is exactly one robot with light leader1 in $C(T'')$.

Proof. Let us assume there are $h$ robots on the line $L_1$, denoted by $\{r_i : i \in [1, h] \cap \mathbb{N} \text{ and } h > 2\}$. Suppose $r_1$ and $r_2$ are the two terminal robots. Since $r_1$ and $r_2$ can
identify themselves as terminal robots and can see there is no robot in $H^O_L(r)$ and no robot $\text{leader}_1$ in $R_I(r)$, they will change their light to $\text{terminal}_1$ and move left. Note that after $r_1$ or $r_2$ move left, $L_1$ now denotes the vertical line on which the robots are located now.

Now, with the same argument as in both the cases of Lemma 2, we can easily say that after the execution of Algorithm 1, the configuration is either stable or has exactly one robot with light $\text{leader}_1$.

Theorem 2. For any initial configuration $C(0)$, $\exists T > 0$ such that either $C(T)$ is a stable configuration or has exactly one robot with light $\text{leader}_1$.

Proof. This follows directly from Lemma 1, Lemma 2 and Lemma 3.

3.1.2. Phase 2

After completion of Phase 1, two configurations may occur. In the first possible configuration, there exists exactly one robot with light $\text{leader}_1$ and other robots with light $\text{off}$. The other possible configuration is a stable configuration. After completion of Phase 2, these configurations transforms into a leader configuration.

First, let us assume that after Phase 1, the configuration transforms into the stable configuration. In this configuration, there are two robots with light $\text{decider}$ such that their $H^C_L \cap H^C_H$ or $H^O_L \cap H^O_H$ have no other robots. All the other robots have light $\text{off}$. In Phase 2, a robot with light $\text{off}$ (say, $r$) checks whether it can see two robots with light $\text{decider}$ on $L_I(r)$. If $r$ can see two such robots on $L_I(r)$ and $r$ is on $K \cap L_V(r)$, then $r$ changes its light to $\text{leader}_1$. Note that $r$ can calculate the line $K$ as it can see both the robots having light $\text{decider}$. On the other hand, if $r$ is not on $K \cap L_V(r)$, then it checks the symmetry of $R_I(r)$ with respect to $K$. If $R_I(r)$ is not symmetric, then the terminal robot on $L_V(r)$ on the dominant half changes its light to $\text{leader}_1$. Otherwise, if $R_I(r)$ is symmetric with respect to $K$, the robot closest to $K$ and on $L_V(r)$ changes its light to $\text{call}$. Note that a robot with light $\text{off}$ also changes its light to $\text{call}$ when it sees another robot with light $\text{call}$ on the same vertical line. In this way, if $r$ is not on $K \cap L_V(r)$ and $R_I(r)$ is symmetric with respect to $K$, all the robots on $L_V(r)$ will eventually change their lights to $\text{call}$ (Figure 12, 13).

Now let $r_{d1}$ be a robot with light $\text{decider}$. It checks whether there is a robot on $K \cap R_I(r_{d1})$. Observe that in the beginning of Phase 2, $r_{d1}$ can see another robot with light $\text{decider}$ (say $r_{d2}$) on $L_V(r_{d1})$, thus can calculate the line $K$. If there is no robot on $R_I(r_{d1}) \cap K$ and $R_I(r_{d1})$ is not symmetric with respect to $K$, then if $r_{d1}$ was in dominant half, it changes its light to $\text{leader}_1$ (Figure 14). In this case, since there are two robots with light $\text{decider}$ and both are terminal, one will always be in the dominant half. Now if there is no robot on $R_I(r_{d1}) \cap K$ and the line $R_I(r)$ (where, $r \in R_I(r_{d1})$) is symmetric with respect to $K$, after a certain time all the robots on $R_I(r_{d1})$ will have light $\text{call}$. 


Algorithm 2: Phase 2

Procedure Phase2()

\[ r \leftarrow \text{myself} \]

if \( r.\text{light} = \text{decider} \) then

if there is a robot with light \text{leader1} on \( \mathcal{R}_I(r) \) or \( \mathcal{L}_V(r) \) then

\[ r.\text{light} \leftarrow \text{off} \]

else

if there is a robot with light \text{decider} on \( \mathcal{L}_V(r) \) and no robots on \( K \cap \mathcal{R}_I(r) \) then

if \( \mathcal{R}_I(r) \) is symmetric with respect to \( K \) then

if all robots in \( \mathcal{R}_I(r) \) are \text{call} then

move right

else

if \( r \) is in dominant half then

\[ r.\text{light} \leftarrow \text{leader1} \]

else if there is a robot with light \text{decider} in \( \mathcal{R}_I(r) \) then

move right

else if there is a robot with light \text{call} in \( \mathcal{L}_V(r) \) and no robot with light \text{decider} in \( \mathcal{L}_I(r) \) then

if all robots in \( \mathcal{R}_I(r) \) are \text{call} then

move right

else if \( r.\text{light} = \text{off} \) then

if there are two robots with light \text{decider} in \( \mathcal{L}_I(r) \) then

if \( r \) is on \( K \cap \mathcal{L}_V(r) \) then

\[ r.\text{light} \leftarrow \text{leader1} \]

else

if \( \mathcal{R}_I(r) \) is symmetric with respect to \( K \) then

if \( r \) is closest to \( K \) or there is a robot with light \text{call} on \( \mathcal{L}_V(r) \) then

\[ r.\text{light} \leftarrow \text{call} \]

else

if \( r \) is terminal on \( \mathcal{L}_V(r) \) and in dominant half then

\[ r.\text{light} \leftarrow \text{leader1} \]

else if \( r.\text{light} = \text{call} \) then

if there is a robot with light \text{leader1} in \( \mathcal{R}_I(r) \) or \( \mathcal{L}_I(r) \) then

\[ r.\text{light} \leftarrow \text{off} \]

else if \( r.\text{light} = \text{leader1} \) then

if (all robots in \( \mathcal{L}_I(r) \) are \text{off} \) or \( H^O_L(r) \) is empty and all robots in \( \mathcal{R}_I(r) \) are \text{off} \) then

if there is no other robot in \( H^F_L(r) \) or \( H^E_L(r) \) then

if there is other robot in \( H^F_L(r) \) then

move left

else

\[ r.\text{light} \leftarrow \text{leader} \]

else

if \( r \) is not terminal on \( \mathcal{L}_V(r) \) then

move left

else if \( r \) is terminal on \( \mathcal{L}_V(r) \) and there is another robot \( r' \) on \( \mathcal{L}_V(r) \) then

move vertically opposite to \( r' \)

else if \( r \) is singleton on \( \mathcal{L}_V(r) \) then

move vertically according to its positive \( y-\)axis.

else

if there is a robot with light \text{leader1} in \( \mathcal{L}_I(r) \) then

\[ r.\text{light} \leftarrow \text{off} \]
Then if $\mathcal{R}_l(r_{d1}) (= \mathcal{R}_l(r_{d2}))$ is symmetric, both the robots $r_{d1}$ and $r_{d2}$ move right. In case, due to asynchrony if one robot (say, $r_{d1}$) moves right and wakes before $r_{d2}$ moves, then even if $r_{d1}$ sees all robots on $\mathcal{R}_l(r_{d1})$ with light call, it will not move right until it sees there is no robot with light decider on $\mathcal{L}_l(r_{d1})$ and sees a robot with light call or $r_{d2}$ on $\mathcal{L}_V(r_{d1})$. Note that $r_{d1}$ will see $r_{d2}$ on $\mathcal{L}_l(r_{d1})$ until $r_{d2}$ moves right. On the other hand, $r_{d2}$ sees $r_{d1}$ on $\mathcal{R}_l(r_{d2})$ and then it moves right.

On a lighter note, in this algorithm a robot $r$ with light off basically calculates the line $K$ by seeing the robots with light decider on $\mathcal{L}_l(r)$ and checks the symmetry of $\mathcal{R}_l(r)$ with respect to the line $K$ and checks if it is on $K$. Note that, the robots with light decider also can calculate $K$ when they see each other on the same vertical line and check the symmetry of the next vertical line having robots and also do search for robots on $K$. As the configuration is solvable and movement of robots with light decider do not actually make the configuration unsolvable, after a certain time there will be some robot $r_0$ with light leader1. Note that a robot $r$ with light call changes the light to off if it sees any robot with light leader1 on $\mathcal{R}_l(r)$ or $\mathcal{L}_l(r)$.

Now, let us consider the case where there is a robot with light leader1 (say, $r_0$). It first checks whether all robots on the line $\mathcal{L}_l(r_0)$ have light off or if its left open half is empty. If one of the cases becomes true and all robots on $\mathcal{R}_l(r_0)$ have light off, then $r_0$ checks whether one of $H^C_U(r_0)$ or $H^C_B(r_0)$ has no other robots. If so and also there is other robots on the left closed half ($H^C_L(r_0)$) of $r_0$, then it moves left until $H^C_U(r_0)$ has no other robots. Then it changes its light to leader. On the other hand, if both $H^C_U(r_0)$ and $H^C_B(r_0)$ have other robots, then $r_0$ checks whether it is a terminal robot on $\mathcal{L}_V(r_0)$. Let us assume $r_0$ is terminal on $\mathcal{L}_V(r_0)$. Now, if $r_0$ is singleton on $\mathcal{L}_V(r_0)$, then it moves according to its positive y-axis until either $H^C_U(r_0)$ or $H^C_B(r_0)$ has no other robots. If $r_0$ is not singleton but terminal on $\mathcal{L}_V(r_0)$, then there exists a robot (say, $r'$) on $\mathcal{L}_V(r_0)$. In this case, $r_0$ moves vertically in the opposite direction.
of \( r' \) until \( H^R_U(r_0) \) or \( H^B_U(r_0) \) has no other robots. If \( r_0 \) was not terminal on \( L'_V(r_0) \), then it moves left. Note that using such movements, \( r_0 \) always reaches a grid point such that either \( H^C_U(r_0) \cap H^R_U(r_0) \) or \( H^C_B(r_0) \cap H^R_U(r_0) \) has no other robots where it changes its light to leader. Also, it might happen that a robot with light leader already sees another robot with light leader1 (Figure 17). In this case, the robot with light leader1, who sees the other robot with light leader1 on its left, changes light to off (Figure 18). Also if a robot with light decider (say, \( r \)) finds a robot with light leader1 on \( R_I(r) \), then it changes its light to off. Thus after a certain time, there is exactly one robot with light leader and others have light off.

So, after completion of Phase 2, the configuration transforms into a leader configuration. The following Lemmas 4, 5, 6 and 7 justify the correctness of the Algorithm 2.

**Lemma 4.** In Phase 2, a robot \( r \) with light decider changes its light to leader1 only if \( H^C_U(r) \) has no other robots.

**Proof.** Let \( r \) and \( r' \) be two robots with light decider at some time \( T > 0 \) in Phase 2 on the same vertical line. Without loss of generality, let us assume that \( r \) be the robot which changes its light at time \( T_1 > T \) to leader1 while it was still on the same vertical line at time \( T \). Note that by Algorithm 2, this only happens if \( R_I(r) \) is not symmetric with respect to \( K \) and \( r \) is on the dominant half at \( T_1 \). We will denote the vertical line on which a robot \( r \) situated at any time \( t \) by \( L^C_t(r) \).

Let us assume that \( H^C_U(r) \) has other robots. Now note that a robot \( r \) with light decider moves right if it sees all the robots on \( R_I(r) \) have light call or sees another robot with light decider on \( R_I(r) \). Also in the beginning of Phase 2, \( H^C_U(r) \) (= \( H^C_U(r') \)) has no other robots. So, at the time \( T \), \( H^C_U(r) \) has other robots and still has light decider, implies \( r \) and \( r' \) moved right at least once and did not see any robot with light leader1 on \( R_I(r) \) till that time. Let us assume that at a time \( (0 <)T_2 < T \), \( r \) and \( r' \) were on \( L^C_{T_2}(r) \) which is actually \( L_I(r) \) at time \( T \). Now \( r \) and \( r' \) moves to \( L^C_{T}(r) \) from \( L^C_{T_2}(r) \) only when all robots on \( L^C_{T}(r) \) have light call. That is only possible if at time \( T \), the robots on \( R_I(r) \) are symmetric with respect to \( K \). Thus it is impossible for \( r \) to change its light to leader1 while still on the same line \( L^C_T(r) \) at time \( T_1 \). So our assumption that \( H^C_U(r) \) has other robots at time \( T \), is wrong. Hence the result follows (Figure 14).

\( \Box \)

**Lemma 5.** If after a certain time \( T > 0 \) in Phase 2, the configuration \( C(T) \) has two robots \( r \) and \( r' \) both with light leader1, then \( \exists \ 0 < T_1, T_2 < T \) such that \( r \) (or \( r' \)) changed its light to leader1 from decider at \( T_1 \) and \( r' \) (or, \( r \)) changed its light to leader1 from off at \( T_2 \).

**Proof.** First, we will show that both the robots \( r \) and \( r' \) can not have changed their lights to leader1 from light decider at \( T_1 \) and \( T_2 \). From the Algorithm 2, it is clear that a robot \( r \) with light decider changes its light to leader1 only if \( R_I(r) \) is not symmetric, \( K \cap R_I(r) \) is empty and \( r \) is in dominant half. Now both the robots with
Figure 14. The robot $r$ with light decider changes its light to leader1 with $H_C^E(r)$ has no other robots.

light decider can not be on the dominant half. So, both $r$ and $r'$ can not have change their lights to leader1 from decider at $T_1$ and $T_2$.

Secondly, it has to be shown that both $r$ and $r'$ can not have changed their lights to leader1 from light off at $T_1$ and $T_2$. From Algorithm 2, it is obvious that a robot $r_0$ with light off can only change its light to leader1 if it sees two decider robots $r_{d1}$ and $r_{d2}$ on $L_I(r_0)$. So if at a time $T' < T_1, T_2$ both $r$ and $r'$ had light off, then $r$ and $r'$ was on the same vertical line $L_V(r) (= L_V(r'))$. Now we will show by contradiction that it is not possible for both $r$ and $r'$ to change their lights to leader1. Let us assume $R_I(r) (= R_I(r'))$ is symmetric with respect to $K$. Then $r$ or $r'$ will only change their lights to leader1 if both are on the grid position $K \cap R_I(r_{d1})$. Which is not possible. So, let us now assume $R_I(r)$ is not symmetric with respect to $K$ and also let one of $r$ or, $r'$ is on $K \cap R_I(r_{d1})$. Without loss of generality, let $r$ is on $K \cap R_I(r_{d1})$. Then according to the Algorithm 2, $r$ changes its light to leader1 but no other robot on $L_V(r)$ changes their lights as they are not closest to $K$ or sees no robot with light call on $L_V(r)$. Hence in this case, there can be only one robot on $L_V(r)$ who will change its light to leader1, arriving at a contradiction again. So let $R_I(r)$ is not symmetric and there is no robot on the grid point $K \cap R_I(r_{d1})$. Then only the terminal robot of $L_V(r)$ who is in the dominant half changes its color to leader1. So again a contradiction that both $r$ and $r'$ on $L_V(r)$ change their light to leader1. Hence our assumption was wrong.

Hence one of $r$ or $r'$ will change its light to leader1 from light decider at a time $T_1 < T$ and the other robot will change its light to leader1 from light off at time $T_2 < T$ (Figure 15, 16).
Lemma 6. If in Phase 2, there exists a configuration $C(T)$ at a time $T > 0$ such that $\exists$ a robot $r'$ with light $\text{leader1}$ which sees another robot $r$ with light $\text{leader1}$ on $L_1(r')$ in $C$, then $r'$ will not move and change its light to off.

Proof. Since at time $T$, there are two robots with light $\text{leader1}$ in the configuration during Phase 2, robot $r$ must have changed its light to $\text{leader1}$ from light $\text{decider}$ and the other robot $r'$ must have changed the light to $\text{leader1}$ from off (by Lemma 5). Now from Lemma 4, $H_0^C(r)$ has no other robots. Also $r'$ is on $R_I(r)$ (as $r'$ can only change its light to $\text{leader1}$ if it has seen two robots with light $\text{decider}$ on $L_1(r')$).

Thus it can be seen that if the configuration has two robots with light $\text{leader1}$ at any time $T$, the two robots will be on two consecutive occupied vertical line. So by Algorithm 2, $r'$ will see $r$ on $L_I(r')$ with light $\text{leader1}$ and change its light to off without moving (Figure 17, 18).

Lemma 7. If a robot $r$ with light $\text{leader1}$ is not terminal on $L_V(r)$ and does not see any robot with light $\text{leader1}$ on $L_1(r)$, then $H_0^C(r) \cap L_H(r)$ does not contain any robot.

Proof. Let $r$ is not terminal on $L_V(r)$ with light $\text{leader1}$. Then by Algorithm 2, $r$ is on $K \cap R_I(r_{d_1})$ ($R_I(r_{d_1}) = R_I(r_{d_2})$), where $r_{d_1}$ and $r_{d_2}$ be two robots with light $\text{decider}$. If possible let, there is a robot $r'$ on $H_0^C(r) \cap L_H(r)$. Then $\exists T > 0$ such that $r_{d_1}$ and $r_{d_2}$ are on $L_1(r')$. Now observe that at time $T$, $r'$ is on $K \cap R_I(r_{d_1})$, which implies $r'$ will change its light to $\text{leader1}$. Then both $r_{d_1}$ and $r_{d_2}$ change their lights to off seeing $r'$ on $R_I(r_{d_1})$ with light $\text{leader1}$. Hence $r_{d_1}$ and $r_{d_2}$ never reach $L_I(r)$. Thus $r$ can never change its light to $\text{leader1}$.

So we can conclude that our primary assumption was wrong. Hence the result follows (Figure 19).
3.2. Pattern formation from leader configuration

In this phase, initially the configuration is a leader configuration. Note that the robots can agree on a global co-ordinate system based on the position of the robot $r_0$ with light leader. We denote the position of $r_0$ with the co-ordinate $(0, -1)$. Also all robots with light off lie on one of the open half planes delimited by the horizontal line $L_H(r_0)$. This half plane will correspond to positive direction of $Y$-axis (Figure 20).
Algorithm 3: Pattern Formation from Leader Configuration

Input : The configuration of robots visible to me.

1 Procedure PatternFormationFromLeaderConfiguration()
2  \( r \leftarrow \) myself
3  \( r_0 \leftarrow \) the robot with light leader
4  if \( r \text{.light} = \text{off} \) then
5      if \( (r_0 \in H^O_H(r)) \) and \( (r \text{ is leftmost on } \mathcal{L}_H(r)) \) and (there is no robot in 
6      \( H^O_H(r) \cap H^O_H(r_0) \)) then
7          if there is a robot with light done then
8              \( r \text{.light} \leftarrow \text{done} \)
9          else
10             TargetMove(\( n - 2 \))
11    else if there are \( i \) robots on \( \mathcal{L}_H(r_0) \) other than \( r_0 \) at
12       \((1, -1), \ldots, (i, -1)\) then
13           LineMove(\( i + 1 \))
14    else if there are \( i \) robots on \( \mathcal{L}_H(r_0) \) other than \( r_0 \) at
15       \((n - i, -1), \ldots, (n - 1, -1)\) then
16           if \( r \) is at \( t_{n-i-2} \) then
17               \( r \text{.light} \leftarrow \text{done} \)
18           else
19               TargetMove(\( n - i - 2 \))
20  else if \( r_0 \in \mathcal{L}_H(r) \) and \( H^O_H(r) \) has no robots with light off then
21     if \( r \) is at \((i, -1)\) then
22          \text{Move to } (i, 0)
23  else if \( r \text{.light} = \text{leader} \) then
24     if there are no robots with light off then
25        if \( r \) is at \( t_{n-1} \) then
26           \( r \text{.light} \leftarrow \text{done} \)
27        else
28           TargetMove(\( n - 1 \))
29
30 Procedure LineMove(\( j \))
31  if \( r \) is on \( \mathcal{L}_{H_1}(r) \) then
32     if \( r \) is at \((j, 0)\) then
33         \text{Move to } (j, -1)
34     else
35         \text{Move horizontally towards } (j, 0)
36  else \text{Move vertically towards } \mathcal{L}_{H_1}(r)
37
38 Procedure TargetMove(\( j \))
39  if \( r \) is on \( \mathcal{L}_{t_j-1}(r) \) then
40     if \( r \) is at \((t_j(x), t_j(y) - 1)\) then
41         \text{Move to } (t_j(x), t_j(y))
42     else \text{Move horizontally towards } (t_j(x), t_j(y) - 1)
43  else \text{Move vertically towards } \mathcal{L}_{t_j-1}(r)
Therefore an agreement on a global co-ordinate system can happen between the robots who see \( r_0 \) at \((0, -1)\). After completion of this phase, the robots achieve the target configuration (Figure 21). The robots first form a compact line and from that line the robots then move to their designated target positions. The difficulty of this phase is to differentiate between two configurations where a robot is going to form a compact line and where the robot is going to its target position which are described in subsections 3.2.1 and 3.2.2.

### 3.2.1. Compact line formation

Observe that according to the Algorithm 3, a robot with light leader will not move during the formation of line. Also at the beginning of Phase 3, there are no other robots on the line \( L_H(r_0) \). A robot \( r \) with light off will first check if it can see \( r_0 \) and if it is the leftmost robot on the line \( L_H(r) \) and also if there are any robots on \( H_D^1(r) \cap H_D^0(r_0) \). If all the conditions are true, (i.e \( r \) is the leftmost robot in its horizontal line and there are no other robots in between horizontal lines of \( r \) and \( r_0) \) \( r \) counts the number of robots on \( L_H(r_0) \). Lets assume if there are no robots on \( L_H(r_0) \) except \( r_0 \). In this case, \( r \) checks for other robots with light done on the grid. Note that a robot changes its light to done only if it has reached its target position. So, clearly while forming the line, \( r \) finds that there are no robots with light done on the grid and moves to the position \((1, -1)\) following the procedure Linemove(1).

On the other hand if there are \( i \) robots except \( r_0 \) who are already in the line \( L_H(r_0) \) occupying the positions \((1, -1), (2, -1)\ldots(i, -1)\), then \( r \) simply move to \((i + 1, -1)\) following the procedure Linemove\((i + 1)\).

During the procedure Linemove\((j)\), a robot (say, \( r \)) can recognize if it is on the line \( L_{H_1} \), where \( L_{H_1} \) is the immediate horizontal line above \( L_H(r_0) \). If it is not on \( L_{H_1} \), it moves vertically downwards until it reaches \( L_{H_1} \). Otherwise, if \( r \) is already on \( L_{H_1} \), it checks if it is on the co-ordinate \((j, 0)\). If it is not on \((j, 0)\), it moves horizontally to \((j, 0)\). Observe that during this horizontal movement, there will be no collision as \( r \) will be the only robot on \( L_{H_1} \) due to the fact that the robots in Algorithm 3 move sequentially. Now, if \( r \) is on the co-ordinate \((j, 0)\), it moves vertically to the co-ordinate \((j, -1)\) (Figure 22, 23). Note that since the robots move sequentially, no robot will change their light to done before forming the compact line.

### 3.2.2. Target Pattern Formation

After formation of the compact line, the robots now will move to the target co-ordinates. Note that the target co-ordinates are unique for each robot who can see the robot \( r_0 \) with light leader as there is an agreement on global co-ordinate. The target co-ordinates are denoted as \( t_i \), where \( i \in \{0, 1, 2, \ldots n-1\} \). Also observe that if \( t_i \) and \( t_j \) are in the same horizontal line where \( i < j \), then \( t_i \) is on the right of \( t_j \). And if \( t_i \) and \( t_j \) are not on the same horizontal line, then \( t_i \) will be above of \( t_j \).

If a robot sees \( r_0 \) on the same horizontal line, it vertically moves to \( L_{H_1} \) with \( y \)-coordinate 0. Now observe that after moving to \( L_{H_1} \) with \( y \)-coordinate 0, it can see all the robots on the line including the robot \( r_0 \).
Figure 20. A leader configuration where robot $r_0$ is the robot with light leader at $(0, -1)$ in the agreed coordinate system.

Figure 21. The pattern embedded in the coordinate system.

Figure 22. Movement of $r_5$ to $L_H(r_0)$.

Figure 23. All robots forms an compact line on $L_H(r_0)$.

From this information about the number of robots on the line except $r_0$, it can calculate the target position it need to reach and follow the procedure Target-Move() to reach that position. Note that the robot changes its light to done only after reaching its designated target position and in the mean time, no other robot moves from the line while they see robots with light off above them. This technique, of moving the robots sequentially and the ordering of the target co-ordinates, avoid collision in our algorithm. Thus all robots except $r_0$ and $r_{n-1}$ reach their designated target co-ordinates. Now while $r_{n-1}$ moves above from $(n - 1, -1)$ to $(n - 1, 0)$, it sees
there is no other robot on the line $L_H(r_0)$ except $r_0$ and moves to $t_{n-2}$. And finally
the robot with light leader moves to the only remaining vacant target position $t_{n-1}$.

If a robot (say, $r$) executes the procedure $\text{TARGET\_MOVE}(j)$, that means that
the target position of $r$ is $t_j$ with co-ordinate $(t_j(x), t_j(y))$. Observe that, $r$ executes
this procedure when it is on $L_{H1}$. Now during this procedure, if $r$ is not on $L_{t_j-1}$,
then it moves vertically upwards until it reaches $L_{t_j-1}$. Now, when $r$ is at $L_{t_j-1}$, it
checks if the co-ordinate of its current position is $(t_j(x), t_j(y) - 1)$. If not, then it
moves horizontally to reach the point with the co-ordinate $(t_j(x), t_j(y) - 1)$. After
that it moves vertically upwards to the point with co-ordinate $(t_j(x), t_j(y))$, which
is the target position of $r$ (Figure 24, 25, 26). Note that during the movement of $r$,
there will be no collision as the closed half delimited by $L_{t_{i-1}}$ and $L_H(r)$ does not contain any other robot.

Hence from the above discussions, we can conclude the following theorem.

**Theorem 3.** If $C(T_1)$ is a leader configuration, then $\exists \ T_2 > T_1$ such that $C(T_2)$ is the target configuration.

4. Conclusion

Arbitrary pattern formation ($APF$) has been a very active topic in the field of swarm robotics. It has been thoroughly researched in many different settings. For example, it has been studied when the robots are on a plane or on an infinite grid. Considering obstructed visibility model for robots on a plane, it has been shown that for certain initial configurations, $APF$ is solvable with opaque robots having one axis agreement and 6 lights under asynchronous scheduler ([4]). In [2], $APF$ has been solved even with opaque fat robots with light on plane. Comparing to how thoroughly $APF$ has been studied with robots on plane with obstructed visibility model, it remains quite far behind when it comes to robots on infinite grid with obstructed visibility model. This paper is a stepping stone towards the goal of removing this gap of research regarding $APF$ between opaque robots on the plane and robots on infinite grid.

In this paper, we have provided a deterministic algorithm for $APF$ for all solvable initial configurations with one axis agreement and 8 lights under the asynchronous scheduler. For the immediate course of future research, one can think of solving this problem with less numbers of lights. Another interesting way of extending this problem would be to allow multiplicities in the pattern.

Acknowledgements. The second author is supported by UGC, Government of India and the third author is supported by West Bengal State government Fellowship Scheme.

References

[1] Adhikary R., Kundu M.K., Sau B.: Circle formation by asynchronous opaque robots on infinite grid. In: *Comput. Sci.*, vol. 22(1), 2021. URL http://dx.doi.org/10.7494/csci.2021.22.1.3840.

[2] Bose K., Adhikary R., Kundu M.K., Sau B.: Arbitrary Pattern Formation by Opaque Fat Robots with Lights. In: *CoRR*, vol. abs/1910.02706, 2019. URL http://arxiv.org/abs/1910.02706.

[3] Bose K., Adhikary R., Kundu M.K., Sau B.: Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. In: *Theor. Comput. Sci.*, vol. 815, pp. 213–227, 2020. URL http://dx.doi.org/10.1016/j.tcs.2020.02.016.

[4] Bose K., Kundu M.K., Adhikary R., Sau B.: Arbitrary pattern formation by asynchronous opaque robots with lights. In: *Theor. Comput. Sci.*, vol. 849, pp. 138–158, 2021. URL http://dx.doi.org/10.1016/j.tcs.2020.10.015.
[5] Bramas Q., Tixeuil S.: Probabilistic Asynchronous Arbitrary Pattern Formation (Short Paper). In: B. Bonakdarpour, F. Petit, eds., Stabilization, Safety, and Security of Distributed Systems - 18th International Symposium, SSS 2016, Lyon, France, November 7-10, 2016, Proceedings, Lecture Notes in Computer Science, vol. 10083, pp. 88–93. 2016. URL http://dx.doi.org/10.1007/978-3-319-49259-9\_7.

[6] Bramas Q., Tixeuil S.: Arbitrary Pattern Formation with Four Robots. In: T. Izumi, P. Kuznetsov, eds., Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018, Tokyo, Japan, November 4-7, 2018, Proceedings, Lecture Notes in Computer Science, vol. 11201, pp. 333–348. Springer, 2018. URL http://dx.doi.org/10.1007/978-3-030-03232-6\_22.

[7] Cicerone S., Stefano G.D., Navarra A.: Asynchronous Arbitrary Pattern Formation: the effects of a rigorous approach. In: Distributed Comput., vol. 32(2), pp. 91–132, 2019. URL http://dx.doi.org/10.1007/s00446-018-0325-7.

[8] Cicerone S., Stefano G.D., Navarra A.: Embedded pattern formation by asynchronous robots without chirality. In: Distributed Comput., vol. 32(4), pp. 291–315, 2019. URL http://dx.doi.org/10.1007/s00446-018-0333-7.

[9] Das S., Flocchini P., Santoro N., Yamashita M.: Forming sequences of geometric patterns with oblivious mobile robots. In: Distributed Comput., vol. 28(2), pp. 131–145, 2015. URL http://dx.doi.org/10.1007/s00446-014-0220-9.

[10] Dieudonné Y., Petit F., Villain V.: Leader Election Problem versus Pattern Formation Problem. In: N.A. Lynch, A.A. Shvartsman, eds., Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings, Lecture Notes in Computer Science, vol. 6343, pp. 267–281. Springer, 2010. URL http://dx.doi.org/10.1007/978-3-642-15763-9\_26.

[11] Feletti C., Mereghetti C., Palano B.: Uniform Circle Formation for Swarms of Opaque Robots with Lights. In: T. Izumi, P. Kuznetsov, eds., Stabilization, Safety, and Security of Distributed Systems - 20th International Symposium, SSS 2018, Tokyo, Japan, November 4-7, 2018, Proceedings, Lecture Notes in Computer Science, vol. 11201, pp. 317–332. Springer, 2018. URL http://dx.doi.org/10.1007/978-3-030-03232-6\_21.

[12] Flocchini P., Prencipe G., Santoro N., Widmayer P.: Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. In: Theor. Comput. Sci., vol. 407(1-3), pp. 412–447, 2008. URL http://dx.doi.org/10.1016/j.tcs.2008.07.026.

[13] Fujinaga N., Yamauchi Y., Ono H., Kijima S., Yamashita M.: Pattern Formation by Oblivious Asynchronous Mobile Robots. In: SIAM J. Comput., vol. 44(3), pp. 740–785, 2015. URL http://dx.doi.org/10.1137/140958682.

[14] Lukovszki T., auf der Heide F.M.: Fast Collisionless Pattern Formation by Anonymous, Position-Aware Robots. In: M.K. Aguilera, L. Querzoni, M. Shapiro, eds., Principles of Distributed Systems - 18th International Conference, OPODIS 2014, Cortina d’Ampezzo, Italy, December 16-19, 2014. Proceedings, Lecture
[15] Peleg D.: Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges. In: A. Pal, A.D. Kshemkalyani, R. Kumar, A. Gupta, eds., Distributed Computing - IWDC 2005, 7th International Workshop, Kharagpur, India, December 27-30, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3741, pp. 1–12. Springer, 2005. URL http://dx.doi.org/10.1007/11603771_1.

[16] Suzuki I., Yarnashita M.: Distributed Anonymous Mobile Robots - Formation and Agreement Problems. In: Problems, in the Proceedings of the 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO ’96), pp. 1347–1363. 1996.

[17] Vaidyanathan R., Sharma G., Trahan J.: On fast pattern formation by autonomous robots. In: Information and Computation, p. 104699, 2021. ISSN 0890-5401. URL http://dx.doi.org/https://doi.org/10.1016/j.ic.2021.104699.

[18] Yamashita M., Suzuki I.: Characterizing geometric patterns formable by oblivious anonymous mobile robots. In: Theoretical Computer Science, vol. 411(26), pp. 2433–2453, 2010. ISSN 0304-3975. URL http://dx.doi.org/https://doi.org/10.1016/j.tcs.2010.01.037.

[19] Yamauchi Y., Yamashita M.: Pattern Formation by Mobile Robots with Limited Visibility. 2013. ISBN 978-3-319-03577-2. URL http://dx.doi.org/10.1007/978-3-319-03578-9_17.
Affiliations

Manash Kumar Kundu
Gayeshpur Government Polytechnic, Department of Science and Humanities, Kalyani, West Bengal - 741234, India, ORCID iD: https://orcid.org/0000-0003-4179-8293, email: manashkrkundu.rs@jadavpuruniversity.in

Pritam Goswami
Jadavpur University, Department of Mathematics, Kolkata, West Bengal - 700032, India, ORCID iD: https://orcid.org/0000-0002-0546-3894, email: pritamgoswami.math.rs@jadavpuruniversity.in

Satakshi Ghosh
Jadavpur University, Department of Mathematics, Kolkata, West Bengal - 700032, India, ORCID iD: https://orcid.org/0000-0003-1747-4037, email: satakshighosh.math.rs@jadavpuruniversity.in

Buddhadeb Sau
Jadavpur University, Department of Mathematics, Kolkata, West Bengal - 700032, India, email: buddhadeb.sau@jadavpuruniversity.in

Received: ???
Revised: ???
Accepted: ???